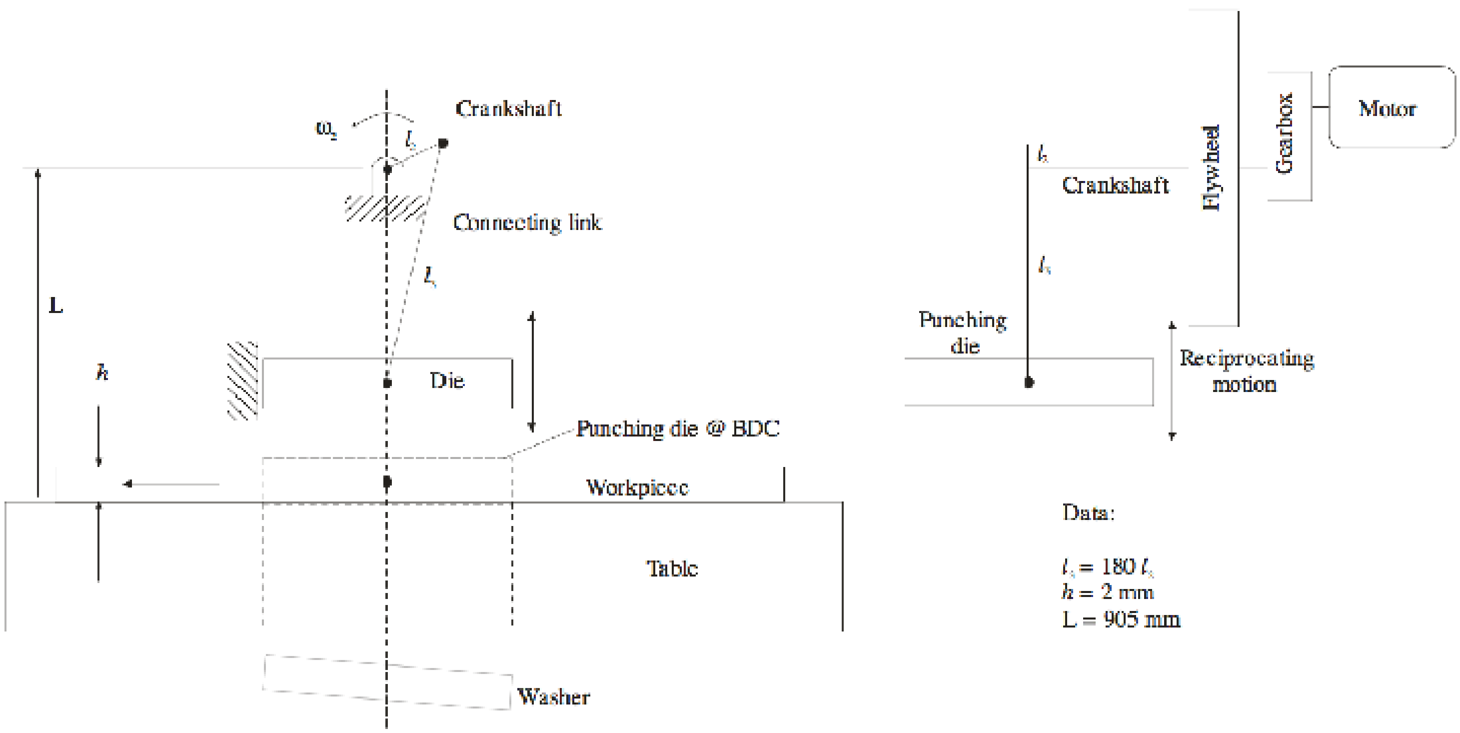
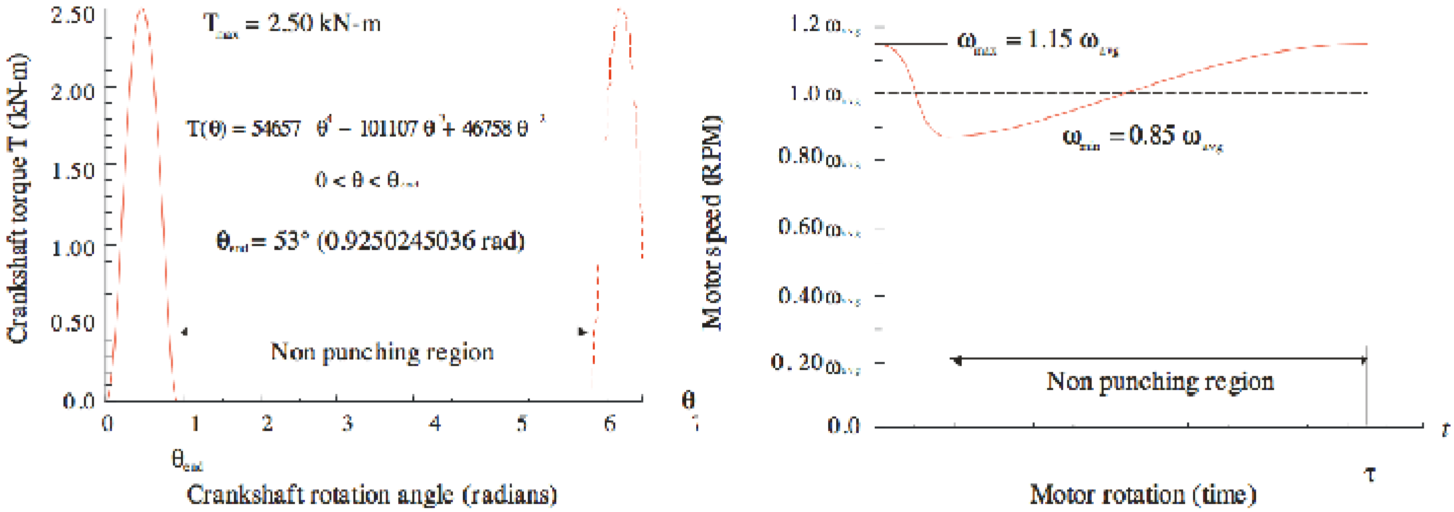
INME 4012 - Project

Machine Design

Scenario:

A punch press is used to stamp circular steel washers from a workpiece. A schematic of the washer producing device is shown below. A flywheel is directly coupled to a crankshaft. An electric motor drives the crankshaft via a gear reducer. As the motor rotates, a punching die reciprocates producing washers each revolution of the crankshaft. The purpose of the flywheel is to reduce the size of the motor and gearbox necessary to produce these washers. The optimal motor speed is 1000-1,100 RPM. The crankshaft diameter is 55 mm and fabricated from ASTM 1018 annealed steel. In order to meet production demands, 100 disks are produced each minute. The crankshaft torque necessary to stamp each washer is shown in a separate figure below. During “punching”, the flywheel speed is reduced and the energy to reduce the flywheel speed is used to “help” produce the washers. The motor increases the speed of the flywheel during the non-punching region. Also shown below is the flywheel speed variation.





The following information is given:

```
crankshaftDiameter = 55; % mm
l2 = crankshaftDiameter;
h = 2; % mm
l3 = 180*l2;
L = 905; % mm
```

