

---

```

clc;
clear all;
close all;

%Raw data
%Ignoring last two data points due to experiment malfunction
rpm = [46500 49300 55000 60000 65100 70500];
Tm2 = [21.559 20.9850 21.1541 21.2047 20.8731 20.9066] + 273.15; %Cross flow
Tm3 = [68.5792 74.8547 86.1913 117.9889 129.9338 148.9507 171.9082]...
+ 273.15; %Cross flow
Tm4 = [503.7978 501.4857 509.7827 535.9689 547.0686 597.3990]...
+ 273.15; %Cross flow
Tm5 = [484.6335 474.2929 482.9551 507.9552 513.6135 556.9158]...
+ 273.15; %Axial flow
Tm8 = [488.5296 484.472 486.4055 494.6287 500.1043 514.3652]...
+ 273.15; %Cross flow
Tm_oil = [45.6508 49.6376 57.1879 64.9292 70.94 75.7677] + 273.15;
dp2 = [0.5141 0.6161 0.8152 1.0346 1.3044 1.6022] * 10^3; %Differential
pt3 = [50.4636 57.6214 73.4094 89.7863 109.6062 132.8239] * 10^3; %Stagnation
p4 = [46.6808 53.9519 69.468 85.8223 105.2571 128.2483] * 10^3; %Static
pt5 = [5.4178 6.1699 7.9933 10.2856 12.6753 15.7876] * 10^3; %Stagnation
pt8 = [3.5038 4.2604 5.9569 7.7169 9.3909 11.2959] * 10^3; %Stagnation
m_dot_fuel = [0.0021 0.0023 0.0025 0.0027 0.0029 0.0032]; %kg/s
thrust = [3.7 3.9 5.1 6 6.5 7.9] * 4.4482216; %N

Po2 = 101.3 * 10^3; % Pa

%Convert pressures from gauge to absolute
pt3 = pt3 + Po2;
p4 = p4 + Po2;
pt5 = pt5 + Po2;
pt8 = pt8 + Po2;

%Given/known information
A1 = 27.3 * 0.00064516;
A2 = 6.4 * 0.00064516;
A3 = 9 * 0.00064516;
A4 = 7.2 * 0.00064516;
A5 = 4.7 * 0.00064516;
A8 = 3.87 * 0.00064516;
RF_c = 0.68;
RF_a = 0.86;

%Time to actually find air m_dot, Ma, U, and rho at state 2
%Assumption - since Ma will be small, T2 = T2_measured ~= T2_actual
[~, ~, k, R] = sp_heats(Tm2, 'air');
Po2_over_P = Po2 ./ (Po2 - dp2);
Ma_2 = sqrt((Po2_over_P.^((k - 1) ./ k) - 1) .* (2 ./ (k - 1)));
U_2 = sqrt(k .* R .* Tm2) .* Ma_2;
rho_2 = (Po2 - dp2) ./ (R .* Tm2);
m_dot = rho_2 .* U_2 .* A2;

