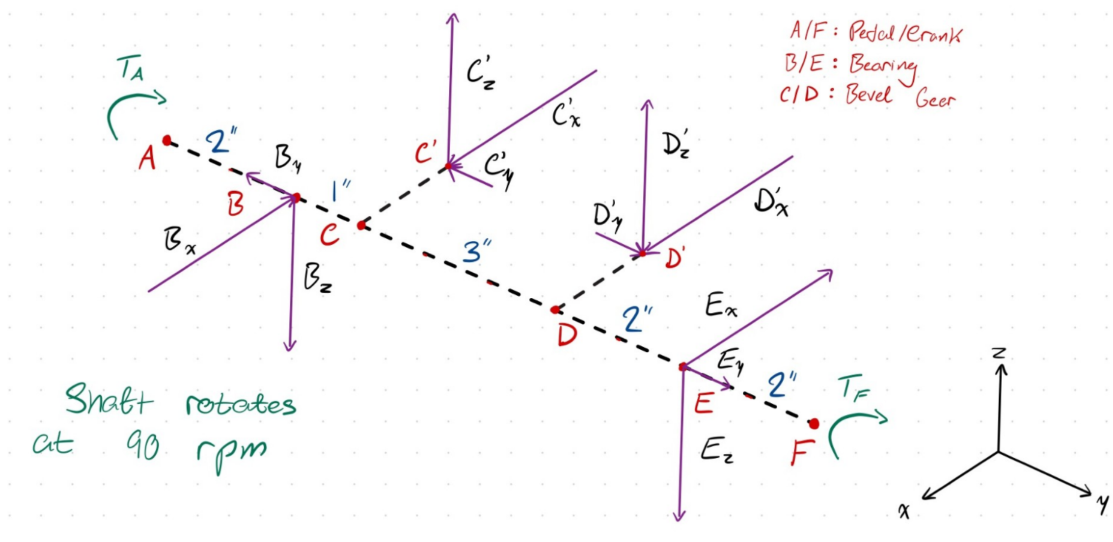
shaft_analysis

December 6, 2025

[18]:
```
import numpy as np
import matplotlib.pyplot as plt
```

1 FBD Analysis



1.1 Reaction Forces (lbf)

[19]:
```
Cx = 165.2 # Wr
Cy = 82.6 # Wx
Cz = 507.5 # Wt
Dx = 233.9 # Wr
Dy = 116.9 # Wx
Dz = 718.5 # Wt
```

1.2 Distances (inch)

[20]:
```
AB = 2
BC = 1
CD = 3
DE = 2
```

```
EF = 2
CC = 0.832 #R_mc
DD = 0.457 #R_md
```

1.3 Force and Moment Equations

Assume axial load is entirely on bearing B

[21]:
```
By = Cy - Dy #lbf
Ey = 0
```

$\Sigma F_x = 0 = -B_x + C_x + D_x - E_x$

$\Sigma F_z = 0 = -B_z + C_z + D_z - E_z$

Take moments about B

$\Sigma M_x = 0 = \vec{BC} \cdot C_z + \vec{BD} \cdot D_z + \vec{BE} \cdot E_z$

$\Sigma M_y = 0 = \vec{CC} \cdot C_z + \vec{DD} \cdot D_z + T_A - T_F$

$T_A = T_F$

$\Sigma M_z = 0 = \vec{BC} \cdot C_x + \vec{BD} \cdot D_x + \vec{BE} \cdot E_x$

[22]:
```
BD = (BC+CD)
BE = (BC+CD+DE)

A = np.array([
    [ 1,   0,   1,   0,   0,   0],
    [ 0,   1,   0,   1,   0,   0],
    [ 0,   0,   0,  BE,   0,   0],
    [ 0,   0,   0,   0,   1,  -1],
    [ 0,   0,   0,   0,   1,   1],
    [ 0,   0,  BE,   0,   0,   0],
], dtype=float)

b = np.array([
     (Cx + Dx),
     (Cz + Dz),
     (BC*Cz + BD*Dz),
    0,
     (CC*Cz + DD*Dz),
     (BC*Cx + BD*Dx),
], dtype=float)

sol = np.linalg.solve(A, b)

Bx, Bz, Ex, Ez, TA, TF = sol

print("Bx =", Bx, "lbf")
```

