Machine Design Homework 4

June 19, 2022

Gabe Morris

[1]:
```
# Notebook Preamble
%matplotlib inline
import matplotlib.pyplot as plt
import sympy as sp
import numpy as np
from IPython.display import display, Markdown

plt.style.use('maroon_ipynb.mplstyle')
```

Contents

1 Problem 6-1 **3**
- 1.1 Given . . . 3
- 1.2 Find . . . 3
- 1.3 Solution . . . 3

2 Problem 6-3 **5**
- 2.1 Given . . . 5
- 2.2 Find . . . 5
- 2.3 Solution . . . 5

3 Problem 6-17 **7**
- 3.1 Given . . . 7
- 3.2 Find . . . 7
- 3.3 Solution . . . 7

4 Problem 6-30 **11**
- 4.1 Given . . . 11
- 4.2 Find . . . 11
- 4.3 Solution . . . 11

1 Problem 6-1

1.1 Given

A 10-mm steel drill rod was heat treated and ground. The measured hardness was found to be 300 Brinell.

1.2 Find

Estimate the endurance strength in MPa if the rod is used in rotating bending.

1.3 Solution

Eq. 6-10 on p. 305,

$$S'_e = \begin{cases} 0.5S_{ut} & S_{ut} \leq 200\ ksi\ (1400\ MPa) \\ 100 & S_{ut} > 200\ ksi \\ 700\ MPa & S_{ut} > 1400\ MPa \end{cases}$$

The ultimate strength of steel comes from Eq. 2-36,

$$S_{ut} = 3.4H_B$$

[2]:
```
H_B = 300
S_ut = sp.S('3.4')*H_B

if S_ut <= 1400:
    S_e_prime = 0.5*S_ut
else:
    S_e_prime = sp.S(700)

S_e_prime  # ksi
```

[2]: 510.0

This value is not the final value. The relationship for the refined value is,

$$S_e = k_a k_b k_c k_d k_e S'_e$$

The only necessary k values used for this analysis is k_a and k_b, whose equations are at 6-18 and 6-19 respectfully.

[3]:
```
# See Table 6-2
k_a = sp.S('1.38')*S_ut**-(sp.S('0.067'))
d = 10
```

```python
k_b = sp.S('1.24')*d**-(sp.S('0.107'))
# display(k_a, k_b)
S_e = k_a*k_b*S_e_prime
S_e  # MPa
```

[3]:

428.839455736079

2 Problem 6-3

2.1 Given

A steel rotating beam test specimen has an ultimate strength of 120 ksi.

2.2 Find

Estimate the life of the specimen if it is tested at completely reversed stress amplitude of 70 ksi.

2.3 Solution

Find S_e first.

[4]:
```
S_ut = sp.S(120)  # ksi

if S_ut <= 200:
    S_e_prime = 0.5*S_ut
else:
    S_e_prime = sp.S(100)

S_e_prime  # ksi
```

[4]:

60.0

The S_e' value will be used in place of S_e from Figure 6-23 description. We can use the following relationships to determine N.

$$N = \left(\frac{\sigma_{ar}}{a}\right)^{1/b}$$
$$a = \frac{(fS_{ut})^2}{S_e}$$
$$b = -\frac{1}{3}\log\left(\frac{fS_{ut}}{Se}\right)$$

The value of f is 0.82 from Figure 6-23. The S_{ut} value is $2(S_e) = 120$ ksi.

[5]:
```
def log10(x_):
    return sp.log(x_)/sp.log(10)

f = sp.S('0.82')
a = (f*S_ut)**2/S_e_prime
b = -sp.Rational(1, 3)*log10(f*S_ut/S_e_prime)

display(sp.Eq(sp.Symbol('a'), a.n()),
        sp.Eq(sp.Symbol('b'), b.n()))
```

```
sig_ar = 70
N = ((sig_ar/a)**(1/b)).n()
N  # cycles
```

