Table of Contents

ME140 Assignment 1

This assignment asks us to model a jet engine.

Authors: Richard Randall, Beck Goodloe, Jason Trinidad, Miller Perrin Date: 2018 April 7

```
clear all;
clc;
```

TODO

- T0_8s for non-constant SH

Resolved

- Fix h0_6 by solving for enthalpy exiting turbine.
- P0_1 and T0_1 from Ma != 0
- Account for mass flow into front of engine when calculating net thrust.

Constants

```
k = 1.4;
cp = 1005;
R = 287; % [kJ/kg/k] Gas cosntant of air

barToPa = 10^5;
```

Givens

These values are given to us in tables 1 and 2 in the problem statement. They vary for SLS or cruise conditions and need to be manually adjusted for each.

```
% Table 1
T_1 = 218.8; % [K]
P_1 = .239*barToPa; % [Pa]
Ma = .78;
fanCompress_pr = 32; % overall pressure ratio
fan_pr = 1.55;
T0_5 = 1450; % [K]
MDot = 110; % [kg/s]
BPR = 10.0; % bypass ratio

% Table 2
intake_pr = 1.00;
fan_eff = 0.95;
compressor_eff = 0.89;
turbine_eff = 0.90;
nozzle_eff = 0.95;
combustor_pr = 0.95;
pressure_recovery_inlet = .998;
```

Some usefull conversions of givens

```
FracBypass = BPR/(BPR+1);
FracCore = 1/(BPR+1);
compressor_pr = fanCompress_pr/fan_pr;
c = k * R * T_1; % [m/s] speed of sound
```

Specific Heat Equation

This is a polynomial expression for the specific heat given in the textbook in table A-2.

```
tmin = 273;
tmax = 1800;
airCoeffs = fliplr([28.11, 0.1967e-2, 0.4802e-5,-1.966e-9]);
intCoeffs = polyint(airCoeffs);

variable_cp = @(T) 1000*polyval(airCoeffs, T)/28.97;
integral_cp = @(T) 1000*polyval(intCoeffs, T)/28.97;
range_avg_cp = @(T1, T2) (integral_cp(T2) - integral_cp(T1)) / (T2-
T1);
integrate_cp = @(T1, T2) (integral_cp(T2) - integral_cp(T1));
combustor_solution = @(T1,T2) (integral_cp(T2)/(R*T2) -
 integral_cp(T1)/(R*T1));
```

Stage 1: Exterior of Jet

It is easier to calculate the flow of air through the engine if we use stagnation temperature and pressures so that we don't need to know velocity at every stage. Let's first convert the outside air to stagnation values.

```
tempToStag = (1 + (k-1)/2*Ma^2);
pressureToStag = tempToStag^(k/(k-1));
T0_1 = T_1 * tempToStag;
P0_1 = P_1 * pressureToStag;

V_1 = Ma * c; % speed of air outside of jet
H_1 = 0; % take H_1 to be zero (ambient air static enthalpy)
H0_1 = H_1;
```

Stage 2: Right Before Fan (ONLY FOR CRUISE)

```
P0_2 = P0_1*.998;
guess = T0_1;
actual=guess-1;
while (guess-actual)>0
    guess = guess-.01;
    cp_integral = integrate_cp(T0_1,guess);
    cp_avg = cp_integral/(guess-T0_1);
    cv_avg = cp_avg-287;
    gammaavg = cp_avg/cv_avg;
    actual = T0_1*(P0_2/P0_1)^((gammaavg-1)/gammaavg);
end
T0_2s = guess;
T0_2 = T0_2s;
```

Stage 8 :Bypass Air

A vast majority of the air in the turbojet enginer bypasses the core of the engine. It simply flows through the fan and then out a nozzle to rejoin the hot exhaust gas of the core.

```
P0_8 = P0_2*intake_pr*fan_pr;

T0_8s = T0_2*(P0_8/P0_2)^((k-1)/k); %TODO: REPLACE FOR NON-CONSTANT SH

% calculate average cp over temperature range
cp_avg = range_avg_cp(T0_2, T0_8s);

DeltaH_18s = cp_avg*(T0_8s-T0_2);
DeltaH_18 = DeltaH_18s/fan_eff;
T0_8 = T0_2 + DeltaH_18/cp_avg;
H0_8 = DeltaH_18 + H_1;

BypassFanWork = DeltaH_18*FracBypass*MDot;
```

Stage 9: Bypass Nozzle

```
P0_9s = P0_8; % suppose isentropic nozzle
T0_9s = T0_8; % adiabatic
P_9s = P_1;  % must be at atmospheric pressure

% in isentropic case we know P0,T0,P at exit of nozzle. Solve for M
 then
```

```
% use to find T, then use that to find (non-stagnation) enthalpy
 there.
syms M;
eqn_P0_over_P = (1+0.5*(k-1)*M*M)^(k/(k-1)) == P0_9s/P_9s;
M_9s = vpasolve(eqn_P0_over_P,M,[0 Inf]);
T_9s = T0_9s/(1+0.5*(k-1)*M_9s*M_9s);

cp_avg = range_avg_cp(T_1, double(T_9s));
H_9s = double(cp_avg*(T_9s-T_1));
```