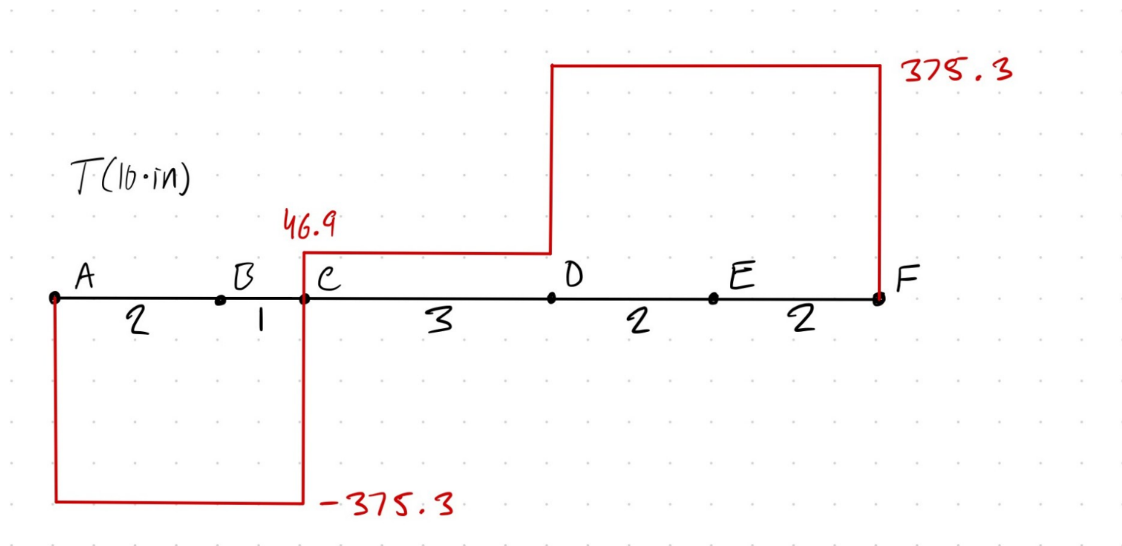
```python
print("Bz =", Bz, "lbf")
print("Ex =", Ex, "lbf")
print("Ez =", Ez, "lbf")
print("TA =", TA, "lbf-in")
print("TF =", TF, "lbf-in")
```

```
Bx = 215.63333333333335 lbf
Bz = 662.4166666666666 lbf
Ex = 183.46666666666667 lbf
Ez = 563.5833333333334 lbf
TA = 375.29724999999996 lbf-in
TF = 375.29724999999996 lbf-in
```

1.4 Torques (lbf-in)

[23]:
```python
TB = 0
TC = CC * Cz
TD = DD * Dz
TE = 0

Torque_AB = -TA
Torque_BC = Torque_AB + TB
Torque_CD = Torque_BC + TC
Torque_DE = Torque_CD + TD
Torque_EF = Torque_DE + TE

Torque_AB, Torque_BC, Torque_CD, Torque_DE, Torque_EF
```

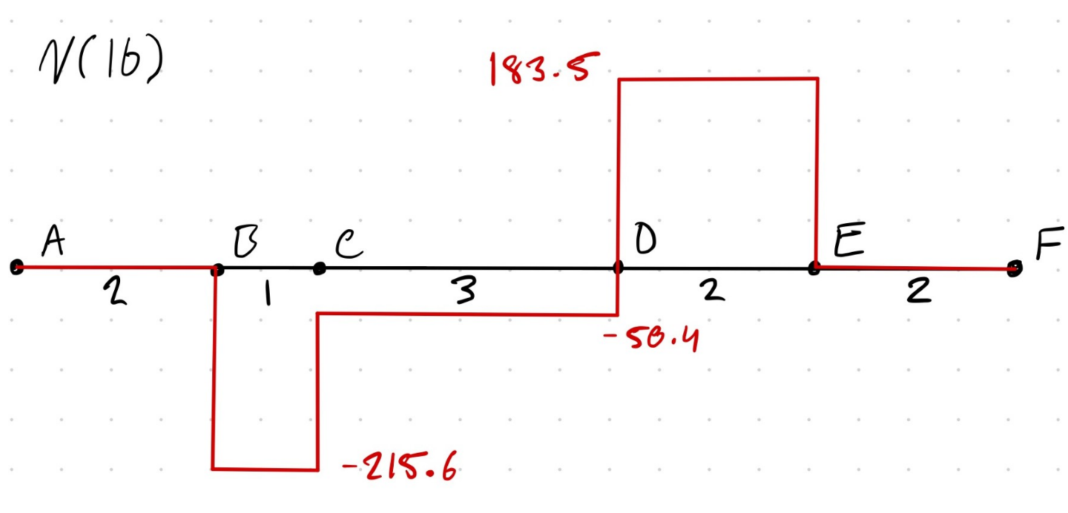
[23]:
```
(np.float64(-375.29724999999996),
 np.float64(-375.29724999999996),
 np.float64(46.94274999999999),
 np.float64(375.29725),
 np.float64(375.29725))
```