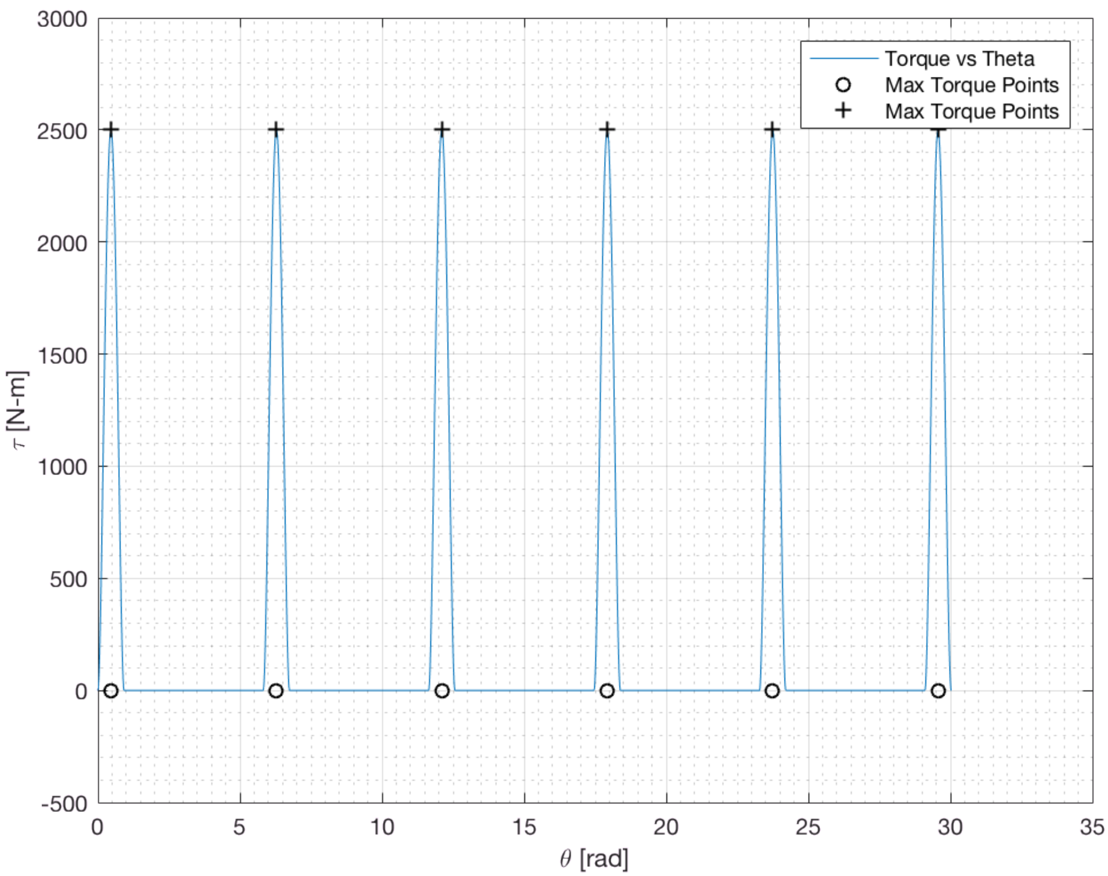
The torque can be computed as:

```
thetaEnd = 53; %deg
fracPress = round(10000*(thetaEnd/360));
fracNoPress = round(10000*((360-thetaEnd)/360));
thetaPress = linspace(0,0.9250245036,fracPress);
torqueEq = @(thetaVar) 54657.*(thetaVar.^4)-101107*(thetaVar.^3)+46758*(thetaVar.^2);
torquePress = torqueEq(thetaPress);
SecondPress = 360-0.5*thetaEnd;
SecondPress = SecondPress*(pi/180);
thetaNoPress = linspace(0.9250245036,SecondPress,fracNoPress+1);
thetaPress2 = linspace(SecondPress,SecondPress+0.9250245036,fracPress);
thetaNoPress2 = linspace(SecondPress+0.9250245036,2*SecondPress,fracNoPress+1);
thetaPress3 = linspace(2*SecondPress,2*SecondPress+0.9250245036,fracPress);
thetaNoPress3 = linspace(2*SecondPress+0.9250245036,3*SecondPress,fracNoPress+1);
thetaPress4 = linspace(3*SecondPress,3*SecondPress+0.9250245036,fracPress);
thetaNoPress4 = linspace(3*SecondPress+0.9250245036,4*SecondPress,fracNoPress+1);
thetaPress5 = linspace(4*SecondPress,4*SecondPress+0.9250245036,fracPress);
thetaNoPress5 = linspace(4*SecondPress+0.9250245036,5*SecondPress,fracNoPress+1);
thetaPress6 = linspace(5*SecondPress,5*SecondPress+0.9250245036,fracPress);
```

The torque behaves as:

```
theta = [thetaPress,thetaNoPress,thetaPress2,thetaNoPress2,...
    thetaPress3,thetaNoPress3,thetaPress4,thetaNoPress4,thetaPress5,thetaNoPress5,thetaPress6]
torque = [torquePress,zeros(1,numel(thetaNoPress)),torquePress,...
    zeros(1,numel(thetaNoPress2)),torquePress,zeros(1,numel(thetaNoPress3)),...
    torquePress,zeros(1,numel(thetaNoPress2)),torquePress,zeros(1,numel(thetaNoPress2)),torque
figure('Name','Torque vs Theta')
plot(theta,torque); hold on
```

```
scatter([(0.5*0.9250245036),2*pi,4*pi-(0.5*0.9250245036),...
    6*pi-2*(0.5*0.9250245036),8*pi-3*(0.5*0.9250245036),...
    10*pi-4*(0.5*0.9250245036)],[0,0,0,0,0,0],'ko')
scatter([(0.5*0.9250245036),2*pi,4*pi-(0.5*0.9250245036),...
    6*pi-2*(0.5*0.9250245036),8*pi-3*(0.5*0.9250245036),...
    10*pi-4*(0.5*0.9250245036)],[2500,2500,2500,2500,2500,2500],'k+')

legend('Torque vs Theta','Max Torque Points','Max Torque Points')
xlabel('\theta [rad]')
ylabel('\tau [N-m]')
grid on
grid minor
```



Attempting to determine the punch press' speed, we determine how many radians are required per spike. From this estimate, whose accuracy will increase with more spikes as a cummulative error exists. From the problem statement, we know that 100 disks are required per minute.

The central point of any spike beyond the third spike can be determined using the following equation:

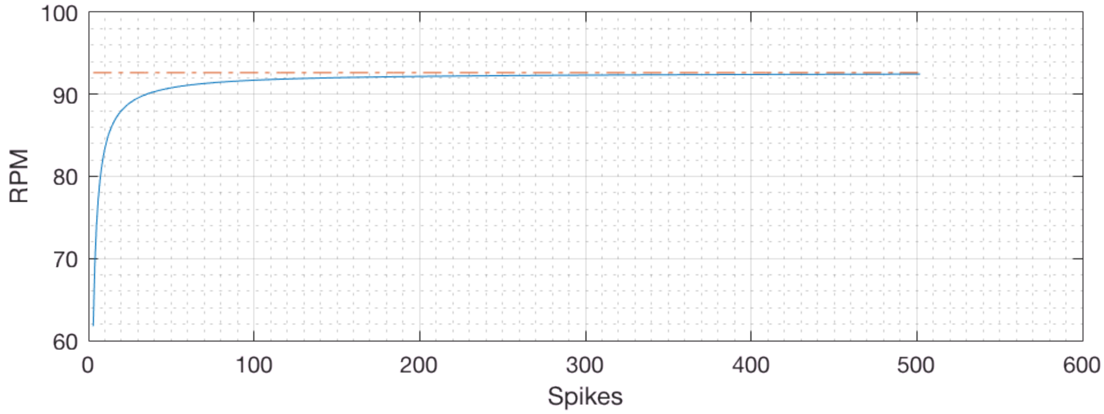
$$T_{max_{location}} = 2(n-1)\pi - (n-2)\left(\frac{53\pi}{360}\right)$$

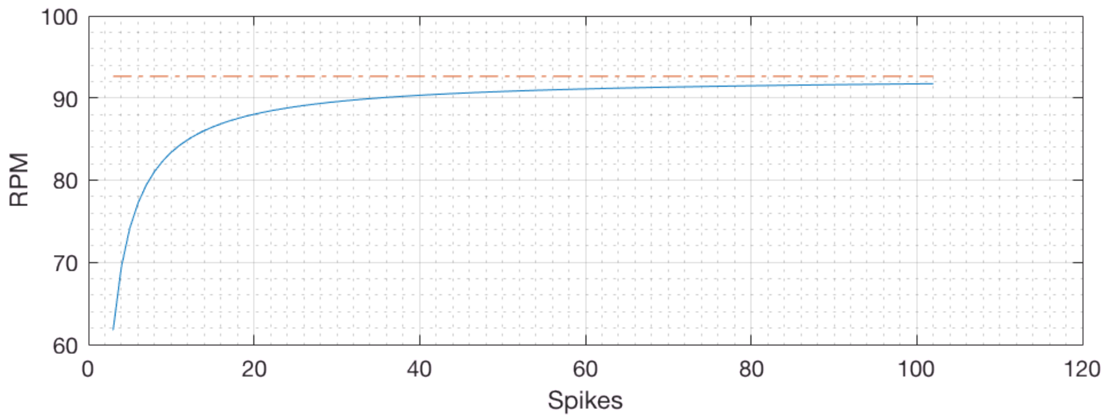
The rightmost point can be easily determined by adjusting the second term in the previous equation:

$$T_{max_{location}} = 2\pi(n-1) - (n-1)\left(\frac{53\pi}{360}\right)$$
$$T_{max_{location}} = \left(2\pi - \left(\frac{53\pi}{360}\right)\right)(n-1)$$
$$T_{max_{location}} \approx 5.8207(n-1)$$