Core::Fan

```
P0_3 = P0_2*intake_pr*fan_pr;
guess = T0_2;
actual = T0_2 + 1;
while(guess-actual)<0
    guess = guess+.01;
    cp_avg = range_avg_cp(T0_2,guess);
    cv_avg = cp_avg-287;
    k_avg = cp_avg/cv_avg;
    actual = T0_2*(P0_3/P0_2)^((k-1)/k);
end
T0_3s = guess;

cp_avg = range_avg_cp(T0_2, T0_3s);
DeltaH_13s = cp_avg*(T0_3s - T0_2);
DeltaH_13 = DeltaH_13s/fan_eff; % fan efficiency scales enthalpy
T0_3 = T0_2 + DeltaH_13/cp;

H0_3 = H0_1 + DeltaH_13;
BypassEnthalpy = DeltaH_13;
```

Core::Compressors

We assume that a compressor is isentropic through its flow and therefore the following isentropic relation holds:

```
P0_4 = P0_3*compressor_pr;
guess = T0_3;
actual=guess+1;
while (guess-actual)<0
    guess = guess+.01;
    cp_avg = range_avg_cp(T0_3,guess);
    cv_avg = cp_avg-287;
    gammaavg = cp_avg/cv_avg;
    actual = T0_3*(P0_4/P0_3)^((gammaavg-1)/gammaavg);
end
T0_4s = guess;
T0_4 = T0_3+(T0_4s-T0_3)/compressor_eff;

DeltaH_34 = integrate_cp(T0_3,T0_4);
H0_4 = H0_3+DeltaH_34;
```

```
% P0_4 = P0_3*intake_pr*compressor_pr;
% T0_4s = T0_3*(P0_4/P0_3)^((k-1)/k);
%
% cp_avg = range_avg_cp(T0_3, T0_4s);
% DeltaH_34s = cp_avg*(T0_4s - T0_3);
% DeltaH_34 = DeltaH_34s/compressor_eff;
% T0_4 = T0_3 + DeltaH_34/cp_avg;
%
% H0_4 = H0_3 + DeltaH_34;

CoreEnthalpy = DeltaH_34 + DeltaH_13; %SUPPOSED TO BE 512,540,
 CURRENTLY 509,761
```

Core::Combustor

```
P0_5 = P0_4*combustor_pr;

cp_avg = range_avg_cp(T0_4, T0_5);
DeltaH_45 = (T0_5 - T0_4) * cp_avg;
H0_5 = H0_4 + DeltaH_45;
```

Core::Turbine

The turbine drives both the bypass fan and the compressor. Therefore, we can calculate the enthalpy loss of the turbine by equating it with this work.

```
CoreWork = CoreEnthalpy*MDot*FracCore;
TotalWorkRequired = BypassFanWork+CoreWork;
TotalEnthalpyRequired = CoreEnthalpy+BypassEnthalpy;

guess = T0_5-1;
actual = T0_5-2;
while(actual-guess)<0
    guess = guess-.01;
    cp_avg_temp = range_avg_cp(T0_5,guess);
    actual = T0_5-(TotalWorkRequired/
(turbine_eff*MDot*FracCore*cp_avg_temp));
end
T0_6 = guess;

cp_avg_temp = range_avg_cp(T0_5,T0_6);
DeltaH_56 = cp_avg_temp*(T0_6-T0_5);
cv_avg = cp_avg-287;
k_avg = cp_avg/cv_avg;
P0_6 = P0_5*(T0_6/T0_5)^(k_avg/(k_avg-1));
H0_6 = H0_5+DeltaH_56;
EnthalpyLoss = TotalWorkRequired/(FracCore*MDot);
```

Core::Nozzle

```
P0_7s = P0_6; % suppose isentropic nozzle
```

```
T0_7s = T0_6; % adiabatic
P_7s = P_1; % must be at atmospheric pressure

%in isentropic case we know P0,T0,P at exit of nozzle. Solve for M
 then use
%to find T, then use that to find (non-stagnation) enthalpy there.
syms M



eqn_P0_over_P = (1+0.5*(k-1)*M*M)^(k/(k-1)) == P0_7s/P_7s;
M_7s = vpasolve(eqn_P0_over_P,M,[0 Inf]);
T_7s = T0_7s/(1+0.5*(k-1)*M_7s*M_7s);
cp_avg = range_avg_cp(T_1, double(T_7s));
H_7s = double(cp_avg*(T_7s-T_1));
```

Find Thrust of Engine

The thrust of the engine is given in general by the equation $$ F = \dot{m} (U_{exit - U_{in}) $$. We can brake this up into the component parts of the thrust from the bypassed air and the thrust from the core.

```
Error updating Text.

 Character vector must have valid interpreter syntax:
$$ F = \dot{m} (U_{exit - U_{in}) $$
```

Bypass Thrust

nu = (H0_8 - H_9)/(H0_8 - H_9s) but (H0_8 - H_9s) is the difference between the stagnation enthalpy before the ideal nozzle and the enthalpy remaining after it, i.e. the energy that went into kinetic energy... so nu * that energy is the kinetic energy given to the gas coming out of the non-ideal nozzle.

```
Specific_K_9 = (H0_8 - H_9s)*nozzle_eff;
V_9 = sqrt(2*Specific_K_9);
Bypass_Thrust = (V_9 - V_1) *FracBypass*MDot; % [N]
```

Core Thrust

```
Specific_K_7 = (H0_6 - H_7s)*nozzle_eff;
V_7 = abs(sqrt(2*Specific_K_7));

Core_Thrust = (V_7 - V_1)*FracCore*MDot; % [N]

disp(Core_Thrust + Bypass_Thrust)
```

```
  -7.5136e+06
```

Published with MATLAB® R2017b