$a = 161.376$

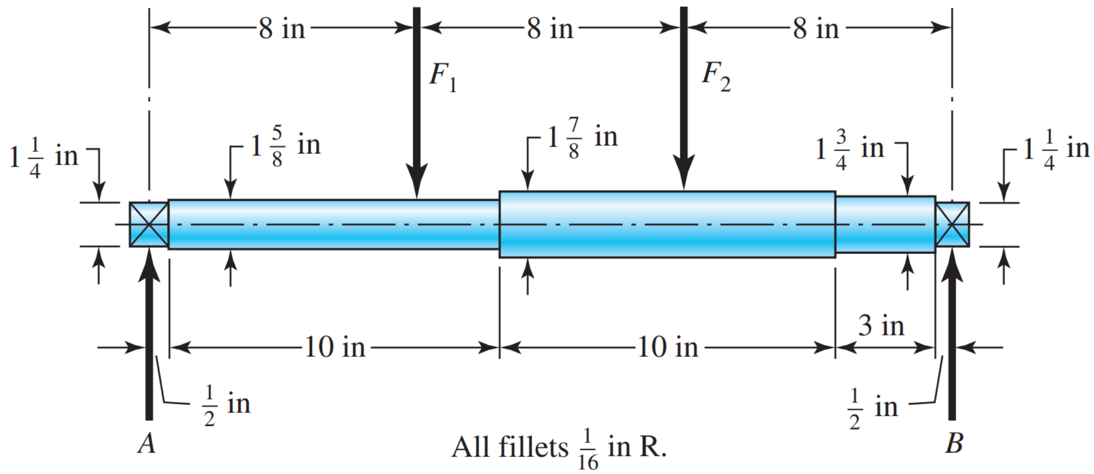
$b = -0.0716146160158993$

[5]:

116192.956004683

3 Problem 6-17

3.1 Given



The shaft shown in the figure above is machined from AISI 1040 CD steel. The shaft rotates at 1600 rpm and is supported in roller bearings at A and B. The applied forces are $F_1 = 2500\ lbf$ and $F_2 = 1000\ lbf$.

3.2 Find

Determine the minimum fatigue factor of safety based on achieving infinite life. If infinite life is not predicted, estimate the number of cycles to failure. Also check for yielding.

3.3 Solution

The reaction forces need to be solved first.

[6]:
```
A, B = sp.symbols('A B')
F1, F2 = 2500, 1000

eq1 = sp.Eq(A + B, F1 + F2)
eq2 = sp.Eq(B*24 - F1*8 - F2*16, 0)

sol = sp.solve([eq1, eq2], dict=True)[0]

display(eq1, eq2, Markdown('---'))
for key, value in sol.items():
    display(sp.Eq(key, value))
```

$A + B = 3500$

$24B - 36000 = 0$

$A = 2000$

$B = 1500$

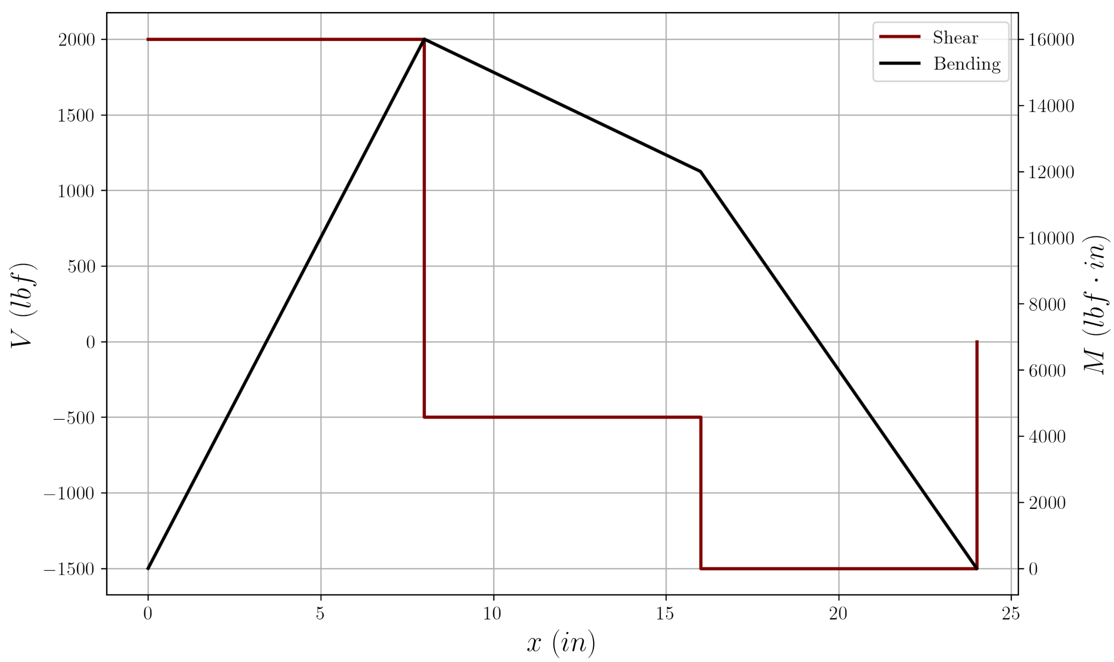
[7]:
```python
# Plotting Shear and Bending Moment Diagram
x = [0, 8, 16, 24]
x_shear = [0, 8, 8, 16, 16, 24, 24]
V1, V2, V3, V4 = [sol[A], sol[A] - F1, sol[A] - F1 - F2, sol[A] - F1 - F2 +
 ↪sol[B]]
V = [V1, V1, V2, V2, V3, V3, V4]
M = M1, M2, M3, M4 = [0, V1*8, V1*8 + V2*8, V1*8 + V2*8 + V3*8]

fig, ax = plt.subplots()
ax2 = ax.twinx()

ax.plot(x_shear, V, label='Shear')
ax2.plot(x, M, color='black', label='Bending')
ax2.grid(visible=False)

ax.legend(handles=[ax.lines[0], ax2.lines[0]])
ax.set_xlabel('$x$ ($in$)')
ax.set_ylabel('$V$ ($lbf$)')
ax2.set_ylabel(r'$M$ ($lbf\cdot in$)')

plt.show()
```