1.5 Shear and Moments in x

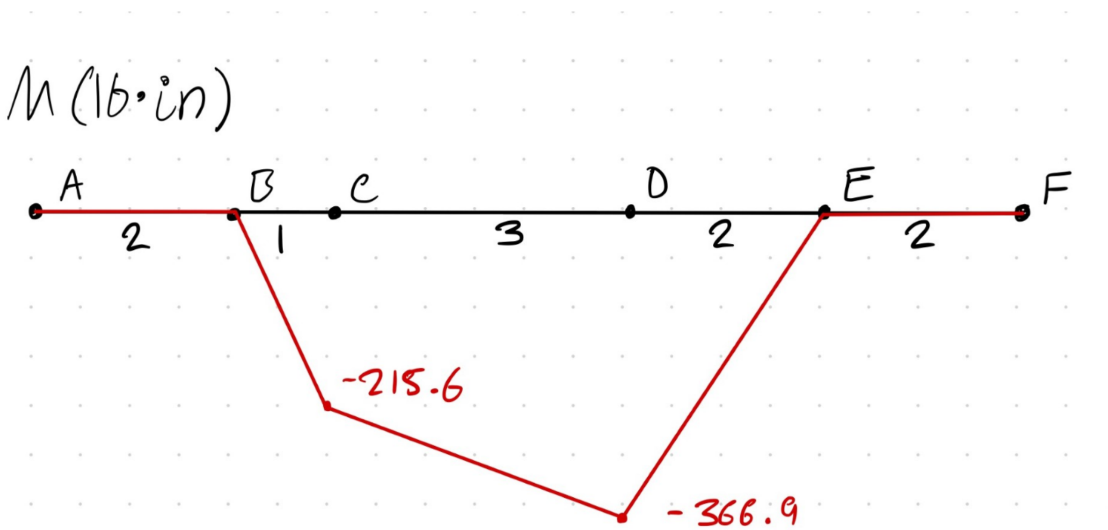
[24]:
```
t_abx = 0 # shear
t_bcx = - Bx
t_cdx = t_bcx + Cx
t_dex = t_cdx + Dx
t_efx = t_dex - Ex

t_abx, t_bcx, t_cdx, t_dex, t_efx
```

[24]:
```
(0,
 np.float64(-215.63333333333335),
 np.float64(-50.433333333333366),
 np.float64(183.46666666666664),
 np.float64(-2.842170943040401e-14))
```



[25]:
```
T_ax = 0 # moments

T_bx = T_ax + t_abx * AB

T_cx = T_bx + t_bcx * BC

T_dx = T_cx + t_cdx * CD

T_ex = T_dx + t_dex * DE

T_fx = T_ex + t_efx * EF

T_ax, T_bx, T_cx, T_dx, T_ex, T_fx
```

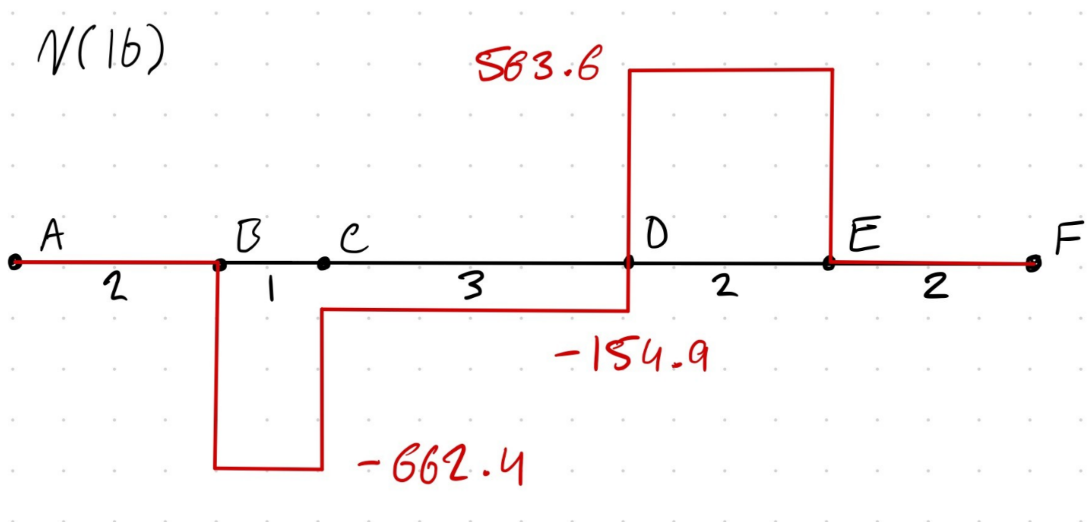
[25]:
```
(0,
 0,
 np.float64(-215.63333333333335),
 np.float64(-366.93333333333345),
 np.float64(-1.7053025658242404e-13),
 np.float64(-2.2737367544323206e-13))
```