```

---

---

```

%Calculate air-fuel ratio
af = m_dot ./ m_dot_fuel;

%Define LHV
LHV = (42800 * 10^3) * 170.145/1000; %converted to J/mol

MM.O2 = 32;
MM.N2 = 28.02;
MM.C = 12.01;
MM.H = 1.008;
MM.H2O = 18.016;
MM.CO2 = 44.01;
MM.JetA = 170.145;

%stoichiometric air fuel and equivalence ratio
AF_s = (17.85 * MM.O2 + 17.85*(79/21) * MM.N2) / (12.3 * MM.C + 22.2 * MM.H);
phi = AF_s ./ af;

for i=1:length(rpm)
    [Ma2(i), To2(i), T2(i), Po2_ratio(i)] = ...
        zachStuart(Tm2(i), Po2, m_dot(i), A2, RF_c, 'air');
    [Ma3(i), To3(i), T3(i), Po3_ratio(i)] = ...
        zachStuart(Tm3(i), pt3(i), m_dot(i), A3, RF_c, 'air');
    %assume static = stagnation pressure at station 4 due to low Ma
    [Ma4(i), To4(i), T4(i), Po4_ratio(i)] = ...
        viggyFresh(Tm4(i), p4(i), m_dot(i) + m_dot_fuel(i), A4, RF_c, phi(i), MM);
    [Ma5(i), To5(i), T5(i), Po5_ratio(i)] = ...
        viggyFresh(Tm5(i), pt5(i), m_dot(i) + m_dot_fuel(i), A5, RF_a, phi(i), MM);
    [Ma8(i), To8(i), T8(i), Po8_ratio(i)] = ...
        viggyFresh(Tm8(i), pt8(i), m_dot(i) + m_dot_fuel(i), A8, RF_c, phi(i), MM);
end

%Find station 1 values
Po1 = Po2;
To1 = To2;
for i = 1:length(rpm)
    [Ma1(i), T1(i), Po1_ratio(i)] = richieTran(To1(i), Po1, m_dot(i), A1);
end

%Recalculate To4 with JetA chemistry things (Part 3)
for i=1:length(rpm)
    To4_JetA(i) = combustor(MM, phi(i), To3(i));
end

%Calcualate To5s_JetA using To4_JetA
To5s_JetA = turb_Ts(To4_JetA, (pt5 ./ p4), length(rpm), 'JetA', phi, MM);

%Calculate speed of sound at each station
[~, ~, gamma1, ~] = sp_heats(T1, 'air');
[~, ~, gamma2, ~] = sp_heats(T2, 'air');
[~, ~, gamma3, ~] = sp_heats(T3, 'air');
[~, ~, gamma4, ~] = sp_heats_JetA(T4, phi, MM);
[~, ~, gamma5, ~] = sp_heats_JetA(T5, phi, MM);
[~, ~, gamma8, ~] = sp_heats_JetA(T8, phi, MM);

```

---

---

```

U1 = Ma1 .* sqrt(gamma1 .* R .* T1);
U2 = Ma2 .* sqrt(gamma2 .* R .* T2);
U3 = Ma3 .* sqrt(gamma3 .* R .* T3);
U4 = Ma4 .* sqrt(gamma4 .* R .* T4);
U5 = Ma5 .* sqrt(gamma5 .* R .* T5);
U8 = Ma8 .* sqrt(gamma8 .* R .* T8);

%Compute all static pressures from stagnation ratios
stag_two = ones(1, length(rpm)) * Po2;
P1 = stag_two ./ Po1_ratio;
P2 = stag_two ./ Po2_ratio;
P3 = pt3 ./ Po3_ratio;
P4 = p4 ./ Po4_ratio;
P5 = pt5 ./ Po5_ratio;
P8 = pt8 ./ Po8_ratio;

%Calculate thrust terms - CV from state 0 to state 8
Ft_calc = ((m_dot + m_dot_fuel) .* U8);
thrust_sp = Ft_calc ./ m_dot;
TSFC = m_dot_fuel ./ Ft_calc;

%Marker size var
markerSize = 10;

%Convert rpm tp krpm
krpm = rpm ./ 1000;

%Plot stagnation temperature vs. rpm (by station)
figure;
plot(krpm, To1, 'x-', 'Color', [0.9,0.4,0.6], 'MarkerSize', markerSize);
hold on
plot(krpm, To2, krpm, To3, krpm, To4, krpm, To4_JetA,...
    krpm, To5, krpm, To5s_JetA, krpm, To8,...
    'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Stagnation Temperature (K)');
title('Stagnation Temperature vs. Spool Speed');
legend('To1', 'To2', 'To3', 'To4', ...
    'To4s (JetA)', 'To5', ...
    'To5s (JetA)', 'To8', ...
    'location', 'bestoutside');
set(gcf, 'color', 'w');

%Plot stagnation pressure vs. krpm (by station)
figure;
plot(krpm, ones(1,length(krpm))*Po1/10^3, krpm, ...
    ones(1,length(krpm))*Po2/10^3, krpm, pt3/10^3, krpm, p4/10^3, krpm, ...
    pt5/10^3, krpm, pt8/10^3, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Stagnation Pressure (KPa, Absolute)');
title('Stagnation Pressure vs. Spool Speed ');
legend('Po1', 'Po2', 'Po3', 'Po4', 'Po5', ...
    'Po8', 'location', 'bestoutside');

