non_ferrous_fluctuating_compound

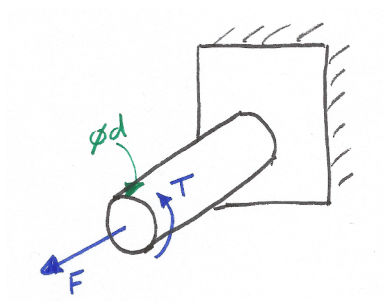
December 6, 2015

1 Problem Statement

The round bar shown is under the fluctuating loads F and T, which are in phase. The material is cold drawn 2024 Aluminum. Find the number of cycles to failure if $F_{max} = 80kN$, $F_{min} = 50kN$, $T_{max} = 1000Nm$, $T_{min} = 250Nm$, and $d = 40mm$.

In [1]:
```
from IPython.display import Image
Image('bar.png')
```

Out[1]:



In [2]:
```
from sympy import *
init_printing(use_latex='mathjax')
```

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
from IPython.core.pylabtools import figsize
%matplotlib inline
figsize(12, 8)
```

1.1 Knowns

In [4]:
```
Syt, Sut, Se, Sf50E7p = symbols("S_yt, S_ut, S_e, S'_f_50E7")
Fmax, Fmin, Tmax, Tmin, d = symbols('F_max, F_min, T_max, T_min, d')
```

In [5]:
```
knowns = {Syt: 76E6, Sut: 186E6, Sf50E7p: 90E6, d: 0.04, Fmax: 8E4, Fmin: 5E4, Tmax: 1000, Tmin
knowns
```

Out[5]:

$$\left\{F_{max} : 80000.0, \quad F_{min} : 50000.0, \quad S'_{f50E7} : 90000000.0, \quad S_{ut} : 186000000.0, \quad S_{yt} : 76000000.0, \quad T_{max} : 1000, \quad T_{min} : 250, \quad d : 0.04\right\}$$

1.2 Unknowns

```
In [6]: smax, smin, tmax, tmin = symbols('sigma_max, sigma_min, tau_max, tau_min')
        sm, sa, tm, ta, smp, sap = symbols("sigma_m, sigma_a, tau_m, tau_a, \sigma'_m, \sigma'_a")
        ka, kb, Sf50E7, Sf1E3, srev = symbols('k_a, k_b, S_f_50E7, S_f_1E3, sigma_rev')
```

1.3 Find the maximum and minimum stresses

```
In [7]: smax_eq = Eq(smax, Fmax / pi / (d / 2)**2)
        smax_eq, smax_eq.subs(knowns).evalf()
```

Out[7]:

$$\left(\sigma_{max} = \frac{4F_{max}}{\pi d^2}, \quad \sigma_{max} = 63661977.2367581\right)$$

```
In [8]: smin_eq = Eq(smin, Fmin / pi / (d / 2)**2)
        smin_eq, smin_eq.subs(knowns).evalf()
```

Out[8]:

$$\left(\sigma_{min} = \frac{4F_{min}}{\pi d^2}, \quad \sigma_{min} = 39788735.7729738\right)$$

```
In [9]: tmax_eq = Eq(tmax, Tmax * (d / 2) / pi / d**4 * 32)
        tmax_eq, tmax_eq.subs(knowns).evalf()
```

Out[9]:

$$\left(\tau_{max} = \frac{16T_{max}}{\pi d^3}, \quad \tau_{max} = 79577471.5459477\right)$$

```
In [10]: tmin_eq = Eq(tmin, Tmin * (d / 2) / pi / d**4 * 32)
         tmin_eq, tmin_eq.subs(knowns).evalf()
```

Out[10]:

$$\left(\tau_{min} = \frac{16T_{min}}{\pi d^3}, \quad \tau_{min} = 19894367.8864869\right)$$

```
In [11]: sub_exprs = {e.lhs: e.rhs for e in [smax_eq, smin_eq, tmax_eq, tmin_eq]}
         sub_exprs
```

Out[11]:

$$\left\{\sigma_{max} : \frac{4F_{max}}{\pi d^2}, \quad \sigma_{min} : \frac{4F_{min}}{\pi d^2}, \quad \tau_{max} : \frac{16T_{max}}{\pi d^3}, \quad \tau_{min} : \frac{16T_{min}}{\pi d^3}\right\}$$

1.4 Find the mean and amplitude stress