1.6 Shear and Moments in z

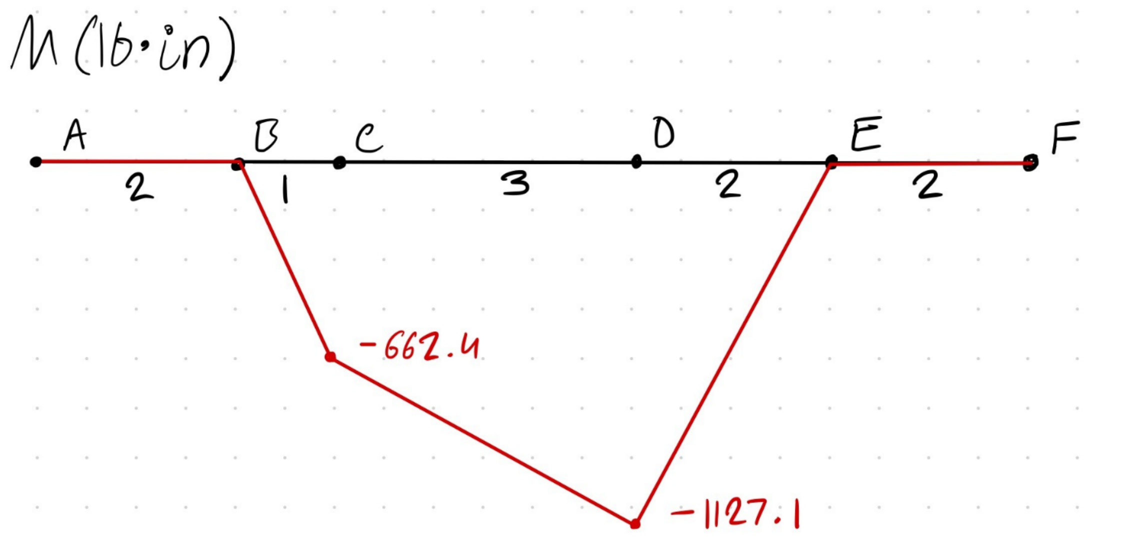
[26]:
```
t_abz = 0 # shear
t_bcz = - Bz
t_cdz = t_bcz + Cz
t_dez = t_cdz + Dz
t_efz = t_dez - Ez

t_abz, t_bcz, t_cdz, t_dez, t_efz
```

[26]: (0,
 np.float64(-662.4166666666666),
 np.float64(-154.91666666666663),
 np.float64(563.5833333333334),
 np.float64(0.0))



[27]:
```
T_az = 0 # moments
T_bz = T_az + t_abz * AB
T_cz = T_bz + t_bcz * BC
T_dz = T_cz + t_cdz * CD
T_ez = T_dz + t_dez * DE
T_fz = T_ez + t_efz * EF

T_az, T_bz, T_cz, T_dz, T_ez, T_fz
```

[27]: (0,
 0,
 np.float64(-662.4166666666666),
 np.float64(-1127.1666666666665),
 np.float64(2.2737367544323206e-13),
 np.float64(2.2737367544323206e-13))



1.7 Resultant Shear and Moments

[28]:
```
V_ab = np.linalg.norm(np.array([t_abx, t_abz]))
V_bc = np.linalg.norm(np.array([t_bcx, t_bcz]))
V_cd = np.linalg.norm(np.array([t_cdx, t_cdz]))
V_de = np.linalg.norm(np.array([t_dex, t_dez]))
V_ef = np.linalg.norm(np.array([t_efx, t_efz]))

V_ab, V_bc, V_cd, V_de, V_ef
```

[28]:
```
(np.float64(0.0),
 np.float64(696.6301563399493),
 np.float64(162.91928898145306),
 np.float64(592.6940116020145),
 np.float64(2.842170943040401e-14))
```

[29]:
```
M_a = np.linalg.norm(np.array([T_ax, T_az]))
M_b = np.linalg.norm(np.array([T_bx, T_bz]))
M_c = np.linalg.norm(np.array([T_cx, T_cz]))
M_d = np.linalg.norm(np.array([T_dx, T_dz]))
M_e = np.linalg.norm(np.array([T_ex, T_ez]))
M_f = np.linalg.norm(np.array([T_fx, T_fx]))

M_a, M_b, M_c, M_d, M_e, M_f
```

[29]:
```
(np.float64(0.0),
 np.float64(0.0),
 np.float64(696.6301563399493),
 np.float64(1185.3880232040287),
 np.float64(2.8421709430404007e-13),
```

```
np.float64(3.2155493553843715e-13))
```

1.8 Shaft Diameter Determination

The shaft will be made of a martensitic steel due to the marine application. We select SAE 416 Q&T 1400 from table A-6 in Mott;

