MEA313 Automotive Theory

Major Task 1

Amal Amr AbdelHamid Zewita
19P1504

Car Used: 1969 Ford Mustang Sportsroof 200 SIX

Table of Contents

1.0 FORMULA FOR ENGINE BRAKE TORQUE AND POWER

1.1 ENGINE BRAKE TORQUE

Code:

```
Ne = 1000:100:4300; % rotational speed array
% torque array for each Ne
Te = [112.9, 136.1, 157.2, 176.4, 193.5, 208.6, 221.7, 232.8, 241.9, 248.9,
254, 257, 258, 257.8, 257.3, 256.5, 255.4, 253.9, 252.1, 250, 247.5, 244.7,
241.6, 238.2, 234.4, 230.3, 225.9, 221.2, 216.1, 209.7, 202, 193.1, 183,
171.9];
% find the coefficients of brake engine torque polynomial
T = polyfit(Ne, Te, 2);
% brake engine torque equation
Te = T(1) * Ne.^2 + T(2) .* Ne + T(3);
```

Torque Equation:

$$T_e = (-4.0456 \times 10^{-5})Ne^2 + 0.2214Ne - 41.7226$$

1.2 ENGINE BRAKE POWER

Code:

```
Ne = 1000:100:4300; % rotational speed array
% power array for each Ne
Pe = [11.8, 15.7, 19.8, 24, 28.4, 32.8, 37.1, 41.4, 45.6, 49.5, 53.2, 56.5,
59.4, 62.1, 64.7, 67.2, 69.5, 71.8, 73.9, 75.9, 77.8, 79.4, 81, 82.3, 83.5,
84.4, 85.2, 85.7, 86, 85.6, 84.6, 82.9, 80.5, 77.4];
% find the coefficients of brake engine power polynomial
P = polyfit(Ne, 1.4.*Pe, 3);
% brake engine power equation
Pe = P(1) * Ne.^3 + P(2) * Ne.^2 + P(3) .* Ne + P(4);
```

Power Equation:

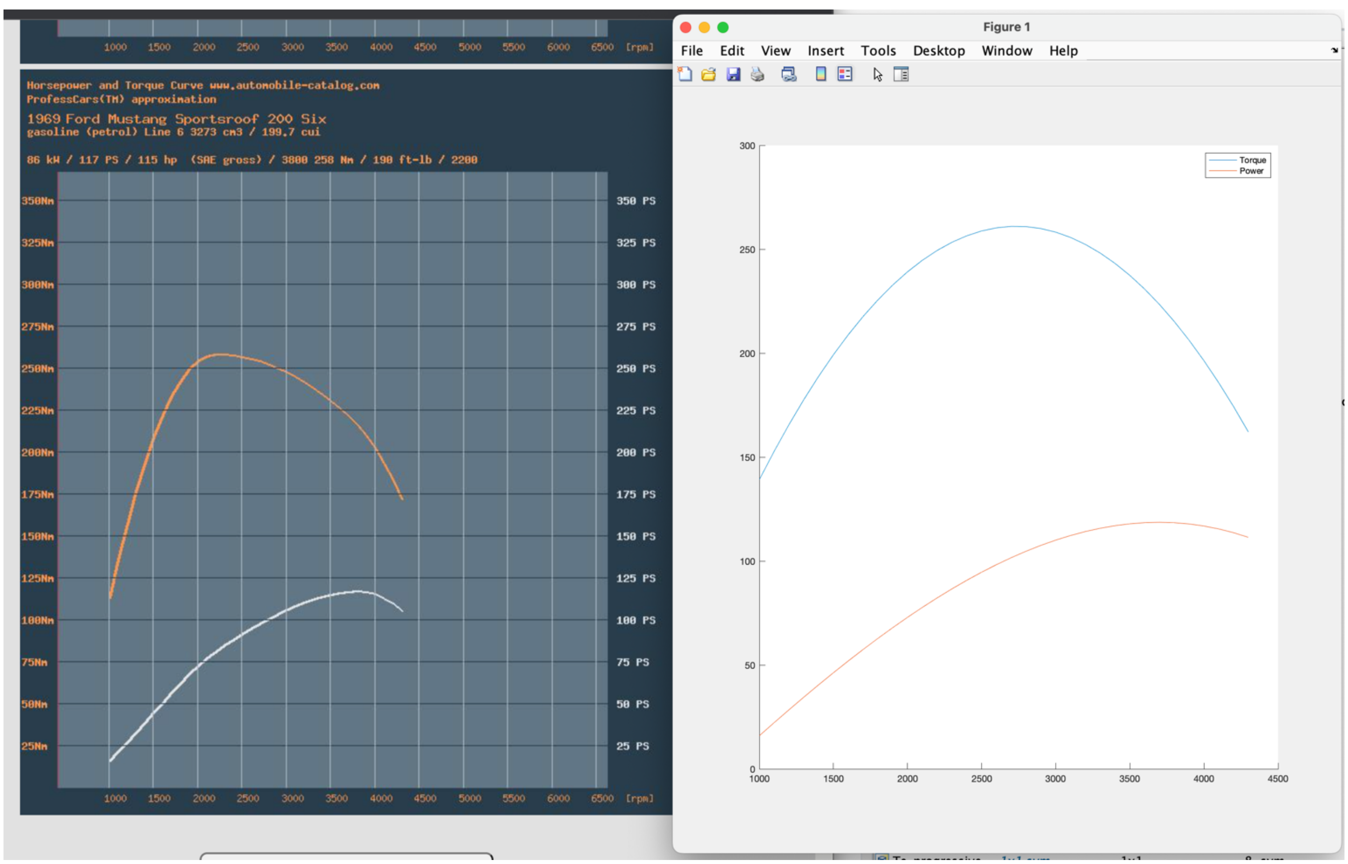
$$P_e = (-1.754586 \times 10^{-9})Ne^3 + (6.14067 \times 10^{-7})Ne^2 + 0.0674Ne - 50.2884$$

