Table of Contents


```
clear
clc
```

Inputs

```
%All in standard SI units unless noted
OperatingVoltage = 40                                                    %V
TorqueConstant = 8.474E-3
  %N.m/A
VoltageConstant = 1125
  %RPM/V
ArmatureResistance = 0.072
  %Ohms
reqOutputSpeed = 575
  %RPM
reqOutputTorque = 55
  %N.m
Efficiency = 90/100
Duty = 2000
  %Hours

%Gears
PressureAngle = 20*pi/180
  %Rad
FaceWidth = [1.25, 1, 0.75, 0.5, 0.375, 0.25, 0.188, 0.125]
  %inch
DiametralPitch = [8, 10, 12, 16, 20, 24, 32, 48]
  %teeth/inch
k = 1
  %Teeth Depth (1 = full)
m1 = 5
  %Gear Ratio Stage 1
m2 = 4
  %Gear Ratio Stage 2
```

```
OperatingVoltage =

    40
```

```
TorqueConstant =

    0.0085

VoltageConstant =

        1125

ArmatureResistance =

    0.0720

reqOutputSpeed =

   575

reqOutputTorque =

    55

Efficiency =

    0.9000

Duty =

        2000

PressureAngle =

    0.3491

FaceWidth =

  Columns 1 through 7

    1.2500    1.0000    0.7500    0.5000    0.3750    0.2500    0.1880

  Column 8

    0.1250

DiametralPitch =
```

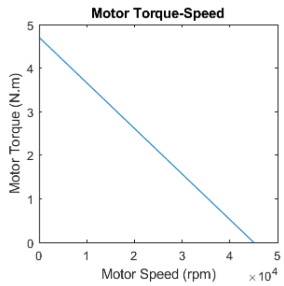
```
     8     10     12     16     20     24     32     48


k =

     1


m1 =

     5


m2 =

     4
```

Motor Torque-Speed

```
%point 1:
NoLoadSpeed = VoltageConstant*OperatingVoltage;
NoLoadTorque = 0;
%point 2:
StallTorque = TorqueConstant*OperatingVoltage/ArmatureResistance;
StallSpeed = 0;

%determine Torque-speed graph
points = [StallSpeed, StallTorque;
          NoLoadSpeed, NoLoadTorque];
polynomialDegree = length(points)-1;
MotorLine = polyfit(points(:,1),points(:,2),polynomialDegree);
MotorLineE = Efficiency.*MotorLine;

f1 = figure('Renderer', 'painters', 'Position', [10 10 1200 300])
subplot(1,3,1)
plot([StallSpeed:NoLoadSpeed], polyval(MotorLine,
[StallSpeed:NoLoadSpeed]))
hold on
title("Motor Torque-Speed")
xlabel("Motor Speed (rpm)")
ylabel("Motor Torque (N.m)")
hold off
```

```
f1 =

  Figure (6) with properties:

      Number: 6
        Name: ''
       Color: [0.9400 0.9400 0.9400]
```

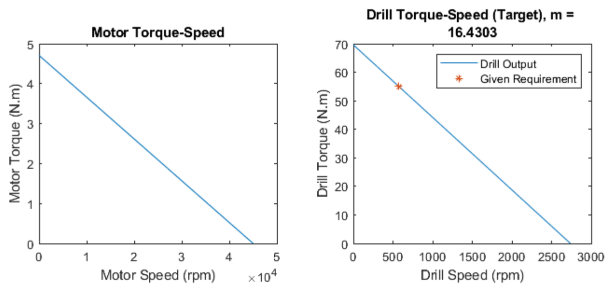
```
    Position: [10 10 1200 300]
       Units: 'pixels'

  Use GET to show all properties
```



Gear Ratio Target (minimization)

```
syms r
eqn = MotorLineE(1)*reqOutputSpeed*r^2 + MotorLineE(2)*r ==
 reqOutputTorque;
soln = double(solve(eqn,r))

ratioTarget = soln(1);

TargetLine = [MotorLineE(1)*ratioTarget^2, MotorLineE(2)*ratioTarget];

subplot(1,3,2)
plot([StallSpeed/ratioTarget: NoLoadSpeed/ratioTarget],
 polyval(TargetLine,[StallSpeed/ratioTarget: NoLoadSpeed/
ratioTarget]))
hold on
plot(reqOutputSpeed, reqOutputTorque,'*')
title(["Drill Torque-Speed (Target), m = ", ratioTarget])
xlabel("Drill Speed (rpm)")
ylabel("Drill Torque (N.m)")
legend("Drill Output", "Given Requirement")
hold off
```

```
soln =

   16.4303
   61.8306
```