```
In [12]: sm_eq = Eq(sm, (smax + smin) / 2)
         sm_eq, sm_eq.subs(sub_exprs), sm_eq.subs(sub_exprs).subs(knowns).evalf()
```

Out[12]:

$$\left(\sigma_m = \frac{\sigma_{max}}{2} + \frac{\sigma_{min}}{2}, \quad \sigma_m = \frac{2F_{max}}{\pi d^2} + \frac{2F_{min}}{\pi d^2}, \quad \sigma_m = 51725356.504866\right)$$

```
In [13]: sa_eq = Eq(sa, (smax - smin) / 2)
         sa_eq, sa_eq.subs(sub_exprs), sa_eq.subs(sub_exprs).subs(knowns).evalf()
```

Out[13]:

$$\left(\sigma_a = \frac{\sigma_{max}}{2} - \frac{\sigma_{min}}{2}, \quad \sigma_a = \frac{2F_{max}}{\pi d^2} - \frac{2F_{min}}{\pi d^2}, \quad \sigma_a = 11936620.7318922\right)$$

```
In [14]: tm_eq = Eq(tm, (tmax + tmin) / 2)
         tm_eq, tm_eq.subs(sub_exprs), tm_eq.subs(sub_exprs).subs(knowns).evalf()
```

Out[14]:

$$\left(\tau_{m} = \frac{\tau_{max}}{2} + \frac{\tau_{min}}{2}, \quad \tau_{m} = \frac{8T_{max}}{\pi d^{3}} + \frac{8T_{min}}{\pi d^{3}}, \quad \tau_{m} = 49735919.7162173\right)$$

```
In [15]: ta_eq = Eq(ta, (tmax - tmin) / 2)
         ta_eq, ta_eq.subs(sub_exprs), ta_eq.subs(sub_exprs).subs(knowns).evalf()
```

Out[15]:

$$\left(\tau_{a} = \frac{\tau_{max}}{2} - \frac{\tau_{min}}{2}, \quad \tau_{a} = \frac{8T_{max}}{\pi d^{3}} - \frac{8T_{min}}{\pi d^{3}}, \quad \tau_{a} = 29841551.8297304\right)$$

```
In [16]: for e in [sm_eq, sa_eq, tm_eq, ta_eq]:
             sub_exprs[e.lhs] = e.rhs.subs(sub_exprs)
```

1.5 Find the von Mises's stress

```
In [17]: sap_eq = Eq(sap, sqrt((sa / 0.85)**2 + 3 * ta**2))
         sap_eq, sap_eq.subs(sub_exprs), sap_eq.subs(sub_exprs).subs(knowns).evalf()
```

Out[17]:

$$\left(\sigma'_{a} = \sqrt{1.3840830449827\sigma_{a}^{2} + 3\tau_{a}^{2}}, \quad \sigma'_{a} = \sqrt{1.3840830449827\left(\frac{2F_{max}}{\pi d^{2}} - \frac{2F_{min}}{\pi d^{2}}\right)^{2} + 3\left(\frac{8T_{max}}{\pi d^{3}} - \frac{8T_{min}}{\pi d^{3}}\right)^{2}}, \quad \sigma'_{a} = 53560833.0123283\right)$$

```
In [18]: smp_eq = Eq(smp, sqrt(sm**2 + 3 * tm**2))
         smp_eq, smp_eq.subs(sub_exprs), smp_eq.subs(sub_exprs).subs(knowns).evalf()
```

Out[18]:

$$\left(\sigma'_{m} = \sqrt{\sigma_{m}^{2} + 3\tau_{m}^{2}}, \quad \sigma'_{m} = \sqrt{\left(\frac{2F_{max}}{\pi d^{2}} + \frac{2F_{min}}{\pi d^{2}}\right)^{2} + 3\left(\frac{8T_{max}}{\pi d^{3}} + \frac{8T_{min}}{\pi d^{3}}\right)^{2}}, \quad \sigma'_{m} = 100481329.786232\right)$$