$S_u = 90$ ksi

$S_y = 60$ ksi

$S_n = 35$ ksi, assuming machined/cold drawn, F-5 Mott

Assume wrought steel; $C_m = 1$

Only working with bending stress; $C_{st} = 1$

Assume 99% reliability; $C_R = 0.81$

Will guess 2" shaft; $C_s = 0.8$

All elements are held by retaining rings from both sides; $K_t = 3$

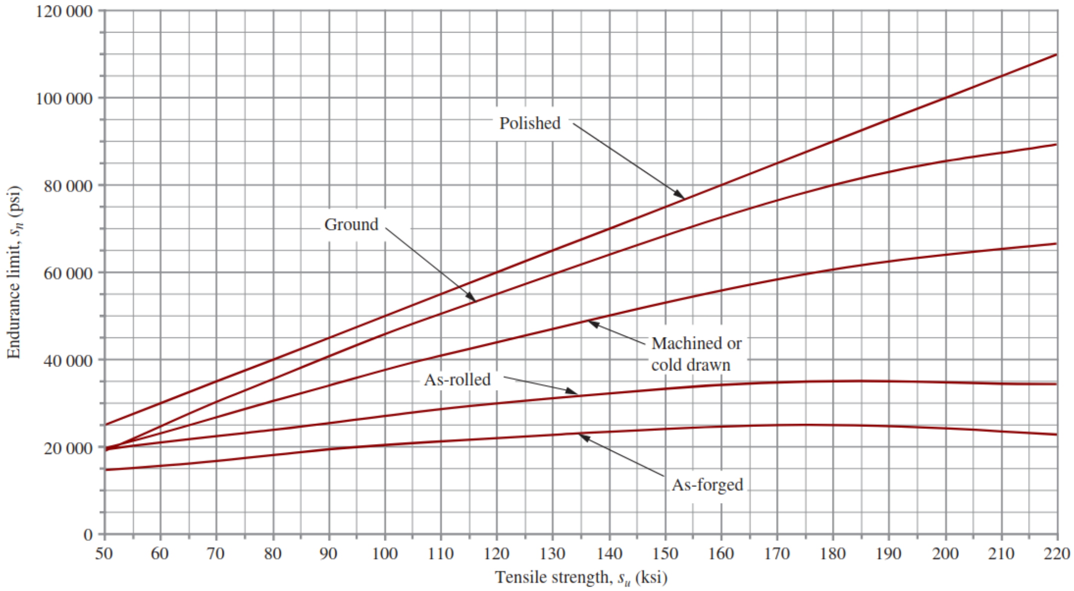
Assume a saftey factor; $N = 3$

Assume torque/moments dominate;

$$D = \left(\frac{32N}{\pi}\sqrt{\left(\frac{K_t M}{S_n'}\right)^2 + \frac{3}{4}\left(\frac{T}{S_y}\right)^2}\right)^{\frac{1}{3}}$$

APPENDIX 6 Properties of Stainless Steels

Material designation		Condition	Tensile strength		Yield strength		Ductility (percent elongation in 2 in)
SAE number	UNS		(ksi)	(MPa)	(ksi)	(MPa)	
Austenitic steels							
201	S20100	Annealed	115	793	55	379	55
		1/4 hard	125	862	75	517	20
		1/2 hard	150	1030	110	758	10
		3/4 hard	175	1210	135	931	5
		Full hard	185	1280	140	966	4
301	S30100	Annealed	110	758	40	276	60
		1/4 hard	125	862	75	517	25
		1/2 hard	150	1030	110	758	15
		3/4 hard	175	1210	135	931	12
		Full hard	185	1280	140	966	8
304	S30400	Annealed	85	586	35	241	60
310	S31000	Annealed	95	655	45	310	45
316	S31600	Annealed	80	552	30	207	60
Ferritic steels							
405	S40500	Annealed	70	483	40	276	30
430	S43000	Annealed	75	517	40	276	30
		Full hard	90	621	80	552	15
446	S44600	Annealed	80	552	50	345	25
Martensitic steels							
410	S41000	Annealed	75	517	40	276	30
416	S41600	Q&T 600	180	1240	140	966	15
		Q&T 1000	145	1000	115	793	20
		Q&T 1400	90	621	60	414	30
431	S43100	Q&T 600	195	1344	150	1034	15
440A	S44002	Q&T 600	280	1930	270	1860	3
501	S50100	Annealed	70	483	30	207	28
		OQT 1000	175	1210	135	931	15
Precipitation-hardening steels							
17-4PH	S17400	H 900	210	1450	185	1280	14
		H 1150	145	1000	125	862	19
17-7PH	S17700	RH 950	200	1380	175	1210	10
		TH 1050	175	1210	155	1070	12
PH 13-8 Mo	S13800	H 950	220	1517	205	1413	10
		H 1050	175	1207	165	1138	12
		H 1150	135	931	90	621	14



(a) U.S. customary units

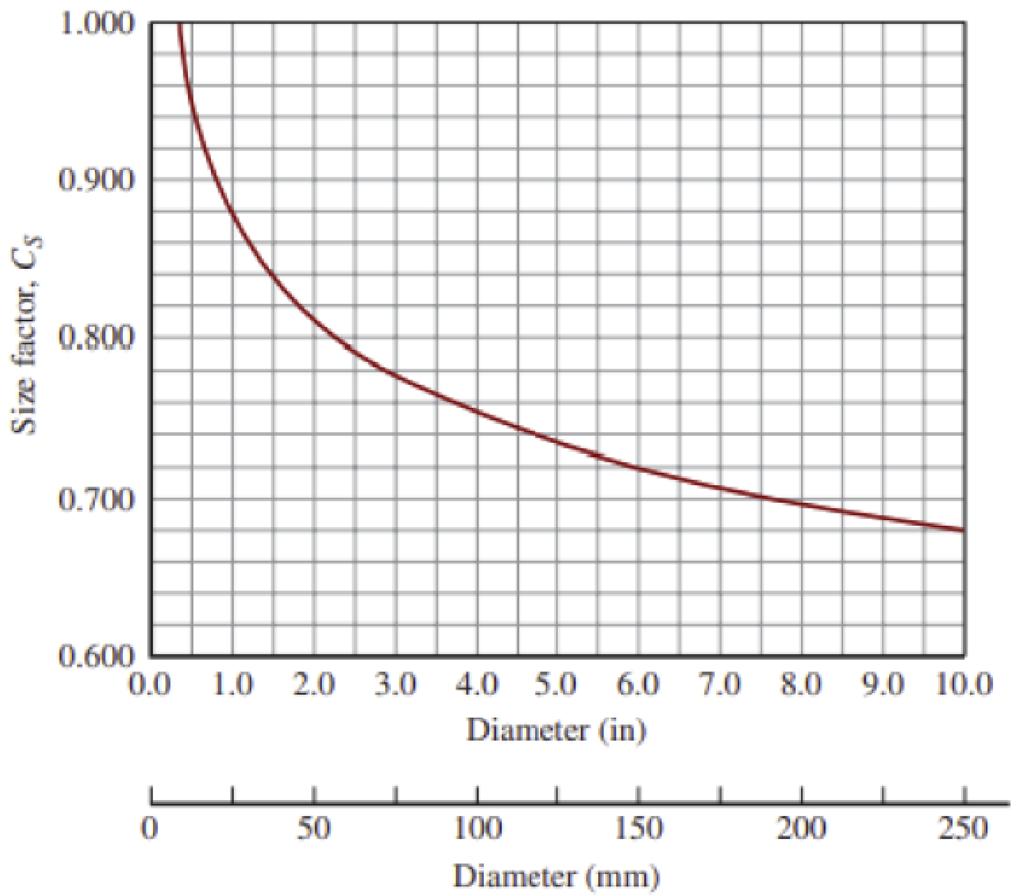
[30]:
```
Su = 87 * 10**3
Sy = 60 * 10**3
Sn = 33 * 10**3
Cm = 1
Cst = 1
Cr = 0.81
Cs = 0.8
Kt = 3
N=3
Sn_p = Sn * Cm * Cst * Cr * Cs
```