Gears

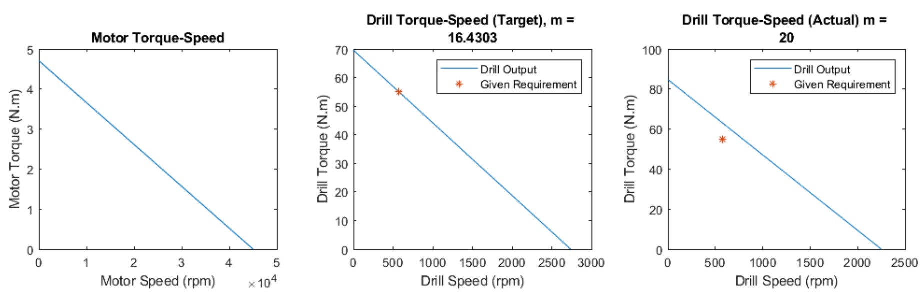
```
%pinions
P1 = ceil((2*k*(m1+(m1^2+(1-2*m1)*(sin(PressureAngle))^2)^0.5))/
((1+2*m1)*(sin(PressureAngle))^2));
P2 = ceil((2*k*(m2+(m2^2+(1-2*m2)*(sin(PressureAngle))^2)^0.5))/
((1+2*m2)*(sin(PressureAngle))^2));

%gear Limits
N1l = floor(((P1^2)*(sin(PressureAngle)^2)-4*k^2)/
(4*k-2*P1*sin(PressureAngle)^2));
N2l = floor(((P1^2)*(sin(PressureAngle)^2)-4*k^2)/
(4*k-2*P2*sin(PressureAngle)^2));

if(m1*P1 > N1l)
    N1 = N1l;   %Gear Ratio m is too large
    disp("Ratio m1 Too Large, increasing P1")
    P1 = P1+1;
    N1 = floor(m1*P1);
else
    N1 = floor(m1*P1);
end

if(m2*P2 > N2l);
    N2 = N2l;    %Gear Ratio m is too large
    disp("Ratio m2 Too Large, increasing P2")
    P2 = P2+1;
    N2 = floor(m2*P2);
else
    N2 = floor(m2*P2);
end

m1actual = N1/P1;
m2actual = N2/P2;
ratio = m1actual*m2actual;

RatioTable = table(m1, m1actual, P1, N1, m2, m2actual, P2, N2, ratio,
 ratioTarget)
```

```
RatioTable = 

  1×10 table
```

m1	m1actual	P1	N1	m2	m2actual	P2	N2	ratio	ratioTarget
5	5	16	80	4	4	16	64	20	16.43

Gear Ratio (Overall)

```
outputTorque = [0];
outputSpeed = [0];

RealLine = [MotorLineE(1)*ratio^2, MotorLineE(2)*ratio];

for x = [StallSpeed:NoLoadSpeed]
   outputSpeed = [outputSpeed, x/ratio];
   outputTorque = [outputTorque, polyval(RealLine,x/ratio)];
end
outputSpeed = outputSpeed(2:end);
outputTorque = outputTorque(2:end);
subplot(1,3,3)
plot(outputSpeed, outputTorque)
hold on
plot(reqOutputSpeed,reqOutputTorque,'*')
title(["Drill Torque-Speed (Actual) m = ", ratio])
xlabel("Drill Speed (rpm)")
ylabel("Drill Torque (N.m)")
legend("Drill Output", "Given Requirement")
hold off
```



Shaft Torque and Speed Estimation

```
%{
Motor - P1  : Shaft A
N1 - P2     : Shaft B
N2 - Chuck  : Shaft C
%}

outputPower = reqOutputSpeed*reqOutputTorque*pi/30 ;
```

```
intermediatePower = outputPower/Efficiency^0.5;
inputPower = intermediatePower/Efficiency^0.5;
efficiency = outputPower/inputPower;

%Shaft C
SpeedC = reqOutputSpeed;
TorqueC = reqOutputTorque;
%Shaft B
SpeedB = SpeedC*m2actual;
TorqueB = intermediatePower/(SpeedB*pi/30);
%Shaft A
SpeedA = SpeedB*m1actual;
TorqueA = inputPower/(SpeedA*pi/30);

TorqueSpeedEstimation = table(TorqueA, SpeedA, TorqueB, SpeedB,
 TorqueC, SpeedC)
PowerTable = table(inputPower, intermediatePower, outputPower,
 efficiency)
```