```
In [19]: sub_exprs[smp] = smp_eq.subs(sub_exprs).rhs
         sub_exprs[sap] = sap_eq.subs(sub_exprs).rhs
```

1.6 Calculate the modified fatigue strength

```
In [20]: ka_eq = Eq(ka, 4.51 * (Sut / 1E6) **-0.265)   # Sut must be in MPa
         ka_eq
```

Out[20]:

$$k_{a} = \frac{175.459360392421}{S_{ut}^{0.265}}$$

```
In [21]: kb_eq = Eq(kb, 1.24 * (d * 1E3)**-0.107)   # d must be in mm
         kb_eq
```

Out[21]:

$$k_{b} = \frac{0.592136299335537}{d^{0.107}}$$

```
In [22]: Sf50E7_eq = Eq(Sf50E7, ka_eq.rhs * kb_eq.rhs * Sf50E7p)
         Sf50E7_eq.evalf(n=3), Sf50E7_eq.subs(knowns)
```

Out[22]:

$$\left(S_{f50E7} = \frac{104.0S'_{f50E7}}{S_{ut}^{0.265}d^{0.107}}, \quad S_{f50E7} = 84917927.6802828\right)$$

```
In [23]: sub_exprs[Sf50E7] = Sf50E7_eq.rhs
```

1.7 Calculate the low cycle fatigue strength

```
In [24]: Sf1E3_eq = Eq(Sf1E3, 0.9 * Sut)  # 186 MPa = 27 kpsi < 70 kpsi
         Sf1E3_eq, Sf1E3_eq.subs(knowns)
```

Out[24]:

$$\left(S_{f1E3} = 0.9 S_{ut}, \quad S_{f1E3} = 167400000.0\right)$$

```
In [25]: sub_exprs[Sf1E3] = Sf1E3_eq.rhs
```

1.8 Find the equivalent fully reversed stress

```
In [26]: srev_eq = Eq(srev, solve(sap / srev + smp / Sut - 1, srev)[0])
         srev_eq, srev_eq.subs(sub_exprs), srev_eq.subs(sub_exprs).subs(knowns).evalf()
```

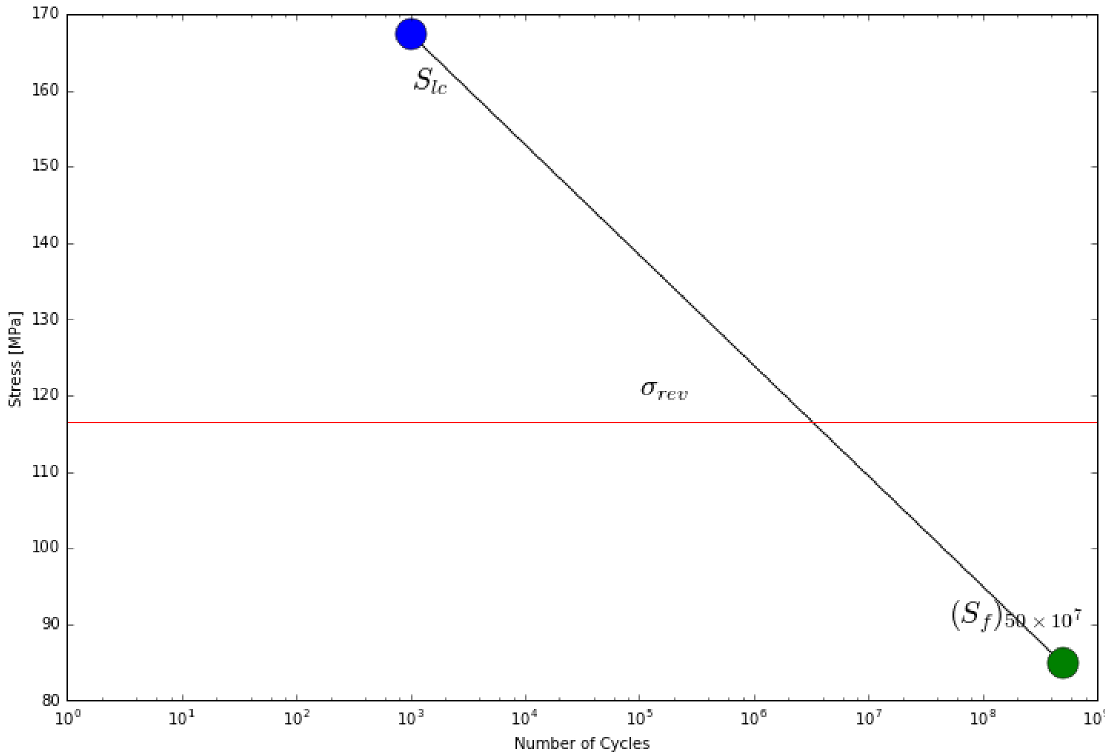
Out[26]:

$$\left(\sigma_{rev} = \frac{S_{ut} \sigma'_{a}}{S_{ut} - \sigma'_{m}}, \quad \sigma_{rev} = \frac{S_{ut} \sqrt{1.38408304498 27 \left(\frac{2 F_{max}}{\pi d^{2}} - \frac{2 F_{min}}{\pi d^{2}}\right)^{2} + 3 \left(\frac{8 T_{max}}{\pi d^{3}} - \frac{8 T_{min}}{\pi d^{3}}\right)^{2}}}{S_{ut} - \sqrt{\left(\frac{2 F_{max}}{\pi d^{2}} + \frac{2 F_{min}}{\pi d^{2}}\right)^{2} + 3 \left(\frac{8 T_{max}}{\pi d^{3}} + \frac{8 T_{min}}{\pi d^{3}}\right)^{2}}}, \quad \sigma_{rev} = 116492865.422143\right)$$

1.9 Find the number of cycles

The following S-N curve shows the two points that define the non-ferrous finite life line. The horizontal red line shows the equivalent fully reveresed stress.