We are interested in the stress at the fillet radius in which the smaller diameter is used.

[8]:
```
M_mid = (M3 - M2)/8*(sp.S('10.5') - 8) + M2
M_mid  # in lbf*in
```

[8]:

14750.0

[9]:
```
c = sp.S('1.625')/2
I = sp.pi.n()/4*c**4
sig = M_mid*c/I
sig  # in psi
```

[9]:

35013.218176932

The yield strength is 71 ksi, and this stress is far below this value. The ultimate strength is $S_{ut} = 0.5(H_B) = 0.5(170) = 85$ ksi. To determine whether infinite life can be reached,

$$n_f = \frac{S_e}{K_f \sigma}$$

This is a variation of Eq. 6-42, but we are multiplying by the fatigue concentration factor to obtain the maximum stress value from the fillet geometry. From Eq. 6-32,

$$K_f = 1 + \frac{K_t - 1}{1 + \sqrt{a/r}}$$

A maybe calculated using Eq. 6-35. K_t comes from Figure A-15-9.

[10]:
```
r = sp.S('0.0625')
K_t = sp.S('1.95')
S_ut = sp.S(85)
a = (sp.S('0.246') - sp.S('3.08e-3')*S_ut + sp.S('1.51e-5')*S_ut**2 - sp.S('2.
↪67e-8')*S_ut**3)**2
Kf = 1 + (K_t - 1)/(1 + sp.sqrt(a/r))
Kf
```

[10]:

1.72652106649163

S_e maybe calculated using the same procedure as before.

[11]:
```
if S_ut <= 200:
    S_e_prime = 0.5*S_ut
else:
    S_e_prime = sp.S(100)

S_e_prime  # ksi
```

[11]:

42.5

[12]:
```python
a_factor, b_exponent = 2, sp.S('-0.217')  # Table 6-2
k_a = a_factor*S_ut**b_exponent
d = sp.S('1.625')
k_b = sp.S('0.879')*d**sp.S('-0.107')
S_e = k_a*k_b*S_e_prime
S_e  # ksi
```

[12]:

27.0497081578753

[13]:
```python
# Getting nf
S_e/(Kf*sig/1000)
```

[13]:

0.447464588712579

Because the factor of safety is less than one, infinite fatigue cannot be reached. There must be some finite number of cycles, N.