TorqueSpeedEstimation =

1×6 table

TorqueA	SpeedA	TorqueB	SpeedB	TorqueC	SpeedC
3.0556	11500	14.494	2300	55	575

PowerTable =

1×4 table

inputPower	intermediatePower	outputPower	efficiency
3679.7	3490.9	3311.8	0.9

Operating Gear Forces, Torques and Diameters

```
Force1t = [0];
Force1r = [0];
Force2t = [0];
Force2r = [0];

P1diameter = [0];
N1diameter = [0];
P2diameter = [0];
N2diameter = [0];

for x = DiametralPitch
```

```
    x = x/0.0254;  %Teeth/inch -> teeth/meter
    P1dia = P1/x; %Teeth / (Teeth/meter) = meter
    N1dia = N1/x;
    P2dia = P2/x;
    N2dia = N2/x;

    P1diameter = [P1diameter, P1dia];
    N1diameter = [N1diameter, N1dia];
    P2diameter = [P2diameter, P2dia];
    N2diameter = [N2diameter, N2dia];

    %Transmitted Load: Pinion 1 -> Gear 1
    Wt1 = (60000*inputPower*10^-3)/(pi*P1dia*SpeedA);
    %Radial Load: Pinion 1 -> Gear 1
    Wr1 = tan(PressureAngle)*Wt1;

    %Transmitted Load: Pinion 2 -> Gear 2
    Wt2 = (60000*outputPower*10^-3)/(pi*N2dia*SpeedC);
    %Radial Load: Pinion 2 -> Gear 2
    Wr2 = tan(PressureAngle)*Wt2;

    Force1t = [Force1t, Wt1];
    Force1r = [Force1r, Wr1];
    Force2t = [Force2t, Wt2];
    Force2r = [Force2r, Wr2];
end

Force1t = transpose(Force1t(2:end));
Force2t = transpose(Force2t(2:end));
Force1r = transpose(Force1r(2:end));
Force2r = transpose(Force2r(2:end));

P1diameter = transpose(P1diameter(2:end)).*100; %meters ->
 centimeters;
N1diameter = transpose(N1diameter(2:end)).*100; %meters ->
 centimeters;
P2diameter = transpose(P2diameter(2:end)).*100; %meters ->
 centimeters;
N2diameter = transpose(N2diameter(2:end)).*100; %meters ->
 centimeters;

ForceDiameterTable = table(transpose(DiametralPitch), Force1t,
 Force1r, P1diameter, N1diameter, Force2t, Force2r, P2diameter,
 N2diameter)

%{

Not confident that this is correct

%Wt, Tangential Forces
Force1t = [0];
Force2t = [0];
%Wr, Radial Forces
Force1r = [0];
```

```
Force2r = [0];
%Gear Diameters
P1diameter = [0];
N1diameter = [0];
P2diameter = [0];
N2diameter = [0];
%Shaft Torques
AT = [0];
BT = [0];
CT = [0];

for x = DiametralPitch
DiametralPitch1 = x/0.0254;
DiametralPitch2 = x/0.0254;  %Teeth/inch -> teeth/meter

P1dia = P1/DiametralPitch1; %Teeth / (Teeth/meter) = meter
N1dia= N1/DiametralPitch1;
P2dia= P2/DiametralPitch2;
N2dia= N2/DiametralPitch2;

%Logging diameters for the table below
P1diameter = [P1diameter, P1dia];
N1diameter = [N1diameter, N1dia];
P2diameter = [P2diameter, P2dia];
N2diameter = [N2diameter, N2dia];