```

---

---

```

set(gcf,'color','w');

%Plot mach number vs. krpm (by station)
figure;
plot(krpm, Ma1, krpm, Ma2, krpm, Ma3, krpm, Ma4, krpm, Ma5, krpm, Ma8,...
     'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Mach Number');
title('Mach Number vs. Spool Speed ');
legend('Ma1', 'Ma2', 'Ma3', 'Ma4', 'Ma5', ...
      'Ma8', 'location', 'bestoutside');
set(gcf,'color','w');

%Plot station velocity vs. krpm (by station)
figure;
plot(krpm, U1, krpm, U2, krpm, U3, krpm, U4, krpm, U5, krpm, U8,...
     'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Velocity (m/s)');
title('Velocity vs. Spool Speed');
legend('U1', 'U2', 'U3', 'U4', 'U5', ...
      'U8', 'location', 'bestoutside');
set(gcf,'color','w');

%Plot mass flow rates
figure;
[ax, h1, h2] = plotyy(krpm, m_dot .* 1000, krpm, m_dot_fuel .* 1000);
set(h1,'Marker','o','MarkerSize', markerSize);
set(h2,'Marker','o','MarkerSize', markerSize)
ylabel(ax(1), 'Mass flow of air (g/s)');
ylabel(ax(2), 'Mass flow of fuel (g/s)');
xlabel('Spool Speed (kRPM)');
title('Mass Flow vs. Spool Speed');
set(gcf, 'color', 'white');

%Plot AF vs krpm
figure;
plot(krpm, af, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Air-Fuel Ratio');
title('Air-Fuel Ratio vs. Spool Speed');
set(gcf,'color','w');

%Plot calculated and measured net thrust vs. krpm (by station)
figure;
plot(krpm, Ft_calc, krpm, thrust, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Thrust (N)');
title('Thrust vs. Spool Speed');
legend('Calculated Thrust', 'Measured Thrust', 'location', 'best');
set(gcf,'color','w');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Part 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

---

---

```

%Plot specific thrust vs. krpm (by station)
figure;
plot(krpm, thrust_sp, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Specific Thrust (N*s/kg)');
title('Specific Thrust vs. Spool Speed');
set(gcf, 'color', 'w');

%Plot thrust-specific fuel consumption vs. krpm (by station)
figure;
plot(krpm, TSFC, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Thrust-specific Fuel Consumption (kg/N*s)');
title('Thrust-Specific Fuel Consumption vs. Spool Speed');
set(gcf, 'color', 'w');
set(gca, 'YTickLabel', num2str(get(gca, 'YTick'), '%f'));

%Find Q_dot into system and work out of turbine
lhv = 42800 * 10^3; %J/kg
Q_dot = m_dot_fuel .* lhv;
W_net = (m_dot + m_dot_fuel) .* (U8.^2) ./ 2;
eta_therm = W_net ./ Q_dot;

%Plot thermal efficiency vs. spool speed
figure;
plot(krpm, eta_therm*100, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Thermal Efficiency (%)');
title('Thermal Efficiency vs. Spool Speed');
set(gcf, 'color', 'w');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Part 4%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Power consumed by compressor and produced by turbine
W_dot_comp_actual = m_dot .* deltaH_var_cp(To2, To3, length(rpm), ...
    'air', phi, MM);
W_dot_turb_actual = (m_dot + m_dot_fuel) .* ...
    deltaH_var_cp(To5, To4, length(rpm), 'JetA', phi, MM);

To3s = comp_Ts(To2, (pt3 ./ Po2), length(rpm), 'air', phi, MM);
eta_comp = deltaH_var_cp(To2, To3s, length(rpm), 'air', phi, MM) ...
    ./ deltaH_var_cp(To2, To3, length(rpm), 'air', phi, MM);

To5s = turb_Ts(To4, (pt5 ./ p4), length(rpm), 'JetA', phi, MM);

eta_turb = deltaH_var_cp(To5, To4, length(rpm), 'JetA', phi, MM) ...
    ./ deltaH_var_cp(To5s, To4, length(rpm), 'JetA', phi, MM);

%Constant Air and Variable Air

for i=1:length(rpm)
    [Ma4_c(i), To4_c(i), T4_c(i), Po4_ratio_c(i)] = ...

```

---

---