From this equation we determine the 100th spike location to be:

```
spikes = [3:10000000];
% Here, the full expression is encoded to facilitate code maintenance.
EstimatedRadians = (2.*(spikes-1).*pi)-((spikes-1).*(0.5*0.9250245036));
EstimatedRevs = EstimatedRadians./(2*pi); % revs
RevsPerSpike = EstimatedRevs./spikes; % revs/spikes
ReqDisks = 100;
RequiredRPMs = RevsPerSpike.*ReqDisks;
figure('Name','EstimatedRevs vs Terms Used')
subplot(2,1,1)
plot(spikes(1:500),RequiredRPMs(1:500)); hold on
plot(spikes(1:500),max(RequiredRPMs).*ones(1,numel(spikes(1:500))),'-.')
xlabel('Spikes')
ylabel('RPM')
grid on
grid minor
subplot(2,1,2)
plot(spikes(1:100),RequiredRPMs(1:100)); hold on
plot(spikes(1:100),max(RequiredRPMs).*ones(1,numel(spikes(1:100))),'-.')
xlabel('Spikes')
ylabel('RPM')
grid on
grid minor
```





As seen from the previous figure, the log-like trend tends asymptotically to a certain limit. However, using 100 spikes yields a reasonable approximation. Given the power of modern computing hardware, we will use a 1M spikes for a smooth approximation.

```
RequiredRPMs = max(RequiredRPMs)
```

```
RequiredRPMs = 92.6389
```

The required RPM for the punch press' motor are much higher than the one required for the actual pressing mechanism.

```
MotorRPMs = 1020; % Yields a value near integer for the reduction
GearboxReduction = round(MotorRPMs/RequiredRPMs)
```

```
GearboxReduction = 11
```

Three reductions will be used:

First Reduction -> 1:2

Second Reduction -> 1:2

Third Reduction -> 1:2.75

Overall GearBox Reduction -> 1:11

The reductions are named such that the third reduction is the largest and connected to the motor.

```
GearNo = 6;
FirstReduction = 2;
SecondReduction = 2;
ThirdReduction = 2.75;
```

Building the gearbox

The proposed gearbox has 3 gear pairs. The governing equations are:

$$\frac{N_2}{N_1} = 2$$
$$\frac{N_4}{N_3} = 2$$
$$\frac{N_6}{N_5} = 2.75$$

$$N_1 + N_2 = N_3 + N_4$$
$$N_3 + N_4 = N_5 + N_6$$
$$N_1 + N_2 = N_5 + N_6$$

From these equations and some algebraic manipulation,

$$N_2 = 2N_1$$
$$N_4 = 2N_3$$
$$N_6 = 2.75N_5$$

$$N_1 = N_3$$
$$N_3 = 1.25N_5$$

We'll leave these expressions momentarily and move to compute the maximum and minimum angular speeds according to the problem statement. Further, we will compute the required energy (ie work) and consequently the FlyWheel's Inertia.

$$\omega_{ave} = 92.64 \ \ RPM \ = 9.7 \frac{rad}{s}$$
$$\omega_{max} = 1.15\omega_{ave} = 11.155 \frac{rad}{s}$$
$$\omega_{min} = 0.85\omega_{ave} = 8.245 \frac{rad}{s}$$

The average torque over 2π radians will be used in determining the power supplied to the flywheel along the punching cycle. The average torque can be determined as,

$$\bar{\tau} = \frac{trapz(\theta_{press}, \tau_{press})}{\theta_{max}}$$

```
averageTorque = trapz(thetaPress,torquePress)/(2*pi);
averageTorqueArray = zeros(1,numel(thetaPress)+2);
averageTorqueArray(2:numel(thetaPress)) = averageTorque;
disp(['Average Torque: ', num2str(averageTorque),' N-m'])
```

```
Average Torque: 196.2748 N-m
```

The *necessary energy* to be provided on average by the motor over the entirety of the punch cycle can now be determined as,

```
cycleX = [thetaPress,thetaNoPress];
cycleY = [torquePress,zeros(1,numel(thetaNoPress))];
work = trapz(cycleX,cycleY); % Joules
clearvars cycleX cycleY
fprintf('Work to Press: %10.2f J',work)
```

```
Work to Press:    1233.23 J
```

```
fprintf('Work to Press: %10.2f kJ',work/1000)
```

```
Work to Press:       1.23 kJ
```

The required inertia can be computed as,

$$\frac{2\left(E_2 - E_1\right)}{\left(\omega_{max}^2 - \omega_{min}^2\right)} = I$$

The change in energy is equaled to the work required per disk,

$$\frac{2(W)}{\left(\omega_{max}^2 - \omega_{min}^2\right)} = I$$

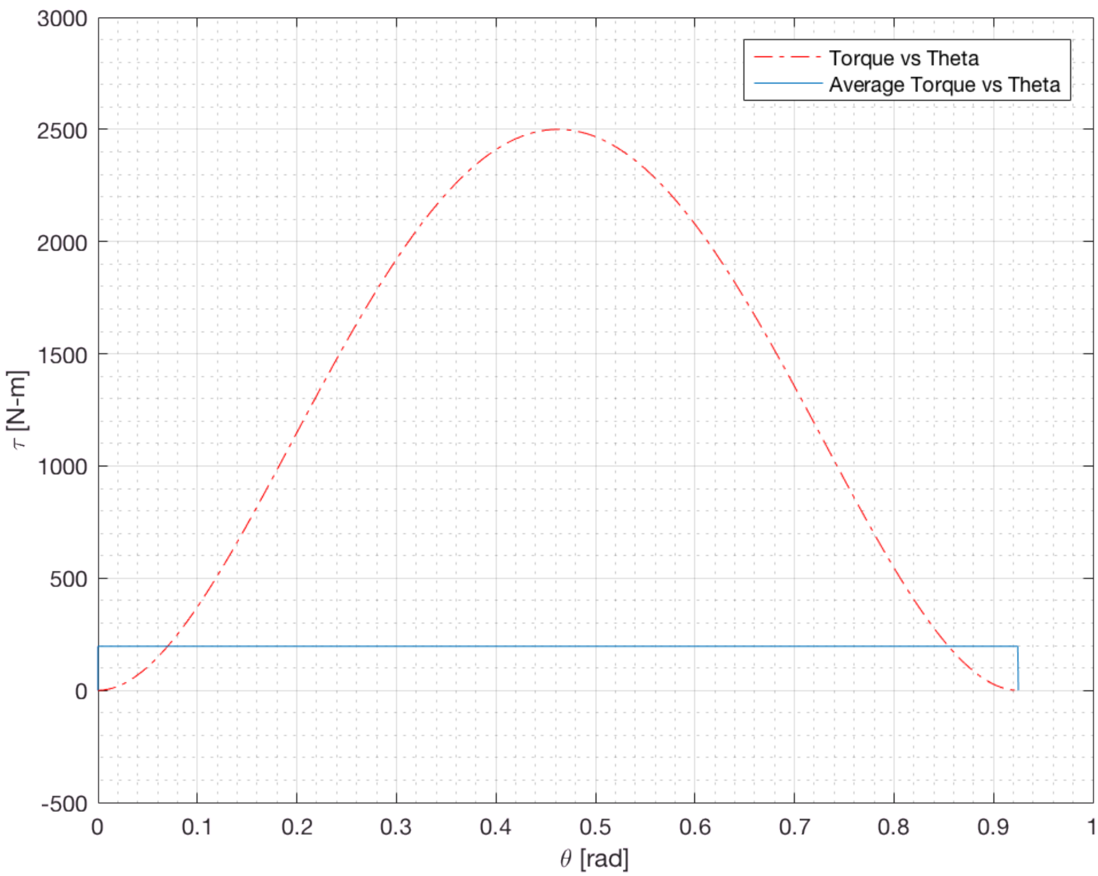
```
omega_ave = (RequiredRPMs*2*pi)/60;
omega_max = 1.15*omega_ave;
omega_min = 0.85*omega_ave;

I = (2*work)/((omega_max^2)-(omega_min^2)); % kg*m^2

fprintf('Flywheel Inertia: %10.2f kg-m^2 \n',I)
```

```
Flywheel Inertia:      43.68 kg-m^2
```

The chosen flywheel must have this inertia.