[31]:
```
def shaft_diameter(M, T):
    term1 = (Kt * M) / Sn_p
    term2 = (T / Sy)

    inside_sqrt = term1**2 + 0.75 * term2**2
    inside_cubic = (32 * N / np.pi) * np.sqrt(inside_sqrt)

    D = inside_cubic ** (1/3)
    return D

D_ab1 = shaft_diameter(M_a, Torque_AB)
D_ab2 = shaft_diameter(M_b, Torque_AB)
D_ab = max(D_ab1, D_ab2)

D_bc1 = shaft_diameter(M_b, Torque_BC)
D_bc2 = shaft_diameter(M_c, Torque_BC)
D_bc = max(D_bc1, D_bc2)
```

```
D_cd1 = shaft_diameter(M_c, Torque_CD)
D_cd2 = shaft_diameter(M_d, Torque_CD)
D_cd = max(D_cd1, D_cd2)

D_de1 = shaft_diameter(M_d, Torque_DE)
D_de2 = shaft_diameter(M_e, Torque_DE)
D_de = max(D_de1, D_de2)

D_ef1 = shaft_diameter(M_f, Torque_EF)
D_ef2 = shaft_diameter(M_f, Torque_EF)
D_ef = max(D_ef1, D_ef2)

D = max(D_ab, D_bc, D_cd, D_de, D_ef)
D = D * 1.06
D
```

[31]: np.float64(1.822723076069883)

The following figure from shigley is used to adjust our C_s factor:



[32]:
```python
Cs = 0.82 # Adjusted for approximation to 1.75"
Sn_p = Sn * Cm * Cst * Cr * Cs
D_ab1 = shaft_diameter(M_a, Torque_AB)
D_ab2 = shaft_diameter(M_b, Torque_AB)
D_ab = max(D_ab1, D_ab2)

D_bc1 = shaft_diameter(M_b, Torque_BC)
D_bc2 = shaft_diameter(M_c, Torque_BC)
D_bc = max(D_bc1, D_bc2)

D_cd1 = shaft_diameter(M_c, Torque_CD)
D_cd2 = shaft_diameter(M_d, Torque_CD)
D_cd = max(D_cd1, D_cd2)

D_de1 = shaft_diameter(M_d, Torque_DE)
D_de2 = shaft_diameter(M_e, Torque_DE)
D_de = max(D_de1, D_de2)

D_ef1 = shaft_diameter(M_f, Torque_EF)
D_ef2 = shaft_diameter(M_f, Torque_EF)
D_ef = max(D_ef1, D_ef2)

D = max(D_ab, D_bc, D_cd, D_de, D_ef)
D = D * 1.06

NFS_SHAFT = ((1.75/1.06)**3 * np.pi) / (32 * np.sqrt((Kt * M_d / Sn_p)**2 + 0.
↪75 * (Torque_DE / Sy)**2))

D, NFS_SHAFT
```

[32]:
```
(np.float64(1.8077982173901768), np.float64(2.7213571321704997))
```

We select a 1.75” shaft made of SAE 416 Q&T 1400, to obtain a safety factor of ~2.72

1.9 Keys

Both gears are bound to shaft with keys. We select a 0.5” square key shaft from table 11-1 in Mott since nominal shaft diameter falls between; 1.75” - 2.25”. We select the keys to be made of SAE 1018; $S_y = 54$ ksi from table 11-4 of Mott.