1.3 Plotting Both Engine Brake Torque and Power

Code:

```
% configure plot
f = figure;
ax = axes(f);
ax.NextPlot = "add";
lg = legend(ax);

% plotting
plot(ax, Ne, Te, DisplayName = 'Torque');

% plotting
plot(ax, Ne, Pe, DisplayName = 'Power');
```

Graph:



3.0 DESIGNING A GEARBOX

To design a gearbox you must firstly get the highest and lowest gear shit.

To get the highest gear shift…

Code :

```
% specs
sympref('FloatingPointOutput', true)
Mass = 1295; % car weight
Cd = 0.5; % drag coefficient
iFinal = 3.08; % final drive
Ro = 1.23; % air density
Rt = 0.3175; % tire radius
Aft = 1.96; % frontal area
Vmax = 161; % top speed (km/hr)
Fr = 0.015; % rolling coefficient
Eff = 0.95; % mechanical efficiency
g = 9.81; % gravitational acceleration

% highest gear shift
syms NE
Te = T(1) * NE.^2 + T(2) .* NE + T(3);
% calculate resistive pressure
Pres = ((0.5 * Ro * Cd * Aft * (Vmax / 3.6)^2) + Fr * g * Mass) * Vmax / 3.6;
% double to convert from class to class
soln = double(solve(Te * 2 * pi * NE / 60 * Eff == Pres, NE));
Ne2 = soln(2);
ig5 = (2 * pi * Ne2 * 3.6 * Rt) / (60 * Vmax * iFinal);
```

Highest Gear Shift:

$$i_{g_5} = 0.588$$

To get the lowest gear shift…

Code :

```
% specs
Mass = 1295; % car weight
Fr = 0.015; % rolling coefficient
iFinal = 3.08; % final drive
Rt = 0.3175; % tire radius
u = 0.9; % coff of friction
Eff = 0.95; % mechanical efficiency
h = 0.781; % height of CG
Kp = 0.9; % progressive coefficient
g = 9.81; % gravitational acceleration
% wheelbase
L = 2.743;
l1 = 1.454;
l2 = 1.289;

% lowest gear shift
Te_max = 258;
% RWD constant
kr = ((l1 - h * Fr) / (L - u * h));
% max tractive effort
Ft_max = kr * u * Mass * g;
ig1 = Ft_max * Rt / iFinal / Eff / Te_max;
```