This result can be superimposed over the actual punch torque as:

```
figure('Name','Torque/AverageTorque vs theta')
plot(thetaPress,torquePress,'r-.',[0,thetaPress,0.9250245036],averageTorqueArray)
legend('Torque vs Theta','Average Torque vs Theta')
xlabel('\theta [rad]')
ylabel('\tau [N-m]')
grid on
grid minor
```



The power transmitted will be approximated using the average torque as,

$$\bar{T}\omega_{ave} = P$$

Therefor, this linear approximation results in the average power transmitted. However, peak motor power can be estimated by presuming that peak torque occurs at the average angular velocity.

$$T_{max}\omega_{ave} = P_{peak}$$

```
Power_ave = averageTorque*omega_ave; disp(['Average Power: ',num2str(Power_ave/1000),' kW']);.
    disp(['Average Power: ',num2str(1.34*Power_ave/1000),' hp']);...
Power_peak = max(torquePress)*omega_ave;...
disp(['Peak Power:    ',num2str(Power_peak/1000),'  kW']);...
disp(['Peak Power:    ',num2str(1.34*Power_peak/1000),'  hp'])
```

```
Average Power: 1.9041 kW
Average Power: 2.5515 hp
Peak Power:    24.253  kW
Peak Power:    32.4991  hp
```

As seen, the maximum power draw exceeds the 24 kW while the motor supplies 1.9 kW on average. This power is used to store energy in the Flywheel. We presume the FlyWheel is made from Cast Iron. The thickness will be assumed to be 57 mm.

$$\rho = 7800 \ \ kg/m^3$$

The process will be solved as,

$$m = \frac{\pi d^2 t \rho}{4}$$

$$I = 43.68kg - m^2 = \frac{md^2}{8}$$

$$m * d^2 = 349.44$$

$$m = \frac{349.44}{d^2}$$

$$\frac{\pi d^2 t \rho}{4} = \frac{349.44}{d^2}$$

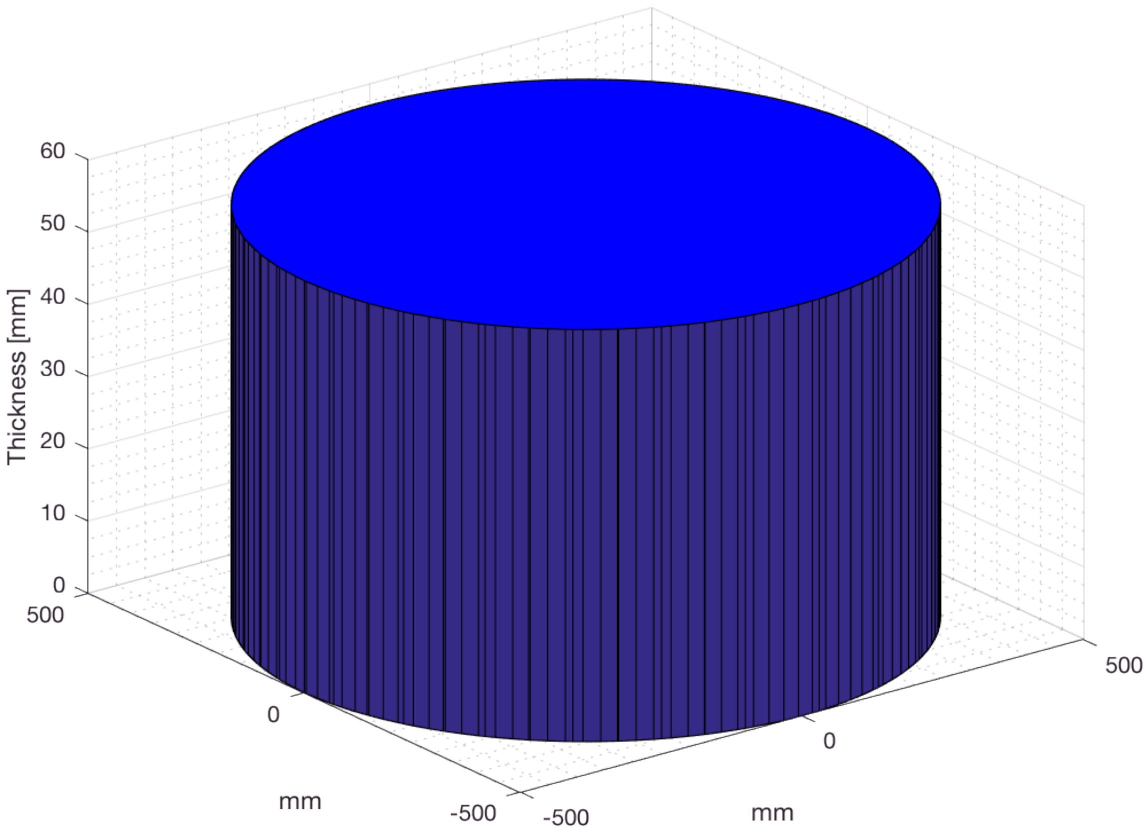
$$d = \left(\frac{1397.8}{\pi(57/1000)7800}\right)^{0.25}$$

```
th = (57/1000);
cost = 1.42; % USD/kg
rho = 7800;
FlyWheeld = (1397.8/(pi*th*rho))^0.25;
m = (pi*(FlyWheeld^2)*th*rho)/4;
disp(['d = ', num2str(1000*round(FlyWheeld,2)),' mm']);...
disp(['m = ', num2str(round(m,2)),' kg']);...
disp(['cost = $', num2str(round(m*cost,2))])
```

```
d = 1000 mm
m = 349.32 kg
cost = $496.03
```

In order to visualize the flywheel,

```
r = round(1000*FlyWheeld^2/2);
h = round(th*1000);
angle = 0:0.05:2*pi;
x = r*cos(angle);
y = r*sin(angle);
y(end) = 0;
z1 = 0;
z2 = h;
[X,Y,Z] = cylinder(1000/2,50);
Z(2,:)=57;
figure('Name','Flywheel')
surf(X,Y,Z); hold on
xlabel('mm')
ylabel('mm')
zlabel('Thickness [mm]')
grid on
grid minor
patch(x,y,z1*ones(size(x)),'b'); hold on
patch(x,y,z2*ones(size(x)),'b'); hold on
surf([x;x],[y;y],[z1*ones(size(x));z2*ones(size(x))]); hold on
```



Gear Definition

The gears employed will be Helical Gears.

The first gear pair (connected to the crankshaft) will be defined as a 17 tooth pinion driving a 34 tooth gear. Th middle pair will be designed to be identical to the first gear pair given the identical reduction. The last reduction will feature a 20 tooth tooth pinion driving a 55 tooth gear. Bringing back the previously worked equations,

$$N_2 = 2N_1$$
$$N_4 = 2N_3$$
$$N_6 = 2.75N_5$$

$$N_1 = N_3$$
$$N_3 = 1.25N_5$$

Following this notation,

$$N_1 = 17$$
$$N_2 = 34$$
$$N_3 = 17$$
$$N_4 = 34$$
$$N_5 = 20$$
$$N_6 = 55$$

The pitch diameter can be computed by setting the following values for the module,

$$m_6 = 4$$
$$m_4 = 8$$
$$m_2 = 10$$

From these values, the pitch diameter can be easily obtained as,

$$d_2 = 10 * 34 = 340 \ \ mm$$
$$d_1 = d_2/2 = 170 \ \ mm$$
$$d_4 = 34 * 8 = 272 \ \ mm$$
$$d_3 = d_4/2 = 136 \ \ mm$$
$$d_6 = 4 * 55 = 220 \ \ mm$$
$$d_5 = d_6/2.75 = 80$$

The following modules are computed from the resulting diameters,

$$m_5 = 4$$
$$m_3 = 8$$
$$m_1 = 10$$

```
N1 = 17;
N3 = N1;
N2 = FirstReduction*N1;
N4 = N2;
N5 = 20;
N6 = ThirdReduction*N5;
m6 = 4;
m4 = 8;
m2 = 10;
d2 = m2*N2;
d1 = d2/FirstReduction;
d4 = m4*N4;
d3 = d4/SecondReduction;
d6 = m6*N6;
d5 = d6/ThirdReduction;
m1 = d1/N1;
```

```
m3 = d3/N3;
m5 = d5/N5;
P1 = N1/d1;
P2 = N2/d2;
P3 = N3/d3;
P4 = N4/d4;
P5 = N5/d5;
P6 = N6/d6;
```

These quantities will be vectorized to facilitate future computing,

```
P = [P1, P2, P3, P4, P5, P6];
m = [m1, m2, m3, m4, m5, m6];
N = [N1, N2, N3, N4, N5, N6]
```

```
N = 1x6 double

    17    34    17    34    20    55
```

```
d = [d1, d2, d3, d4, d5, d6]
```

```
d = 1x6 double

   170   340   136   272    80   220
```