[33]:
```python
D = 1.0
T_max = max(Torque_AB, Torque_BC, Torque_CD, Torque_DE, Torque_EF)
Sy_key = 54000
W = 0.5

L_min = (4 * T_max * N) / (D * W * Sy_key)

L_key = 0.25
```

```
NFS_key = (L_key * D * W * Sy_key) / (4 * T_max)

L_min, NFS_key
```

[33]: (np.float64(0.16679877777777777), np.float64(4.496435825202556))

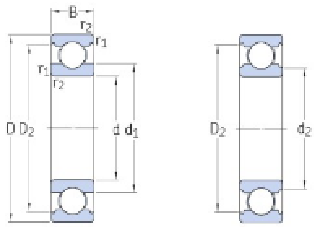
We hence select a 0.5” square key made of SAE 1018, which is 0.25” long to ultimately yield a safety factor of ~4.5.

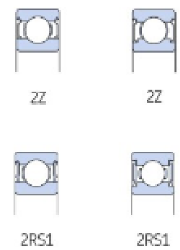
1.10 Bearings

We will use Deep Groove Ball Bearings, just considering the dynamic loads at B and E slightly different (since B also experiences a thrust load). We select consider SKF parameters, and Stainless Steel bearings as we wish for our bearings to be non-corrosive (since they are in water). We are confined to selecting bearings with inner bore diameter of 1.75”.

1.4 Stainless steel deep groove ball bearings
d 30 – 50 mm

1.4





Principal dimensions			Basic load ratings		Fatigue load limit	Speed ratings		Mass	Designation
			dynamic	static		Reference speed	Limiting speed		
d	D	B	C	C_0	P_u				
mm			kN		kN	r/min		g	–
30	72	19	22,9	15	0,64	–	6 300	346	▸ W 6306-2RS1
cont.	72	19	22,9	15	0,64	22 000	11 000	345	W 6306-2Z
	72	19	22,9	15	0,64	22 000	14 000	331	W 6306
35	47	7	3,71	3,35	0,14	–	8 500	29,5	W 61807-2RS1
	55	10	9,36	7,65	0,325	–	7 500	73,5	W 61907-2RS1
	62	14	13,8	10,2	0,44	–	6 700	147	▸ W 6007-2RS1
	62	14	13,8	10,2	0,44	24 000	12 000	148	W 6007-2Z
	62	14	13,8	10,2	0,44	24 000	15 000	138	W 6007
	72	17	22,1	15,3	0,655	–	6 000	276	▸ W 6207-2RS1
	72	17	22,1	15,3	0,655	22 000	11 000	277	W 6207-2Z
	72	17	22,1	15,3	0,655	22 000	14 000	262	W 6207
	80	21	28,6	19	0,815	–	5 600	441	W 6307-2RS1
40	62	12	11,9	9,8	0,425	–	6 700	107	W 61908-2RS1
	68	15	14,6	11,4	0,49	–	6 300	182	▸ W 6008-2RS1
	68	15	14,6	11,4	0,49	22 000	11 000	183	▸ W 6008-2Z
	68	15	14,6	11,4	0,49	22 000	14 000	172	W 6008
	80	18	25,1	17,6	0,75	–	5 600	359	▸ W 6208-2RS1
	80	18	25,1	17,6	0,75	20 000	10 000	359	▸ W 6208-2Z
	80	18	25,1	17,6	0,75	20 000	12 000	342	W 6208
45	68	12	12,1	10,8	0,465	–	6 000	125	▸ W 61909-2RS1
	75	16	18,2	15	0,64	–	5 600	236	▸ W 6009-2RS1
	75	16	18,2	15	0,64	20 000	10 000	237	W 6009-2Z
	85	19	28,1	20,4	0,865	–	5 000	395	▸ W 6209-2RS1
	85	19	28,1	20,4	0,865	18 000	9 000	394	W 6209-2Z
50	65	7	5,07	5,5	0,236	–	6 000	51	W 61810-2RS1
	80	16	19	16,6	0,71	–	5 000	256	▸ W 6010-2RS1
	80	16	19	16,6	0,71	18 000	9 000	256	W 6010-2Z
	90	20	30,2	23,2	0,98	–	4 800	449	▸ W 6210-2RS1
	90	20	30,2	23,2	0,98	17 000	8 500	453	W 6210-2Z

▸ Popular item

326 SKF

Per the above table, we select the W 6209-2RS1; this is a design listed as popular, hence it is easier to obtain and likely a little cheaper.

Tolerance considerations are implicit here - a housing is not required here as the intention is to press fit the bearing into the housing of the gear box.

To then ensure the bearing is adequate, we determine the dynamic load at B and E, and compare it to that allowable for the bearing. We will also obtain our safety factor corresponding to the bearings at each location via this analysis as well.

Note that we assume a combined reliability of 99% and are using SKF Weibull Parameters (since we are using an SKF bearing). We additionally select an application factor of 1.5 per Table 11-5 of Shigley, for a light to moderate impact loading. We also say the design life is 4000 hours per table 11-4 of Shigley (for intermittent use).