Lowest Gear Shift:

$$i_{g_1} = 3.3996$$

3.1 Progressive Gearbox

Code:

```
% specs
N = 5; % number of gears

Kg = nthroot (ig1 / ig5, N - 1);
% progressive gearbox
% get the C(s)
C1 = Kg * Kp^(1 - N / 2);
C2 = C1 * Kp;
C3 = C2 * Kp;
% get the ig(s)
ig2 = ig1 / C1;
ig3 = ig2 / C2;
ig4 = ig3 / C3;
```

Progressive Gear Shits:

$i_{g_2} = 1.872$ $i_{g_3} = 1.146$ $i_{g_4} = 0.779$

3.2 Geometric Gearbox

Code:

```
% specs
N = 5; % number of gears

Kg = nthroot (ig1 / ig5, N - 1);
% geometric gearbox
% get the ig(s)
ig_2 = ig1 / Kg;
ig_3 = ig_2 / Kg;
ig_4 = ig_3 / Kg;
```

Geometric Gear Shits:

$i_{g_2} = 2.193$ $i_{g_3} = 1.414$ $i_{g_4} = 0.912$

4.0 UPSHIFT FORWARD SPEED AT 4th SHIFT

4.1 Progressive Gearbox

Code:

```
% specs
Cd = 0.5; % drag coefficient
iFinal = 3.08; % final drive
Rt = 0.3175; % tire radius
Eff = 0.95; % mechanical efficiency
syms V % velocity for the 4th gear shift
Ne = 1000:100:4300; % rotational speed array


% get Ne3, 4
Ne3 = 60 * iFinal * ig3 * V / 2 / pi / Rt / 3.6;
Ne4 = 60 * iFinal * ig4 * V / 2 / pi / Rt / 3.6;
% find Te in terms of v for both 3rd and 4th shifts
Te3 = T(1) * Ne3.^2 + T(2) .* Ne3 + T(3);
Te4 = T(1) * Ne4.^2 + T(2) .* Ne4 + T(3);
% find Ft in terms of v for both 3rd and 4th shifts
Ft3 = Te3 * ig3 * iFinal * Eff / Rt;
Ft4 = Te4 * ig4 * iFinal * Eff / Rt;
% calculate upshift velocity
Sol = solve(Ft3 == Ft4);
Vshift = max(double(Sol));
```

Upshift Forward Speed for Progressive Gearbox:

$$v_{\text{progressive}} = 141.6412\ km/hr$$

4.2 Geometric Gearbox

Code:

```
% specs
Cd = 0.5; % drag coefficient
iFinal = 3.08; % final drive
Rt = 0.3175; % tire radius
Eff = 0.95; % mechanical efficiency
syms V % velocity for the 4th gear shift
Ne = 1000:100:4300; % rotational speed array

% get Ne3, 4
Ne_3 = 60 * iFinal * ig_3 * V / 2 / pi / Rt / 3.6;
Ne_4 = 60 * iFinal * ig_4 * V / 2 / pi / Rt / 3.6;
% find Te in terms of v for both 3rd and 4th shifts
Te_3 = T(1) * Ne_3.^2 + T(2) .* Ne_3 + T(3);
Te_4 = T(1) * Ne_4.^2 + T(2) .* Ne_4 + T(3);
% find Ft in terms of v for both 3rd and 4th shifts
Ft_3 = Te_3 * ig_3 * iFinal * Eff / Rt;
Ft_4 = Te_4 * ig_4 * iFinal * Eff / Rt;
% calculate upshift velocity
Sol = solve(Ft_3 == Ft_4);
V_shift = max(double(Sol));
```

Upshift Forward Speed for Geometric Gearbox:

$$v_{\text{geometric}} = 116.7569\ km/hr$$

4.3 Comparison Between Progressive and Geometric Upshift Forward Speed

The progressive velocity is greater than the geometric velocity:

$$v_{\text{progressive}} = 141.6412\ km/hr > v_{\text{geometric}} = 116.7569\ km/hr$$