The addendum and dedendum can be easily computed through the following relationships for Helical Gears:

$$a = \frac{1.00}{P_n}$$

$$b = \frac{1.25}{P_n}$$

```
a = 1.00.*m; % Addendum
b = 1.25.*m; % Dedendum
p = pi./P;   % circular pitch
t = p./2;    % tooth thickness
c = b-a;     % clearance
```

The helical and pressure will be explicitly labeled to provide a general framework.

```
helicalAngle = 0;   % Deg
pressureAngle = 20; % Deg
P = P.*cosd(helicalAngle);

m = m.*cosd(helicalAngle);
disp('Pitch Diameter has been generalized to individual gears although gear pairs have the sam
disp(['Pitch Diameter: ',num2str(round(P,4),'%10.5f'),' 1/mm']);...
    disp(['Pitch Diameter: ',num2str(round(P./0.039,4),'%10.5f'),' 1/in'])
```

```
Pitch Diameter has been generalized to individual gears although gear pairs have the same value.
Pitch Diameter: 0.10000   0.10000   0.12500   0.12500   0.25000   0.25000 1/mm
Pitch Diameter: 2.56410   2.56410   3.20510   3.20510   6.41030   6.41030 1/in
```

The pitch diameter can be generalized to:

```
pitchDiameter = N./P
```

```
pitchDiameter = 1x6 double

   170   340   136   272    80   220
```

The base diameter can also be generalized to:

```
baseDiameter = d.*cosd(pressureAngle)
```

```
baseDiameter = 1x6 double

  159.7477  319.4955  127.7982  255.5964   75.1754  206.7324
```

Other relevant quantities include,

- Standard center distance

$$SCD = \frac{D+d}{2}$$

```
SCD = (d(1:2:6)+d(2:2:6))/2
```

```
SCD = 1x3 double

   255   204   150
```

- Outside Diameter

$$OD = D + 2a$$

```
OD = d+2*a
```

```
OD = 1x6 double

   190   360   152   288    88   228
```

- Root Diameter

$$RD = D - 2b$$

```
RD = d-2*b
```

```
RD = 1x6 double

   145   315   116   252    70   210
```

- Base helix angle

$$\tan^{-1}(\tan(\psi)\cos(\varphi))$$

```
BHA = atand(tand(helicalAngle).*cosd(pressureAngle))
```

```
BHA = 0
```

Gear Rating

All gear pairs will be evaluated in accordance to the roadmap for the ANSI/AGMA 2001-D04 standard as provided by [Shigley].

The first step computes the pitch diameter which has been stored in `pitchDiameter`. The tangential velocity is then computed as,

$$V = \pi d_p n_p$$

```
n = [RequiredRPMs,RequiredRPMs.*2,RequiredRPMs.*2,...
(RequiredRPMs.*2).*2,(RequiredRPMs.*2).*2,((RequiredRPMs.*2).*2).*2.75]; % RPM
V = ((pi.*pitchDiameter.*n)./1000)./60 % m/s
```

```
V = 1x6 double

    0.8246    3.2984    1.3194    5.2774    1.5522   11.7384
```

The transmitted load can then be computed through the following expression,

$$W^t[N] = \frac{Power[Watts]}{V[m/s]}$$

```
W_t = Power_ave./V % Newton
```

```
W_t = 1x6 double

   1.0e+03 *
    2.3091    0.5773    1.4432    0.3608    1.2267    0.1622
```

The **Overload Factor**, K_o, can be obtained from the following table:

```
PowerSource = {'Uniform    ';'LightShock ';'MediumShock'};
Uniform = [1.00; 1.25; 1.50];
ModerateShock = [1.25; 1.50; 1.75];
HeavyShock = [1.75; 2.00; 2.25];
KoTable = table(PowerSource,Uniform,ModerateShock,HeavyShock);
disp('                 Table of Overload Facots, Ko                  ');...
disp('-------------------------------------------------------------');...
disp('                        Driven Machine                         ');...
disp('-------------------------------------------------------------');...
disp(KoTable)
```

```
                 Table of Overload Facots, Ko
-------------------------------------------------------------
                        Driven Machine
-------------------------------------------------------------
    PowerSource      Uniform    ModerateShock    HeavyShock
    _____________    _______    _____________    __________

    'Uniform    '       1           1.25            1.75
```

```
    'LightShock '     1.25        1.5                  2
    'MediumShock'      1.5       1.75               2.25
```

```
Ko = ones(1,numel(N)).*1.25
```

```
Ko = 1x6 double

    1.2500    1.2500    1.2500    1.2500    1.2500    1.2500
```

From the problem statement and the derivation made upto this point, we can model the engine as a uniform power source to a moderate shock machine which yields a K_o **of 1.25**.

The **Dynamic Facor**, K_v, can be obtained from the following equation,

$$K_v = \left(\frac{A+\sqrt{200V}}{A}\right)^B$$

where,

$$A = 50 + 56(1 - B)$$
$$B = 0.25(12 - Q_v)^{2/3}$$

$$A = 50 + 56\left(1 - \left(0.25(12 - Q_v)^{2/3}\right)\right)$$

And Q_v is defined as the set of quality number ranging usually from 3 to 7 for commercial applications and between 8 and 12 for precision gearing.

```
Qv = 7;
B = 0.25*((12-Qv)^(2/3));
A = 50 + 56*(1-B);
% The following callback asserts the validity of the selected Qv.
assert(min(((A+(Qv-3)^2)/200) < V),'Please change Quality number as V exceeds the recommended

Kv = ((A+sqrt(200.*V))./A).^B
```

```
Kv = 1x6 double

    1.1407    1.2753    1.1769    1.3446    1.1915    1.5021
```

The **Size Factor**, K_s, can be obtained from the following equation,

$$K_s = 1.192\left(\frac{F\sqrt{Y}}{P}\right)^{0.0535}$$

```
Y = [0.303,0.371,0.303,0.371,0.322,(0.409+0.422)/2];
F = [200,200,200,200,150,150]; disp(['Face Width: ',num2str(F,'%15.2f'), ' mm']); disp(['Face
```

```
Face Width: 200.00          200.00          200.00          200.00          150.00          150.00 mm
Face Width:   7.80            7.80            7.80            7.80            5.85            5.85 in
```

```
Ks = 1.192*(((F.*sqrt(Y))./P).^0.0535)
```

```
Ks = 1x6 double
```

```
    1.7338    1.7433    1.7133    1.7226    1.6283    1.6395
```

When working in SI units, the **Load-Distribution Factor** is denoted as K_H and is determined through:

$$K_H = C_{mf} = 1 + C_{mc}(C_{pf}C_{pm} + C_{ma}C_e)$$

In this expression,

$$\frac{F}{d_p} \leq 2$$

```
assert(min(F./pitchDiameter <= 2),'Condition for this procedure not met!')
```

In order to compute the necessary procedure, several logical decisions must be made,

- Crowned or Uncrowned

```
Crowned = 1; % Mark 1 if crowned, 0 otherwise;
Cmc = zeros(1,round(numel(N)));
if Crowned == 1
    Cmc(:) = 1;
else
    Cmc(:) = 0.8;
end
```