Table 11–4 Bearing-Life Recommendations for Various Classes of Machinery

Type of Application	Life, kh
Instruments and apparatus for infrequent use	Up to 0.5
Aircraft engines	0.5–2
Machines for short or intermittent operation where service interruption is of minor importance	4–8
Machines for intermittent service where reliable operation is of great importance	8–14
Machines for 8-h service that are not always fully utilized	14–20
Machines for 8-h service that are fully utilized	20–30
Machines for continuous 24-h service	50–60
Machines for continuous 24-h service where reliability is of extreme importance	100–200

Table 11–5 Load-Application Factors

Type of Application	Load Factor
Precision gearing	1.0–1.1
Commercial gearing	1.1–1.3
Applications with poor bearing seals	1.2
Machinery with no impact	1.0–1.2
Machinery with light impact	1.2–1.5
Machinery with moderate impact	1.5–3.0

Table 11–1 Equivalent Radial Load Factors for Ball Bearings

		$F_a/(VF_r) \le e$		$F_a/(VF_r) > e$	
F_a/C_0	e	X_1	Y_1	X_2	Y_2
0.014*	0.19	1.00	0	0.56	2.30
0.021	0.21	1.00	0	0.56	2.15
0.028	0.22	1.00	0	0.56	1.99
0.042	0.24	1.00	0	0.56	1.85
0.056	0.26	1.00	0	0.56	1.71
0.070	0.27	1.00	0	0.56	1.63
0.084	0.28	1.00	0	0.56	1.55
0.110	0.30	1.00	0	0.56	1.45
0.17	0.34	1.00	0	0.56	1.31
0.28	0.38	1.00	0	0.56	1.15
0.42	0.42	1.00	0	0.56	1.04
0.56	0.44	1.00	0	0.56	1.00

*Use 0.014 if $F_a/C_0 < 0.014$.

[34]:

```
C10 = 6317.131 #lbf (28.1kN) from SKF catalogue for W 6009-2RS1 - dynamic load␣
↪rating
C0 = 4586.102 #lbf (20.4kN) from SKF catalogue for W 6009-2RS1 - static load␣
↪rating

R = np.sqrt(0.99)
n = 90 # rotation speed in rpm
af = 1.5 # application factor
L_D = 4000 # life in hours

# SKF Weibull Parameters
L_10 = 10**6
x_0 = 0.02
theta = 4.459
b = 1.483

x_D = (60 * L_D * n) / L_10

a = 3 # for ball bearings

# For Bearing B
V = 1 # inner ring rotates
```

```python
AxialDynamicRatio = abs(By / C0) # about 0.005
e = 0.19 # by footnote to Table 11-1 of Shigley
AxialRadialRatio = abs(By/np.sqrt(Bx**2 + Bz**2)) # about 0.05

# e > AxialRadialRatio - the axial load is effectively neglected
X = 1
Y = 0

FB_e = X * V * np.sqrt(Bx**2 + Bz**2)

# Implicitly assume load factor of 1

C10_B = FB_e * af * (x_D/(x_0 + (theta - x_0) * (np.log(1/R))**(1/b)))**(1/a)
NFS_B = C10 / C10_B

# For Bearing E
V = 1 # inner ring rotates
FE_e = X * V * np.sqrt(Ex**2 + Ez**2)
C10_E = FE_e * af * (x_D/(x_0 + (theta - x_0) * (np.log(1/R))**(1/b)))**(1/a)
NFS_E = C10 / C10_E

C10_B, NFS_B, C10_E, NFS_E
```

[34]:
```
(np.float64(5538.425114566881),
 np.float64(1.14060059842373),
 np.float64(4712.100630780212),
 np.float64(1.3406188651268327))
```

We hence have that the bearing at B has a safety factor of ~1.14, and the one at E has a safety factor of ~1.34.

1.11 Cost Analysis

All costs are found on McMaster Carr as they are easily found and clearly listed there - the SKF brand bearings and housings would likely be in a lower price range nonetheless.

Shaft costs $97.58 (https://www.mcmaster.com/products/316-stainless-steel/highly-corrosion-resistant-316-stainless-steel-rods~~/?s=316-stainless-steel)

Bearing costs $156.94 x 2 (https://www.mcmaster.com/products/stainless-steel-bearings/ball-bearings-1~/?s=stainless-steel-bearings)

Cost of keys and retaining rings assumed to be negligible (easily obtained/manufactured via UBC machine shop and other sources)

1.12 Safety Factors of Existing Design

[35]:
```
# Assume same material, just different diameter
D_old = 0.625

NFS_shaft_old = ((D_old/1.06)**3 * np.pi) / (32 * np.sqrt((Kt * M_d / Sn_p)**2
 + 0.75 * (Torque_CD / Sy)**2))

# Assume a worst case for stainless steel bearing of old inner bore size
 (W61802-2RS1)
C10_old_bearing = 370.9

NFS_old_bearing = C10_old_bearing / C10_E

NFS_shaft_old, NFS_old_bearing
```

[35]:
```
(np.float64(0.12403652340184368), np.float64(0.07871224090105812))
```

Old designs appear to be greatly under-speced; the existing system has likely not yet failed because of how little it has been used, and the less than extreme treatment it has likely been subjected to (we assessed for a performance case, and not a recreational case).