```

        zachStuart(Tm4(i), p4(i), m_dot(i), A4, RF_c, 'const');
[Ma5_c(i), To5_c(i), T5_c(i), Po5_ratio_c(i)] = ...
        zachStuart(Tm5(i), pt5(i), m_dot(i), A5, RF_a, 'const');
[Ma4_v(i), To4_v(i), T4_v(i), Po4_ratio_v(i)] = ...
        zachStuart(Tm4(i), p4(i), m_dot(i), A4, RF_c, 'air');
[Ma5_v(i), To5_v(i), T5_v(i), Po5_ratio_v(i)] = ...
        zachStuart(Tm5(i), pt5(i), m_dot(i), A5, RF_a, 'air');
end

W_dot_turb_var = (m_dot + m_dot_fuel) .* deltaH_var_cp(To5_v, To4_v, length(rpm),
    'air', phi, MM);
W_dot_turb_const = (m_dot + m_dot_fuel) .* deltaH_var_cp(To5_c, To4_c, length(rpm),
    'const', phi, MM);
W_dot_turb_isen = (m_dot + m_dot_fuel) .* deltaH_var_cp(To5s_JetA, To4_JetA, ...
    length(rpm), 'JetA', phi, MM);

%New plot of turbine power
figure;
plot(krpm, W_dot_turb_actual, krpm, W_dot_turb_var, krpm, W_dot_turb_const,...
    'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Turbine Power (W)');
legend('Products of Combustion', 'Variable c_p of air', ...
    'Constant c_p of air at 300K', 'location', 'best');
title('Turbine Power vs. Spool Speed');
set(gcf, 'color', 'w');
set(gca, 'YTickLabel', num2str(get(gca, 'YTick'), '%.0f'));

%Plot compressor and turbine power vs. spool speed
figure;
plot(krpm, W_dot_comp_actual, krpm, W_dot_turb_actual,...
    krpm, W_dot_turb_isen, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Power (W)');
legend('Compressor Power Consumed', 'Turbine Power Generated', ...
    'Turbine Power Generated (Isentropic, JetA)', 'location', 'best');
title('Power vs. Spool Speed');
set(gcf, 'color', 'w');
set(gca, 'YTickLabel', num2str(get(gca, 'YTick'), '%.0f'));

%Plot stagnation pressure ratio across combustor
combustor_stag_ratio = p4 ./ pt3;
figure;
plot(krpm, combustor_stag_ratio, 'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Stagnation Pressure Ratio');
title('Combustor Stagnation Pressure Ratio vs. Spool Speed');
set(gcf, 'color', 'w');

%Nozzle
T8s = jeannyWang(To5, pt5 ./ (ones(1, length(rpm)) .* Po2),...
    length(rpm), 'JetA', phi, MM);
U8s = sqrt(2 .* deltaH_var_cp(T8s, To5, length(rpm), 'JetA', phi, MM));
eta_nozz = (U8.^2) ./ (U8s.^2);

```

---

---

```

%Plot efficiencies
figure;
plot(krpm, eta_comp*100, krpm, eta_turb*100, krpm, eta_nozz*100,...
     'marker', 'o', 'MarkerSize', markerSize);
xlabel('Spool Speed (kRPM)');
ylabel('Component Efficiency (%)');
legend('Compressor', 'Turbine', 'Nozzle', 'location', 'best')
title('Component Efficiencies vs. Spool Speed');
set(gcf, 'color', 'w');

% PART 2 %
% Calculate heat of formation of JetA
[hf_mol, hf_kg] = heatOfFormation();

% Calculate adiabatic flame temperature wrt. phi
phi_input = 0.05:0.05:0.65;
for i=1:length(phi_input)
    T_a_JetA(i) = flameTemp(phi_input(i), 'JetA', hf_mol, MM);
    T_a_Dodecane(i) = flameTemp(phi_input(i), 'Dodecane', hf_mol, MM);
end

% Plot adiabatic flame temp versus phi
figure;
plot(phi_input, T_a_JetA, 'marker', 'o', 'MarkerSize', markerSize);
hold on;
plot(phi_input, T_a_Dodecane, 'color', [0 1 0], 'marker', 'x',...
     'MarkerSize', markerSize);
xlabel('Equivalence Ratio (phi)');
ylabel('Adiabatic Flame Temperature (K)');
legend('JetA', 'Dodecane', 'location', 'best');
title('Adiabatic Flame Temperature versus Equivalence Ratio');
set(gcf, 'color', 'w');

%PART 3 - moved to other locations hehe

plotfixer;

```

*Published with MATLAB® R2013a*