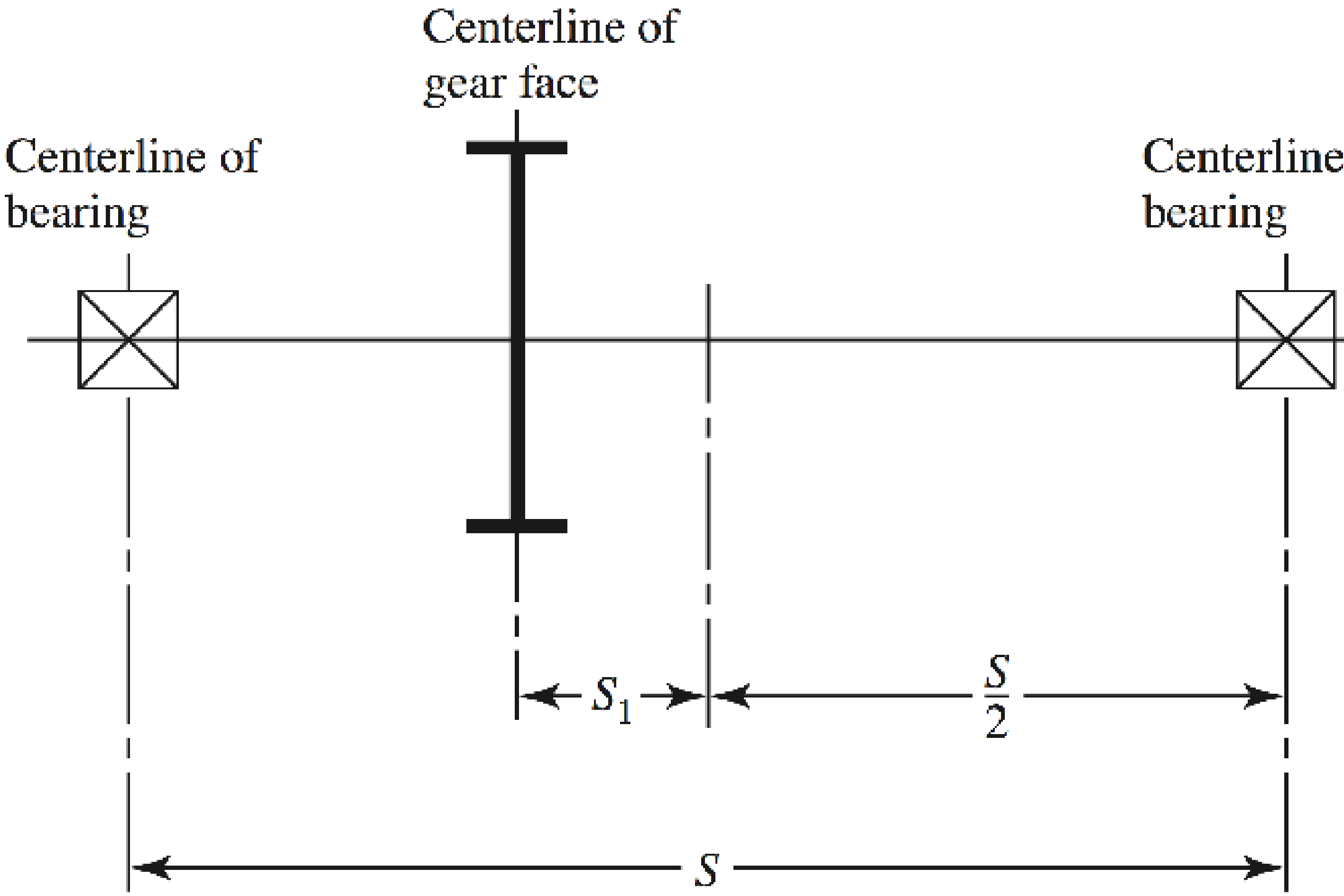
- Determine C_{pf} from the dedendum and pitch diameter.

```
Cpf = zeros(1,round(numel(N)));
bMask = b;
b10d = (bMask./(10.*pitchDiameter));

if any(b10d < 0.05)
    b10d(b10d < 0.05) = 0.05;
end

logicalPath = bMask <= 25;
if any(logicalPath)
    Cpf(logicalPath) = b10d(logicalPath) - 0.025;
end

logicalPath = b > 25 & b <= 425;
if any(logicalPath)
    Cpf(logicalPath) = b10d(logicalPath) - 0.0375+4.92*(10^-4).*bMask(logicalPath);
end

logicalPath = b > 425 & b <= 1000;
if any(logicalPath)
    Cpf(logicalPath) = b10d(logicalPath) - 0.1109 + 8.15*(10^-4).*bMask(logicalPath) - 3.53*(1
end
Cpf
```

```
Cpf = 1x6 double

    0.0250    0.0250    0.0250    0.0250    0.0250    0.0250
```

- For immediatly adjacent bearings, $C_{pm} = 1$. Otherwise, $C_{pm} = 1.1$.

Adjacency will be determined by $\frac{S_1}{S}$,




```
S1_S_factor = 0; % S1/S; 0.25 means the gear is placed at .5 S/2 or 1/4 the full length of the
if S1_S_factor < 0.175
    AdjacentBearing = 1;
elseif S1_S_factor >= 0.175
    AdjacentBearing = 0;
end
Cpm = zeros(1,round(numel(N)));
if AdjacentBearing == 1
    Cpm(:) = 1;
else
    Cpm(:) = 1.1;
end
Cpm
```

```
Cpm = 1x6 double

     1     1     1     1     1     1
```

- The mesh alignment factor, C_{ma}

$$C_{ma} = A + BF + CF^2$$

The conditions must be selected according to the following numeric IDs:

1. Open Gearing
2. Commercial, enclosed units
3. Precision, enclosed units
4. Extraprecision, enclosed gear units

```
Cma = zeros(1,round(numel(N)));
CmaConditions = 2; % Match CmaConditions with the numeric IDs

CmaFact = [0.247, 0.0167, -0.765*(10^-4); 0.127, 0.0158, -0.930*(10^-4);...
    0.0675, 0.0128, -0.926*(10^-4); 0.00360, 0.0102, -0.822*(10^-4)];
Cma(:) = CmaFact(CmaConditions,1) + CmaFact(CmaConditions,2).*F + CmaFact(CmaConditions,3).*(F
```

```
Cma = 1x6 double

   -0.4330   -0.4330   -0.4330   -0.4330    0.4045    0.4045
```

- The mesh alignment correction factor, C_e

1. For gearing adjusted at assembly, or compatibility is improved by lapping, or both: $C_e = 0.8$
2. For all other conditions: $C_e = 1.0$

```
CeConditions = 2;
Ce = zeros(1,round(numel(N)));
if CeConditions == 2
    Ce(:) = 1.0;
else
    Ce(:) = 0.8;
end
Ce
```

```
Ce = 1x6 double

     1     1     1     1     1     1
```

We can now compute K_H,

$$K_H = C_{mf} = 1 + C_{mc}(C_{pf}C_{pm} + C_{ma}C_e)$$

```
Kh = 1 + Cmc.*((Cpf.*Cpm)+(Cma.*Ce))
```

```
Kh = 1x6 double

    0.5920    0.5920    0.5920    0.5920    1.4295    1.4295
```

We must now compute the **Stress-Cycle Factors**, Y_N & Z_N,

$$m_G = \frac{N_G}{N_P}$$

$$(Y_N)_P = \frac{1.3558N^{-0.0178} + 1.6831N^{-0.0323}}{2}$$

$$(Z_N)_P = \frac{1.4488N^{-0.0230} + 2.4660N^{-0.0560}}{2}$$

$$(Y_N)_G = \frac{1.3558\left(\frac{N}{m_G}\right)^{-0.0178} + 1.6831\left(\frac{N}{m_G}\right)^{-0.0323}}{2}$$

$$(Z_N)_G = \frac{1.4488\left(\frac{N}{m_G}\right)^{-0.0230} + 2.4660\left(\frac{N}{m_G}\right)^{-0.0560}}{2}$$

```
mG = N(2:2:6)./...
     N(1:2:6);
YNP = zeros(1,numel(N)/2);
YNG = zeros(1,numel(N)/2);
ZNP = zeros(1,numel(N)/2);
ZNG = zeros(1,numel(N)/2);
YN = zeros(1,numel(N));
ZN = zeros(1,numel(N));

life = 10^9;
lifemg = life ./ mG;

YNP(:) = ((1.3558.*life.^(-0.0178))+(1.4488.*life.^(-0.0323)))/2;
ZNP(:) = ((1.4488.*life.^(-0.0230))+(2.4660.*life.^(-0.0560)))/2;

YNG(1) = ((1.3558.*lifemg(1).^(-0.0178))+(1.4488.*lifemg(1).^(-0.0323)))/2;
ZNG(1) = ((1.4488.*lifemg(1).^(-0.0230))+(2.4660.*lifemg(1).^(-0.0560)))/2;
YNG(2) = ((1.3558.*lifemg(2).^(-0.0178))+(1.4488.*lifemg(2).^(-0.0323)))/2;
ZNG(2) = ((1.4488.*lifemg(2).^(-0.0230))+(2.4660.*lifemg(2).^(-0.0560)))/2;
YNG(3) = ((1.3558.*lifemg(3).^(-0.0178))+(1.4488.*lifemg(3).^(-0.0323)))/2;
ZNG(3) = ((1.4488.*lifemg(3).^(-0.0230))+(2.4660.*lifemg(3).^(-0.0560)))/2;

ZN(2:2:6) = ZNG;
ZN(1:2:6) = ZNP;
YN(2:2:6) = YNG;
YN(1:2:6) = YNP;

disp(table(YNP',YNG',ZNP',ZNG','VariableNames',{'YN_P','YN_G','ZN_P','ZN_G'}));...
    disp(table(YN',ZN','VariableNames',{'YN','ZN'}))
```

YN_P	YN_G	ZN_P	ZN_G
0.83969	0.85391	0.83609	0.85861
0.83969	0.85391	0.83609	0.85861
0.83969	0.86053	0.83609	0.8692

YN	ZN
0.83969	0.83609

```
    0.85391    0.85861
    0.83969    0.83609
    0.85391    0.85861
    0.83969    0.83609
    0.86053     0.8692
```