[14]:
```python
f = sp.S('0.867')

a = (f*S_ut)**2/S_e
b = -sp.Rational(1, 3)*log10(f*S_ut/S_e)

display(sp.Eq(sp.Symbol('a'), a.n()),
        sp.Eq(sp.Symbol('b'), b.n()))

N = ((sig/1000*Kf/a)**(1/b)).n()
N  # cycles
```

$a = 200.776769690168$

$b = -0.145091813123711$

[14]:

3917.08718671478

Important: The answer in the back of the book uses rounded values. For instance,

[15]:
```python
Kf = sp.S('1.72')
a, b = sp.S('200.78'), sp.S('-0.145')
sig = 35
(sig*Kf/a)**(1/b)
```

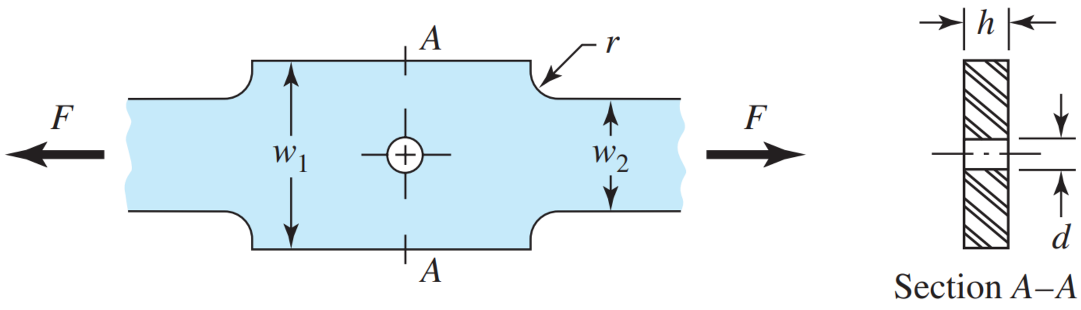
[15]:

4052.76515886349

This relationship is very sensitive.

4 Problem 6-30

4.1 Given



The figure above shows the free body diagram of a connecting link portion having stress concentration at three sections. The dimensions are $r = 0.25\ in$, $d = 0.40\ in$, $h = 0.50\ in$, $w_1 = 3.50\ in$, and $w_2 = 3.0\ in$. The forces F fluctuate between a tension of 5 $kips$ and a compression of 16 $kips$. Neglect column action.

4.2 Find

Find the least factor of safety if the material is cold drawn AISI 1018 steel.

[16]:
```python
Sy = 54  # ksi
Sut = 64  # ksi
r, d, h, w1, w2 = sp.S('0.25'), sp.S('0.4'), sp.S('0.5'), sp.S('3.5'), sp.S(3)
```

4.3 Solution

The least factor of safety comes from the stress concentration due to the diameter because its K_t value is greater than the K_t value from the fillets by observing the figures from Table A-15.

[17]:
```python
d/w1
```

[17]:

0.114285714285714

$K_t = 2.7$

[18]:
```python
# Find notch sensitivity
Kt = sp.S('2.7')
a = (sp.S('0.246') - sp.S('3.08e-3')*Sut + sp.S('1.51e-5')*Sut**2 - sp.S('2.
↪67e-8')*Sut**3)**2
q = 1/(1 + sp.sqrt(a/(d/2)))  # notch radius refers to hole radius
# q = sp.S('0.85')  # Solution's approximation
q
```

[18]:

0.811722489977041

[19]:
```
# Find Kf
Kf = 1 + q*(Kt - 1)
Kf
```

[19]: 2.37992823296097

[20]:
```
# Find Se prime
if Sut <= 200:
    S_e_prime = 0.5*sp.S(Sut)
else:
    S_e_prime = sp.S(100)

S_e_prime  # ksi
```

[20]: 32.0

[21]:
```
a_factor, b_exponent = 2, sp.S('-0.217')  # Table 6-2
k_a = a_factor*Sut**b_exponent
k_b = 1  # Eq. 6-20
k_c = sp.S('0.85')  # Eq. 6-25
S_e = k_a*k_b*k_c*S_e_prime
S_e  # ksi
```

[21]: 22.0626586316956

Use the following to obtain the factor of safety,

$$n_f = \frac{S_e}{K_f \sigma_a}$$

[22]:
```
sig_max = 5/(h*(w1 - d))
sig_min = -16/(h*(w1 - d))
sig_a = (sig_max - sig_min)/2
n_f = S_e/(Kf*sig_a)
n_f
```

[22]: 1.36847347329589

The answer in the back of the book is 1.33, but this comes from approximating that $q = 0.85$, which looks pretty conservative (see Figure 6-26).