```
In [27]: fig, ax = plt.subplots()
         plt.semilogx([1E3, 50E7], [Sf1E3_eq.subs(knowns).rhs / 1E6,
                                    Sf50E7_eq.subs(knowns).rhs / 1E6], color='black')
         plt.semilogx([1E3], [Sf1E3_eq.subs(knowns).rhs / 1E6], 'o', markersize=20)
         ax.annotate('$S_{lc}$', xy=[1E3, 160], fontsize=20)
         plt.semilogx([50E7], [Sf50E7_eq.subs(knowns).rhs / 1E6], 'o', markersize=20)
         ax.annotate(r'$(S_f)_{50 \times 10^7}$', xy=[5E7, 90], fontsize=20)
         plt.semilogx(np.logspace(0, 9), srev_eq.subs(sub_exprs).subs(knowns).evalf().rhs * np.ones(50)
         ax.annotate(r'$\sigma_{rev}$', xy=[1E5, 120], fontsize=20)
         plt.ylabel('Stress [MPa]')
         plt.xlabel('Number of Cycles')
         lims = plt.xlim(10**0, 10**9)
```



To find the number of cycles at σ_{rev}, simply interpolate between the blue and green points.

In [28]:
```
N = symbols('N')
interp = Eq((Sf50E7 - Sf1E3) / (log(50E7, 10) - log(1E3, 10)), (Sf50E7 - srev) / (log(50E7, 10
interp
```

Out[28]:

$$\frac{-S_{f1E3}+S_{f50E7}}{\frac{\log(5)}{\log(10)}+5} = \frac{S_{f50E7}-\sigma_{rev}}{-\frac{\log(N)}{\log(10)}+\frac{\log(5)}{\log(10)}+8}$$

The above can be solved for $log_10(N)$ and then the solution is simply raised to the power of 10 to find the number of cycles.

In [29]:
```
sub_exprs[srev] = srev_eq.subs(sub_exprs).rhs
```

In [30]:
```
logN_sol = solve(interp, log(N, 10))[0].subs(sub_exprs).subs(knowns)
Eq(log(N, 10), logN_sol.evalf())
```

Out[30]:

$$\frac{\log(N)}{\log(10)} = 6.51734898028931$$

In [31]:
```
Eq(N, (10**logN_sol).evalf())
```

Out[31]:

$$N = 3291159.88042854$$