Now, let's compute the **Reliability Factor** $K_R\ (Y_Z)$,

$$K_R = 0.50 - 0.109\ln(1 - R) \qquad 0.99 \le R \le 0.9999$$

```
R = [0.99,0.99,0.99,0.99,0.99,0.99];
KR = zeros(1,numel(N));

KR(R>=0.99) = 0.50 - 0.109.*log(1-R(R>=0.99));
KR(R<0.99) = 0.658 - 0.0759.*log(1-R(R<0.99));
KR
```

```
KR = 1x6 double

    1.0020    1.0020    1.0020    1.0020    1.0020    1.0020
```

The **Temperature Factor** K_T can be equaled to 1 for this application.

```
KT = ones(1,numel(N))
```

```
KT = 1x6 double

     1     1     1     1     1     1
```

A factor denoted as **Rim-Thickness Factor** K_B can also be setted to 1 assuming constant thickness gears. However, the correct value will be estimated using the following procedure,

$$m_B = \frac{t_R}{h_t}$$

$$\begin{aligned} &if\ \ m_B < 1.2 \\ &\qquad K_B = 1.6\ln\left(\frac{2.242}{m_B}\right) \\ &else\ \ m_B \le 1.2 \\ &\qquad K_B = 1 \end{aligned}$$

```
KB = ones(1,numel(N));
tol= 15; % mm
mB = (RD./2)-tol;
disp('Estimated mB');...
disp(mB')
```

```
Estimated mB
   57.5000
  142.5000
   43.0000
  111.0000
   20.0000
```

```
    90.0000
```

```
KB(mB<1.2) = 1.6.*log(2.242./mB(mB<1.2))
```

```
KB = 1x6 double

     1     1     1     1     1     1
```

The pinion and gear **Bending-Strength Geometry Factor** J is estimated from *Figure 14-6* and must be updated if the number of tooths is changed.

```
J1 = 0.295;
J2 = 0.370;
J3 = 0.295;
J4 = 0.370;
J5 = 0.330;
J6 = 0.405;
%---------------------------%
J = [J1, J2, J3, J4, J5, J6];
```

The **Surface-Strength Geometry Factor** I (also called the *pitting-resistance geometry factor*) can be easily computed by setting m_n to 1 and evaluating the following expression:

$$I = \left(\frac{\cos(\varphi)\sin(\varphi)}{2m_n}\right)\left(\frac{m_G}{m_G+1}\right)$$

```
mn = 1; % which is valid for spur gears
I_3 = ((cosd(pressureAngle)*sind(pressureAngle))/2*mn).*(mG./(mG+1));
I = zeros(1,numel(N));
I(1:2) = I_3(1);
I(3:4) = I_3(2);
I(5:6) = I_3(3);
clearvars I_3
I
```

```
I = 1x6 double

    0.1071    0.1071    0.1071    0.1071    0.1178    0.1178
```

The **Elastic Coefficient** C_p [$\sqrt{MPa}$] can be obtained from Table 14-8 [Shigley's] assuming the Pinion and Gear to be Grade 2 Steel with hardness of 300 and 240 BHN respectively.

```
Cp = zeros(1,numel(N));
HBP = zeros(1,numel(N)/2);
HBP(:) = 350;
HBG = zeros(1,numel(N)/2);
HBG(:) = 290;
HB = zeros(1,numel(N));
HB(2:2:6) = HBG;
HB(1:2:6) = HBP;
Cp(:) = 191
```

```
Cp = 1x6 double

   191   191   191   191   191   191
```

```
HB
```

```
HB = 1x6 double

   350   290   350   290   350   290
```

The **allowable bending stress number** can be determined through the following expression:

$$S_t = 0.703 \odot H_B + 113 \;\; MPa$$

```
St = 0.703.*HB+113
```

```
St = 1x6 double

  359.0500  316.8700  359.0500  316.8700  359.0500  316.8700
```

Similarly, **contact-fatigue strength** can be estimated through the following expression:

$$S_c = 2.41 \odot H_B + 237 \;\; MPa$$

```
Sc = 2.41.*HB+237
```

```
Sc = 1x6 double

   1.0e+03 *
    1.0805    0.9359    1.0805    0.9359    1.0805    0.9359
```

The **hardness ratio** is computed per gear pair as,

$$\frac{H_{BP}}{H_{BG}}$$

```
Hratio = HBP./HBG
```

```
Hratio = 1x3 double

    1.2069    1.2069    1.2069
```

The **Hardness-Ratio Factor**, C_H, can be determined as,

$$C_{H_{Pinion}} = 1$$

$$C_{H_{Gear}} = 1.0 + A'\,(m_G - 1.0)$$

$$where,\;\; A' = 8.98\left(10^{-3}\right)\left(\frac{H_{BP}}{H_{BG}}\right) - 8.29\left(10^{-3}\right) \qquad 1.2 \le \frac{H_{BP}}{H_{BG}} \le 1.7$$

```
CH = ones(1,numel(N));
Aprime = (8.98.*(10^-3).*(Hratio))-(8.29*(10^(-3)));
CH(2:2:6) = 1+Aprime.*(8.98.*mG-1.0)
```

```
CH = 1x6 double
```

```
1.0000    1.0432    1.0000    1.0432    1.0000    1.0604
```

Stress, Bending and Wear

$$\sigma = W^t K_o K_v K_s \frac{1}{bm_t} \frac{K_H K_B}{Y_J}$$

```
sigma = W_t.*Ko.*Kv.*Ks.*(1./(F.*m)).*((Kh.*KB)./J);
disp('Sigma (MPa)');...
fprintf('%6.2f MPa\n',sigma)
```

```
Sigma (MPa)
  5.73 MPa
  1.28 MPa
  4.56 MPa
  1.04 MPa
 21.48 MPa
  2.94 MPa
```

The safety factor can be computed as,

$$S_F = \frac{\left(\frac{S_t Y_N}{(K_T K_R)}\right)}{\sigma}$$

```
SafetyF = (((St.*YN)./(KT.*KR))./sigma)
```

```
SafetyF = 1x6 double

   52.5294  210.4101   65.9519  258.5284   14.0097   92.6460
```

The contact stress can now be estimated using the following equation,

$$\sigma_c = C_p \sqrt{W^t K_o K_v K_s \frac{K_H}{d\ F} \frac{1}{I}}$$

```
sigmaC = Cp.*sqrt(W_t.*Ko.*Kv.*Ks.*(Kh./(d.*F)).*(1./I));
disp('Sigma Contact (MPa)');...
fprintf('%6.2f MPa\n',sigmaC)
```

```
Sigma Contact (MPa)
183.98 MPa
 68.96 MPa
164.20 MPa
 62.22 MPa
331.22 MPa
 81.83 MPa
```

The conctact stress' safety factor can be determined through the following expression,

$$S_H = \frac{\left(\frac{S_c Z_N C_H}{(K_T K_R)}\right)}{\sigma_c}$$

```
SafetyH = (((Sc.*ZN.*CH)./(KT.*KR))./sigmaC)
```

```
SafetyH = 1x6 double

    4.9007   12.1317    5.4912   13.4475    2.7221   10.5207
```

```
SafetyHsq = SafetyH.^2
```

```
SafetyHsq = 1x6 double

   24.0164  147.1783   30.1532  180.8362    7.4099  110.6850
```

We can now proceed to compute the gear bending endurance strength and the gear contact endurance strength,

$$\sigma_{all} = \frac{S_t}{S_F}\frac{Y_N}{K_T K_R}$$

$$\sigma_{c,all} = \frac{S_c Z_N C_H}{S_H K_T K_R}$$

```
sigmaAll = (St.*YN)./(SafetyF.*KT.*KR); disp('Sigma All'); fprintf('%5.2f MPa\n', sigmaAll)
```

```
Sigma All
 5.73 MPa
 1.28 MPa
 4.56 MPa
 1.04 MPa
21.48 MPa
 2.94 MPa
```

```
sigmaCAll = (Sc.*ZN.*CH)./(SafetyH.*KT.*KR); disp('SigmaC All'); fprintf('%5.2f MPa\n', sigmaC
```

```
SigmaC All
183.98 MPa
68.96 MPa
164.20 MPa
62.22 MPa
331.22 MPa
81.83 MPa
```