5.0 TIME TAKEN AND TRAVELED DISTANE DURING 4th SHIFT

5.1 Progressive Gearbox

Code:

```
% specs
sympref('FloatingPointOutput', true)
Mass = 1295; % car weight
Cd = 0.5; % drag coefficient
iFinal = 3.08; % final drive
Ro = 1.23; % air density
Rt = 0.3175; % tire radius
Aft = 1.96; % frontal area
Vmax = 161; % top speed (km/hr)
Fr = 0.015; % rolling coefficient
Eff = 0.95; % mechanical efficiency
g = 9.81; % gravitational acceleration
syms V % velocity for the 4th gear shift
Ne = 1000:100:4300; % rotational speed array

% calculate total resistance
Tr_progressive = (0.5 * Ro * Cd * Aft * (V / 3.6)^2) + (Fr * 9.81 * Mass);
Te_progressive = T(1) * (NE.^2) + T(2) .* NE + T(3);
Fe_progressive = Te_progressive * ig3 * iFinal * Eff / Rt;
% substitute NE with...
Fe_progressive = subs(Fe_progressive, NE, 60 * iFinal * ig4 * V / 2 / pi /
Rt);
% calculate time
t_progressive = double(vpaintegral(Mass / (Fe_progressive - Tr_progressive),
[110, 140] / 3.6));
% calculate distance
Vavg = (110 + 140) / 3.6 / 2;
s_progressive = Vavg * t_progressive;
```

Time Taken and Travelled Distance in Progressive Gearbox at limits 110 & 140 km/hr:

$$t_{\text{progressive}} = 4.3634\ s$$

$$s_{\text{progressive}} = 151.50694\ m$$

5.2 Geometric Gearbox

Code:

```
% specs
sympref('FloatingPointOutput', true)
Mass = 1295; % car weight
Cd = 0.5; % drag coefficient
iFinal = 3.08; % final drive
Ro = 1.23; % air density
Rt = 0.3175; % tire radius
Aft = 1.96; % frontal area
Vmax = 161; % top speed (km/hr)
Fr = 0.015; % rolling coefficient
Eff = 0.95; % mechanical efficiency
g = 9.81; % gravitational acceleration
syms V % velocity for the 4th gear shift
Ne = 1000:100:4300; % rotational speed array

% calculate total resistance
Tr_geometric = (0.5 * Ro * Cd * Aft * (V / 3.6)^2) + (Fr * 9.81 * Mass);
Te_geometric = T(1) * (NE.^2) + T(2) .* NE + T(3);
Fe_geometric = Te_geometric * ig3 * iFinal * Eff / Rt;
% substitute NE with...
Fe_geometric = subs(Fe_geometric, NE, 60 * iFinal * ig4 * V / 2 / pi / Rt);
% calculate time
t_geometric = double(vpaintegral(Mass / (Fe_geometric - Tr_geometric), [85,
115] / 3.6));
% calculate distance
Vavg = (85 + 115) / 3.6 / 2;
s_geometric = Vavg * t_geometric;
```

Time Taken and Travelled Distance in Geometric Gearbox at limits 85 & 115 km/hr:

$$t_{\text{geometric}} = 4.7265\ s$$

$$s_{\text{geometric}} = 131.2917\ m$$

5.3 Comparison Between Progressive and Geometric Time Taken and Travelled Distance

The time taken in progressive gearbox is less than geometric gearbox; however, the distance travelled in progressive gearbox is greater than geometric gearbox.