%Tangential Forces
Ctorque = CopTorque;
Wt2 = Ctorque/(cos(PressureAngle2)*N2dia/2);
Btorque = (Wt2*P2dia/2)/cos(PressureAngle2);
Wt1 = Btorque/(cos(PressureAngle1)*N1dia/2);
Atorque = (Wt1*P1dia/2)/cos(PressureAngle1);
Force2t = [Force2t, Wt2];
Force1t = [Force1t, Wt1];
%Radial Forces
Force1r = [Force1r, tan(PressureAngle1)*Wt1];
Force2r = [Force2r, tan(PressureAngle2)*Wt2];
%Shaft Torques
AT = [AT, Atorque];
BT = [BT, Btorque];
CT = [CT, Ctorque];

end

Force1t = Force1t(2:end);
Force2t = Force2t(2:end);
Force1r = Force1r(2:end);
Force2r = Force2r(2:end);

P1diameter = P1diameter(2:end);
N1diameter = N1diameter(2:end);
P2diameter = P2diameter(2:end);
N2diameter = N2diameter(2:end);
```

```
AT = AT(2:end);
BT = BT(2:end);
CT = CT(2:end);

DiametralPitch = transpose(DiametralPitch);
Force1t = transpose(Force1t);
Force2t = transpose(Force2t);
Force1r = transpose(Force1r);
Force2r = transpose(Force2r)
P1diameter = transpose(P1diameter).*100; %meters -> centimeters
N1diameter = transpose(N1diameter).*100;
P2diameter = transpose(P2diameter).*100;
N2diameter = transpose(N2diameter).*100;
AT = transpose(AT);
BT = transpose(BT);
CT = transpose(CT);

ForceTorqueTable = table(DiametralPitch, AT, Force1t, Force1r, BT,
 Force2t, Force2r, CT)

DiameterTable = table(DiametralPitch, P1diameter, N1diameter,
 P2diameter, N2diameter)

%}
```

```
ForceDiameterTable =

  8×9 table

    Var1    Force1t    Force1r    P1diameter    N1diameter    Force2t
   Force2r    P2diameter    N2diameter
    ____    _______    _______    __________    __________    _______
   _______    __________    __________

      8      120.3     43.785          5.08          25.4      541.34
   197.03         5.08         20.32
     10     150.37     54.731         4.064         20.32      676.67
   246.29        4.064        16.256
     12     180.45     65.677        3.3867        16.933      812.01
   295.55       3.3867        13.547
     16     240.59     87.569          2.54          12.7      1082.7
   394.06         2.54         10.16
     20     300.74     109.46         2.032         10.16      1353.3
   492.58        2.032         8.128
     24     360.89     131.35        1.6933        8.4667        1624
   591.09       1.6933        6.7733
     32     481.19     175.14          1.27          6.35      2165.4
   788.12         1.27          5.08
     48     721.78     262.71       0.84667        4.2333        3248
   1182.2      0.84667        3.3867
```