In order to determine the dominant failure mode per gear pair, we can either use the S_H or $\left(S_H^2\right)$

```
if SafetyF(1) < SafetyH(1)
    fail1 = 'Gear Pair N1-N2, attached to the last section prior to the flywheel, dominant''s
else
    fail1 = 'Gear Pair N1-N2, attached to the last section prior to the flywheel, dominant''s
end
if SafetyF(3) < SafetyH(3)
    fail2 = 'Gear Pair N3-N4, the middle gear pair, dominant''s failure mode is Bending.';
else
```

```
    fail2 = 'Gear Pair N3-N4, the middle gear pair, dominant''s failure mode is Pitting.';
end
if SafetyF(5) < SafetyH(5)
    fail3 = 'Gear Pair N5-N6, attached to the first section prior to the engine, dominant''s f
else
    fail3 = 'Gear Pair N5-N6, attached to the first section prior to the engine, dominant''s f
end
if SafetyF(1) < SafetyHsq(1)
    fail1s = 'Gear Pair N1-N2, attached to the last section prior to the flywheel, dominant''s
else
    fail1s = 'Gear Pair N1-N2, attached to the last section prior to the flywheel, dominant''s
end
if SafetyF(3) < SafetyHsq(3)
    fail2s = 'Gear Pair N3-N4, the middle gear pair, dominant''s failure mode is Bending.';
else
    fail2s = 'Gear Pair N3-N4, the middle gear pair, dominant''s failure mode is Pitting.';
end
if SafetyF(5) < SafetyHsq(5)
    fail3s = 'Gear Pair N5-N6, attached to the first section prior to the engine, dominant''s
else
    fail3s = 'Gear Pair N5-N6, attached to the first section prior to the engine, dominant''s
end
disp('Presuming S_H,');disp(fail1); disp(fail2); disp(fail3)
```

```
Presuming S_H,
Gear Pair N1-N2, attached to the last section prior to the flywheel, dominant's failure mode is Pitting.
Gear Pair N3-N4, the middle gear pair, dominant's failure mode is Pitting.
Gear Pair N5-N6, attached to the first section prior to the engine, dominant's failure mode is Pitting.
```

```
disp('Presuming S_H^2,');disp(fail1s); disp(fail2s); disp(fail3s)
```

```
Presuming S_H^2,
Gear Pair N1-N2, attached to the last section prior to the flywheel, dominant's failure mode is Pitting.
Gear Pair N3-N4, the middle gear pair, dominant's failure mode is Pitting.
Gear Pair N5-N6, attached to the first section prior to the engine, dominant's failure mode is Pitting.
```

From the results at this moment, we can conclude that the dominant failure mode is pitting.

Crankshaft Design

As last step, we will design the crankshaft presuming the predominant load to be torsion. The crankshaft will be solid with,

$$J = \frac{\pi d^4}{32}$$

The stress in the shaft can be computed as,

$$\tau_{max} = \frac{Tr}{J} = \frac{Td}{2\left(\frac{\pi d^4}{32}\right)} = \frac{T}{\left(\frac{\pi d^3}{16}\right)} = \frac{16T}{(\pi d^3)}$$

```
d = 55/1000;
tauMaxShaft = (16*max(torquePress))/(pi*d^3);
disp(['Max Torsional Stress: ', num2str(tauMaxShaft/1000000),' MPa'])
```

```
Max Torsional Stress: 76.5291 MPa
```

Summary

The Nominal Motor Speed:

```
disp(['Nominal Motor Speed: ',num2str(MotorRPMs), ' RPM'])
```

```
Nominal Motor Speed: 1020 RPM
```

Work to press each washer:

```
disp(['Work: ', num2str(work), ' J'])
```

```
Work: 1233.2307 J
```

Average torque during punching:

```
disp(['Average Torque: ',num2str(averageTorque),' N-m'])
```

```
Average Torque: 196.2748 N-m
```

Minimum motor power to provide peak crankshaft torque:

```
disp(['Minimum motor power: ', num2str(Power_ave/1000),' kW'])
```

```
Minimum motor power: 1.9041 kW
```

The maximum torsional stress in crankshaft:

```
disp(['Max Torsional Stress (Crankshaft): ', num2str(tauMaxShaft),' Pa']);...
    disp(['Max Torsional Stress (Crankshaft): ', num2str(tauMaxShaft/1000000),' MPa'])
```

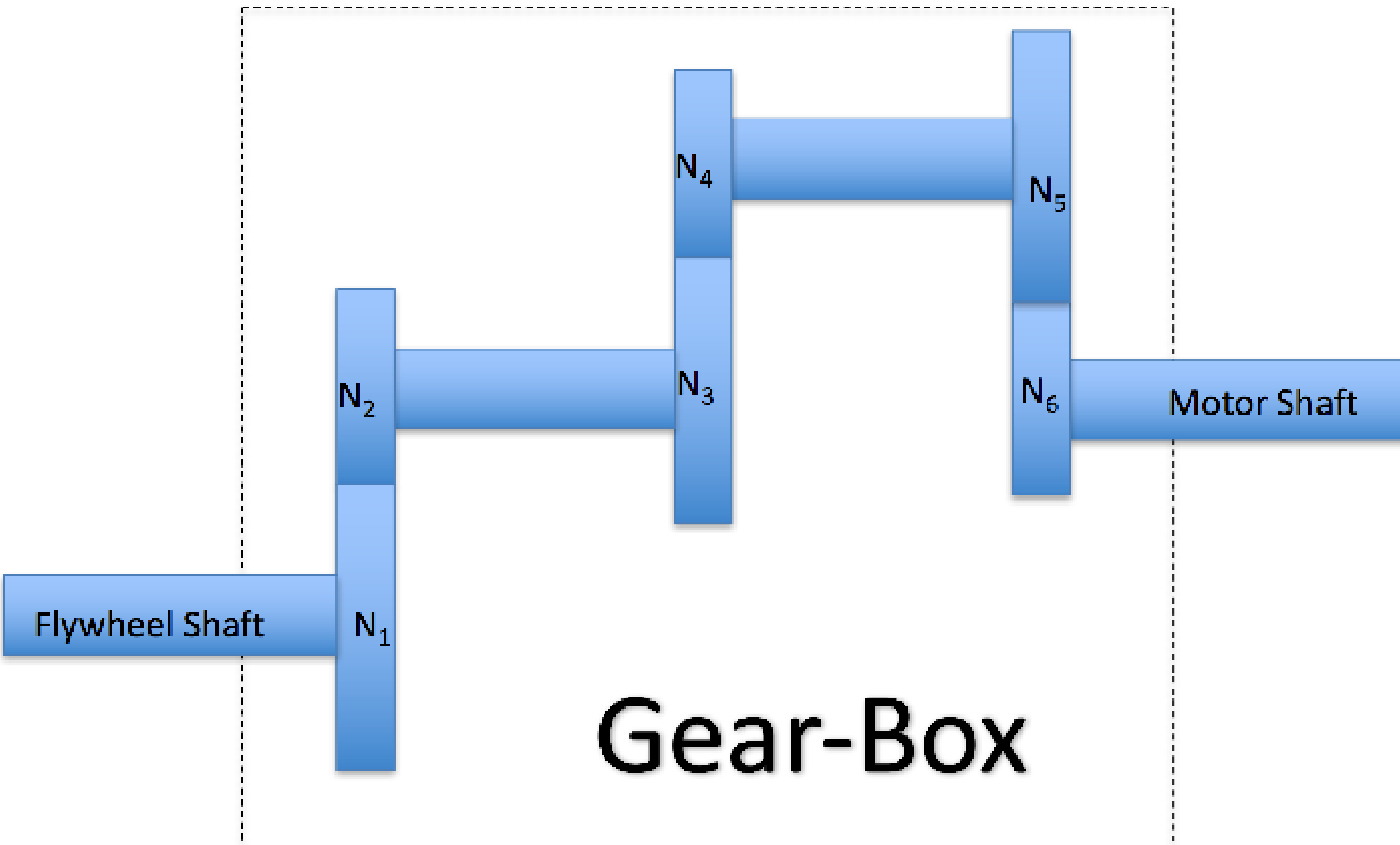
```
Max Torsional Stress (Crankshaft): 76529057.0627 Pa
Max Torsional Stress (Crankshaft): 76.5291 MPa
```

The FlyWheel's diameter and thickness:

```
disp(['Flywheel''s Diameter:  ', num2str(1000*round(FlyWheeld,2)),' mm']);...
disp(['Flywheel''s Thickness:   ', num2str(1000*th),' mm'])
```

```
Flywheel's Diameter:  1000 mm
Flywheel's Thickness:   57 mm
```

Gearbox Diagram



N$_4$
N$_5$
N$_2$
N$_3$
N$_6$
Motor Shaft
Flywheel Shaft
N$_1$
Gear-Box