$$t_{\text{progressive}} = 4.3634\ s < t_{\text{geometric}} = 4.7265\ s$$

$$s_{\text{progressive}} = 151.50694\ m > s_{\text{geometric}} = 131.2917\ m$$

6.0 MATLAB

Code:

```
clearvars
clc

%%% 1969 Ford Mustang Sportsroof 200 Six %%%

% configure plot
f = figure;
ax = axes(f);
ax.NextPlot = "add";
lg = legend(ax);

% specs
sympref('FloatingPointOutput', true)
Mass = 1295; % car weight
Cd = 0.5; % drag coefficient
iFinal = 3.08; % final drive
Ro = 1.23; % air density
Rt = 0.3175; % tire radius
Aft = 1.96; % frontal area
Vmax = 161; % top speed (km/hr)
Fr = 0.015; % rolling coefficient
Eff = 0.95; % mechanical efficiency
g = 9.81; % gravitational acceleration
u = 0.9; % coff of friction
h = 0.781; % height of CG
Kp = 0.9; % progressive coefficient
N = 5; % number of gears
syms V % velocity for the 4th gear shift
Ne = 1000:100:4300; % rotational speed array
% wheelbase
L = 2.743;
l1 = 1.454;
l2 = 1.289;
```

```
% torque array for each Ne
Te = [112.9, 136.1, 157.2, 176.4, 193.5, 208.6, 221.7, 232.8, 241.9, 248.9,
254, 257, 258, 257.8, 257.3, 256.5, 255.4, 253.9, 252.1, 250, 247.5, 244.7,
241.6, 238.2, 234.4, 230.3, 225.9, 221.2, 216.1, 209.7, 202, 193.1, 183,
171.9];
% find the coefficients of brake engine torque polynomial
T = polyfit(Ne, Te, 2);
% brake engine torque equation
Te = T(1) * Ne.^2 + T(2) .* Ne + T(3);
% plotting
plot(ax, Ne, Te, DisplayName = 'Torque');

% power array for each Ne
Pe = [11.8, 15.7, 19.8, 24, 28.4, 32.8, 37.1, 41.4, 45.6, 49.5, 53.2, 56.5,
59.4, 62.1, 64.7, 67.2, 69.5, 71.8, 73.9, 75.9, 77.8, 79.4, 81, 82.3, 83.5,
84.4, 85.2, 85.7, 86, 85.6, 84.6, 82.9, 80.5, 77.4];
% find the coefficients of brake engine power polynomial
P = polyfit(Ne, 1.4.*Pe, 3);
% brake engine power equation
Pe = P(1) * Ne.^3 + P(2) * Ne.^2 + P(3) .* Ne + P(4);
% plotting
plot(ax, Ne, Pe, DisplayName = 'Power');

% highest gear shift
syms NE
Te = T(1) * NE.^2 + T(2) .* NE + T(3);
% calculate resistive pressure
Pres = ((0.5 * Ro * Cd * Aft * (Vmax / 3.6)^2) + Fr * g * Mass) * Vmax / 3.6;
% double to convert from class to class
soln = double(solve(Te * 2 * pi * NE / 60 * Eff == Pres, NE));
Ne2 = soln(2);
ig5 = (2 * pi * Ne2 * 3.6 * Rt) / (60 * Vmax * iFinal);

% lowest gear shift
Te_max = 258;
% RWD constant
```

```
kr = ((l1 - h * Fr) / (L - u * h));
% max tractive effort
Ft_max = kr * u * Mass * g;
ig1 = Ft_max * Rt / iFinal / Eff / Te_max;

Kg = nthroot (ig1 / ig5, N - 1);

% progressive gearbox
% get the C(s)
C1 = Kg * Kp^(1 - N / 2);
C2 = C1 * Kp;
C3 = C2 * Kp;
% get the ig(s)
ig2 = ig1 / C1;
ig3 = ig2 / C2;
ig4 = ig3 / C3;
% get Ne3, 4
Ne3 = 60 * iFinal * ig3 * V / 2 / pi / Rt / 3.6;
Ne4 = 60 * iFinal * ig4 * V / 2 / pi / Rt / 3.6;
% find Te in terms of v for both 3rd and 4th shifts
Te3 = T(1) * Ne3.^2 + T(2) .* Ne3 + T(3);
Te4 = T(1) * Ne4.^2 + T(2) .* Ne4 + T(3);
% find Ft in terms of v for both 3rd and 4th shifts
Ft3 = Te3 * ig3 * iFinal * Eff / Rt;
Ft4 = Te4 * ig4 * iFinal * Eff / Rt;
% calculate upshift velocity
Sol = solve(Ft3 == Ft4);
Vshift = max(double(Sol));
% calculate total resistance
Tr_progressive = (0.5 * Ro * Cd * Aft * (V / 3.6)^2) + (Fr * 9.81 * Mass);
Te_progressive = T(1) * (NE.^2) + T(2) .* NE + T(3);
Fe_progressive = Te_progressive * ig3 * iFinal * Eff / Rt;
% substitute NE with...
Fe_progressive = subs(Fe_progressive, NE, 60 * iFinal * ig4 * V / 2 / pi /
Rt);
% calculate time
```