Stresses

(Calculated in USCS then converted to metric)

```
ContactPinion1 = [0];
BendingPinion1 = [0];
ContactGear1 = [0];
BendingGear1 = [0];
ContactPinion2 = [0];
BendingPinion2 = [0];
ContactGear2 = [0];
BendingGear2 = [0];



for x = [1:length(DiametralPitch)]
    %Pinion 1
    [CP1, BP1] = stresses(DiametralPitch(x), P1, SpeedA, FaceWidth(x),
 Force1t(x), m1actual, PressureAngle);
    %Gear 1
    [CN1, BN1] = stresses(DiametralPitch(x), N1, SpeedB, FaceWidth(x),
 Force1t(x), m1actual, PressureAngle);
    %Pinion 2
    [CP2, BP2] = stresses(DiametralPitch(x), P2, SpeedB, FaceWidth(x),
 Force2t(x), m2actual, PressureAngle);
    %Gear 2
    [CN2, BN2] = stresses(DiametralPitch(x), N2, SpeedC, FaceWidth(x),
 Force2t(x), m2actual, PressureAngle);

    ContactPinion1 = [ContactPinion1, CP1];
    BendingPinion1 = [BendingPinion1, BP1];
    ContactGear1 = [ContactGear1, CN1];
    BendingGear1 = [BendingGear1, BN1];
    ContactPinion2 = [ContactPinion2, CP2];
    BendingPinion2 = [BendingPinion2, BP2];
    ContactGear2 = [ContactGear2, CN2];
    BendingGear2 = [BendingGear2, BN2];

end

ContactPinion1 = transpose(ContactPinion1(2:end));
BendingPinion1 = transpose(BendingPinion1(2:end));
ContactGear1 = transpose(ContactGear1(2:end));
BendingGear1 = transpose(BendingGear1(2:end));
ContactPinion2 = transpose(ContactPinion2(2:end));
BendingPinion2 = transpose(BendingPinion2(2:end));
ContactGear2 = transpose(ContactGear2(2:end));
BendingGear2 = transpose(BendingGear2(2:end));

StressTable = table(transpose(DiametralPitch), ContactPinion1,
 BendingPinion1, ContactGear1, BendingGear1, ContactPinion2,
 BendingPinion2, ContactGear2, BendingGear2)
```

```
%{
Pd = Diametral Pitch
N = Number of Teeth on pinion
n = Angular Velocity (rpm)
F = Face Width
Wt = Tangetntial load
m = gear Ratio
phi = pressure angle
ht = tooth height
tr = rim thickness
%}
function [CS, BS] = stresses(Pd, N, n, F, Wt, m, phi)
    dp = N/Pd;              %Diameter inches
    V = pi*dp*n/12;         %Linear Velocity

    %Ko - Overload Factor
    Ko = 1.2; %Moderate Shock

    %Kv - Dynamic Factor
    Qv = 9; %Low end of precision quality gears, Catalog: pg. 48, AGMA
 class: 9
    B = 0.25*(12 - Qv)^(2/3);
    A = 50 + 56*(1 - B);
    Kv = ((A + V^0.5)/A)^B;

    %Ks - Size Factor
    x = 1.5708/Pd; %Tooth Width, equation from boston
    Y = 2*x*Pd/3;
    Ks = 1.192*((F*Y^0.5)/Pd)^0.0535;

    %Km - Load Distribution Factor
    Cmc = 0.8; %Crowned
    Cpf = F/(10*dp) - 0.025;
    Cpm = 1;
    A = 0.0675; %Precision, enclosed, Table 14-9
    B = 0.0128;
    C = -0.926E-4;
    Cma = A + B*F + C*F^2;
    Ce = 1;
    Km = 1 + Cmc*(Cpf*Cpm + Cma*Ce);

    %Kb - Rim Thickness Factor
    Kb = 1; %Assumption given by professor
    %{
    if Pd < 20 %Whole Depth Equation from Boston
        ht = 2.157/Pd;
    else
        ht = 2.2/Pd + 0.002;
    end

    tr = %Cannot find equation or values specifying rim thickness

    mb = tr/ht;
```

```
    if mb < 1.2
        Kb = 1.6*log(2.242/mb);
    else
        Kb = 1;
    end
    %}

    %J - Bending Strength Geometry Factor
    J = 0.275; %Figure 14-6

    %Kt - Temperature Factor
    Kt = 1;

    %Kr - Reliability Factor
    Kr = 1; %0.99 - Eq. 14-38

    %Cp - Elastic Coefficient
    Cp = 2300; %Table 14-8 steel-steel

    %I - Pitting Resistance Geometry Factor
    mn = 1; %Load sharing ratio: 1 for spur gears
    I = ((cos(phi)*sin(phi))/(2*mn))*(m/(m+1));

    %Cf - surface condition factor
    Cf = 1;

    %Zn - Stress-Cycle Factor
    Zn = 1;

    %Contact Stress
    CS = (Cp*sqrt(Wt*Ko*Kv*Ks*(Km/(dp*F))*(Cf/
I)))*0.00689476; %Contact stress in psi -> MPa

    %Bending Stress
    BS = (Wt*Ko*Kv*Ks*(Pd/F)*((Km*Kb)/J))*0.00689476; %Contact stress
 in psi -> MPa
end
```

StressTable =

8×9 table

Var1	*ContactPinion1*	*BendingPinion1*	*ContactGear1*	*BendingGear1*	*ContactPinion2*	*BendingPinion2*	*ContactGear2*	*BendingGear2*
8	*429.8*	*39.46*	*188.67*	*38.021*	*855.07*	*149.94*	*420.16*	*144.81*

```
    10          584.51              72.982            256.58
70.314            1169.3              280.37            574.53
270.76
    12          788.34              132.76            346.67
128.36            1583.9               514.5            779.57
498.52
    16            1239              327.95            545.84
318.22            2506.2                1288            1235.5
1252.2
    20          1740.6              647.17            767.47
629.12            3538.1              2567.2            1745.8
2500
    24          2485.5              1319.7              1099
 1290            5072.1              5275.7            2509.1
5164.4
    32          3713.3              2945.5            1641.8
 2879            7622.1               11914            3770.5
11662
    48          6572.1              9226.7            2905.8
9018.9             13593               37891            6724.3
37090
```

Published with MATLAB® R2018b