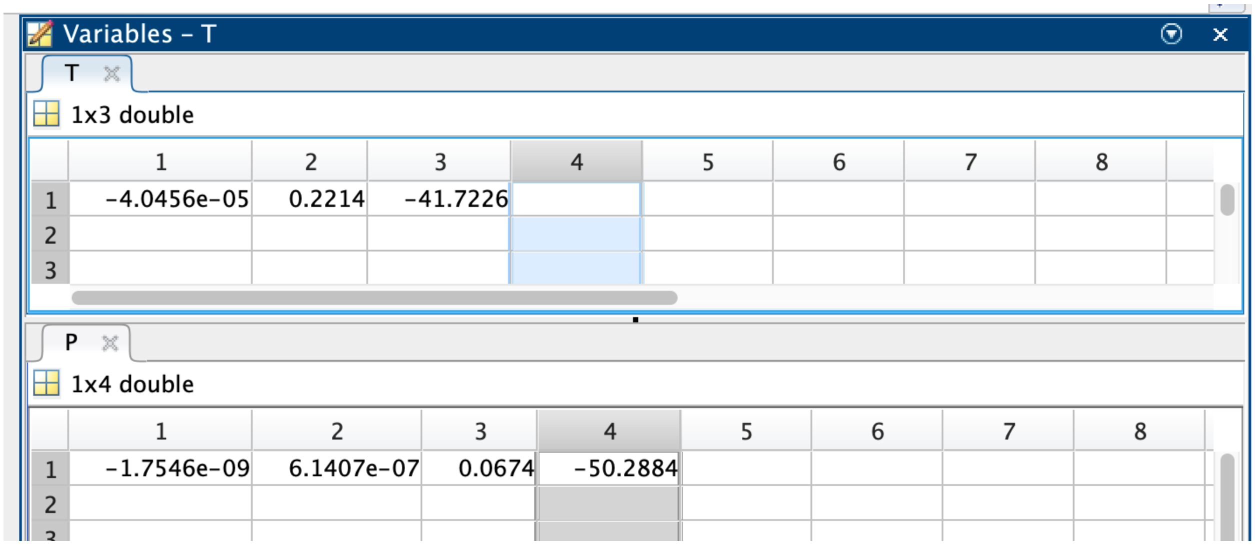
```
t_progressive = double(vpaintegral(Mass / (Fe_progressive - Tr_progressive),
[110, 140] / 3.6));
% calculate distance
Vavg = (110 + 130) / 3.6 / 2;
s_progressive = Vavg * t_progressive;

% geometric gearbox
% get the ig(s)
ig_2 = ig1 / Kg;
ig_3 = ig_2 / Kg;
ig_4 = ig_3 / Kg;
% get Ne3, 4
Ne_3 = 60 * iFinal * ig_3 * V / 2 / pi / Rt / 3.6;
Ne_4 = 60 * iFinal * ig_4 * V / 2 / pi / Rt / 3.6;
% find Te in terms of v for both 3rd and 4th shifts
Te_3 = T(1) * Ne_3.^2 + T(2) .* Ne_3 + T(3);
Te_4 = T(1) * Ne_4.^2 + T(2) .* Ne_4 + T(3);
% find Ft in terms of v for both 3rd and 4th shifts
Ft_3 = Te_3 * ig_3 * iFinal * Eff / Rt;
Ft_4 = Te_4 * ig_4 * iFinal * Eff / Rt;
% calculate upshift velocity
Sol = solve(Ft_3 == Ft_4);
V_shift = max(double(Sol));
% calculate total resistance
Tr_geometric = (0.5 * Ro * Cd * Aft * (V / 3.6)^2) + (Fr * 9.81 * Mass);
Te_geometric = T(1) * (NE.^2) + T(2) .* NE + T(3);
Fe_geometric = Te_geometric * ig3 * iFinal * Eff / Rt;
% substitute NE with...
Fe_geometric = subs(Fe_geometric, NE, 60 * iFinal * ig4 * V / 2 / pi / Rt);
% calculate time
t_geometric = double(vpaintegral(Mass / (Fe_geometric - Tr_geometric), [90,
116] / 3.6));
% calculate distance
Vavg = (110 + 130) / 3.6 / 2;
s_geometric = Vavg * t_geometric;
```

Workspace:

Workspace

Name	Value	Size	Bytes	Class
Af	1.9600	1x1	8	double
ans	1	1x1	1	logical
ax	1x1 Axes	1x1	8	matlab.graphics.axis.Axes
C1	1.8158	1x1	8	double
C2	1.6343	1x1	8	double
C3	1.4708	1x1	8	double
Cd	0.5000	1x1	8	double
Eff	0.9500	1x1	8	double
f	1x1 Figure	1x1	8	matlab.ui.Figure
Fe_geometric	1x1 sym	1x1	8	sym
Fe_progressive	1x1 sym	1x1	8	sym
Fr	0.0150	1x1	8	double
Ft3	1x1 sym	1x1	8	sym
Ft4	1x1 sym	1x1	8	sym
Ft_3	1x1 sym	1x1	8	sym
Ft_4	1x1 sym	1x1	8	sym
Ft_max	8.0832e+03	1x1	8	double
g	9.8100	1x1	8	double
h	0.7810	1x1	8	double
iFinal	3.0800	1x1	8	double
ig1	3.3996	1x1	8	double
ig2	1.8722	1x1	8	double
ig3	1.1456	1x1	8	double
ig4	0.7789	1x1	8	double
ig5	0.5884	1x1	8	double
ig_2	2.1927	1x1	8	double
ig_3	1.4143	1x1	8	double
ig_4	0.9122	1x1	8	double
Kg	1.5504	1x1	8	double
Kp	0.9000	1x1	8	double
kr	0.7070	1x1	8	double
L	2.7430	1x1	8	double
l1	1.4540	1x1	8	double
l2	1.2890	1x1	8	double
lg	1x1 Legend	1x1	8	matlab.graphics.illustration.Legend
Mass	1295	1x1	8	double
N	5	1x1	8	double
Ne	1x34 double	1x34	272	double
NE	1x1 sym	1x1	8	sym
Ne2	2.4376e+03	1x1	8	double
Ne3	1x1 sym	1x1	8	sym
Ne4	1x1 sym	1x1	8	sym
Ne_3	1x1 sym	1x1	8	sym
Ne_4	1x1 sym	1x1	8	sym
P	[-1.7546e-09,6....	1x4	32	double
Pe	1x34 double	1x34	272	double
Pres	6.2432e+04	1x1	8	double
Ro	1.2300	1x1	8	double
Rt	0.3175	1x1	8	double
s_geometric	134.2533	1x1	8	double
s_progressive	145.4467	1x1	8	double
Sol	2x1 sym	2x1	8	sym
soln	[-1.4267e+03;2....	3x1	24	double
T	[-4.0456e-05,0....	1x3	24	double
t_geometric	4.0276	1x1	8	double
t_progressive	4.3634	1x1	8	double
Te	1x1 sym	1x1	8	sym
Te3	1x1 sym	1x1	8	sym
Te4	1x1 sym	1x1	8	sym
Te_3	1x1 sym	1x1	8	sym
Te_4	1x1 sym	1x1	8	sym
Te_geometric	1x1 sym	1x1	8	sym
Te_max	258	1x1	8	double
Te_progressive	1x1 sym	1x1	8	sym
Tr_geometric	1x1 sym	1x1	8	sym
Tr_progressive	1x1 sym	1x1	8	sym
u	0.9000	1x1	8	double
V	1x1 sym	1x1	8	sym
V_shift	116.7569	1x1	8	double
Vavg	33.3333	1x1	8	double
Vmax	161	1x1	8	double
Vshift	141.6412	1x1	8	double

Torque and Power Coefficients:



Variables – T

T

1x3 double

	1	2	3	4	5	6	7	8
1	-4.0456e-05	0.2214	-41.7226					
2								
3								

P

1x4 double

	1	2	3	4	5	6	7	8
1	-1.7546e-09	6.1407e-07	0.0674	-50.2884				
2								

Drive link for the m file:

https://drive.google.com/drive/folders/1QPjeBvkBbXVYi3Szo4-V2yE6wt-uoyY6?usp=sharing