Design and Optimization of a Subsonic Turbojet Engine - Takeoff, Climb, and Cruise

—

Muhammad Abdelsallam 201801356

University of Science and Technology - Zewail City
SPC 417 Gas Turbine Engines

Dr. Ahmed Tawfik Eltaweel

Eng. Asmaa AlaaElDeen

Table of Contents

Table of Contents 1

Introduction 2

Methodology 2

- Flow Chart 3
- Cruise optimum Tt4 & Pi_c 4
- Off Design Takeoff 4
- Off Design Climb 5

Results 5

Code 8

References 13

Introduction

It is required to design a turbojet engine that meets the thrust requirements for all mission segments (takeoff, climb, and cruise) for a subsonic transport aircraft with 4 turbojet engines has the following performance data:

Segment	**Takeoff**	**Climb**	**Cruise**
M_0 **(Flight Mach Number)**	0.286	–	0.89
h (Altitude)	Sea Level	6 Km	8.7 Km
Drag Polar	$C_D = 0.01 + 0.065C_L^2$		
Estimated Average Aircraft Weight (kg)	140301.25	137503.75	113725

- Rate of climb=10 m/s at an altitude of 6 km and speed of 216 m/s.
- Rate of climb= 6.35 m/s at sea level and speed of 241 m/s.
- The maximum TIT allowed is 1170 K.
- Assume the engine specific fuel consumption to remain constant during each segment.
- To get the integration limits assume that the total fuel weight is divided to 10% for takeoff and climb 80% for cruise and 10% for loiter and landing.
- $\alpha = 10°$ at sea level.

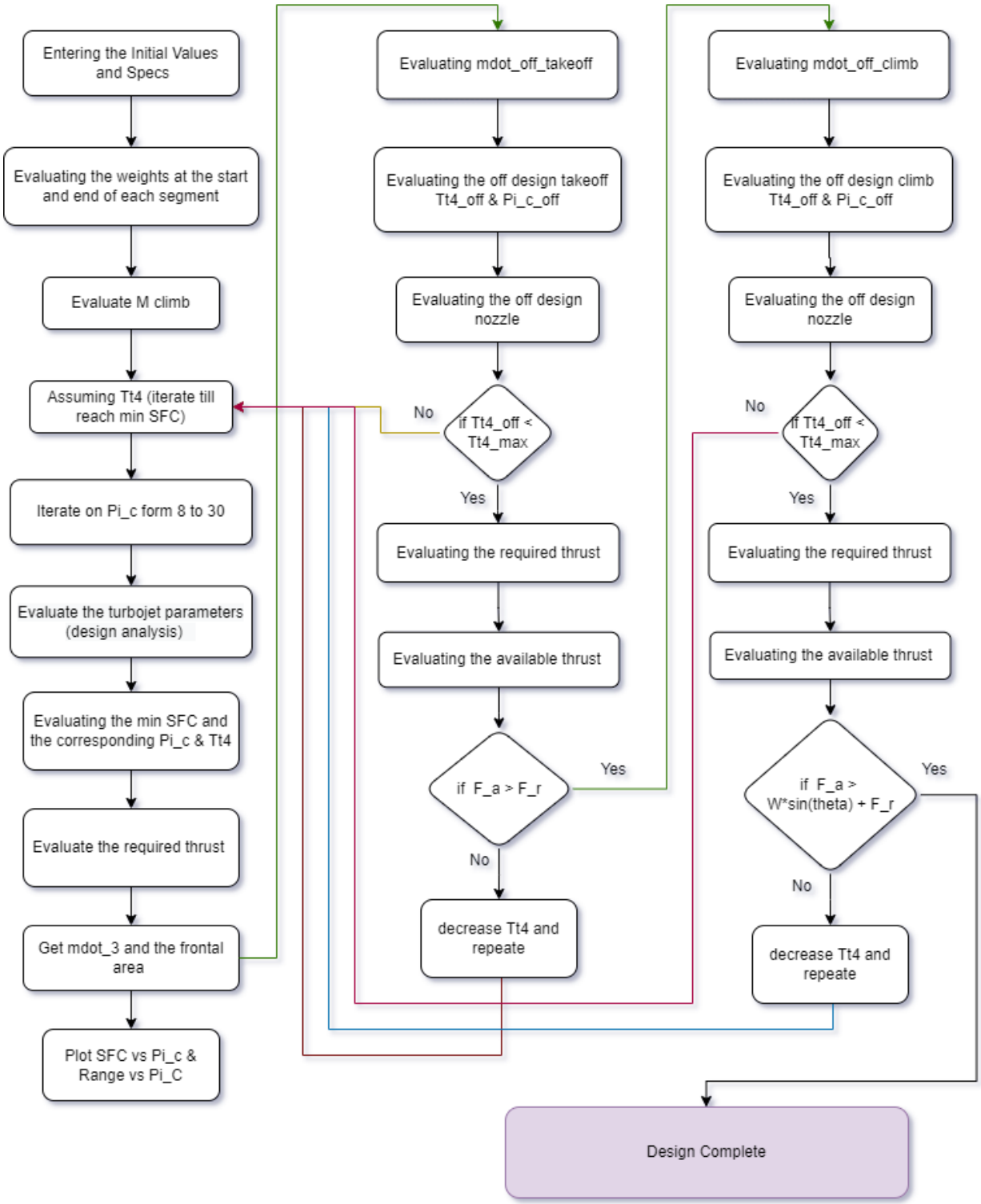
General characteristics of this aircraft:

$$W_{max} = 141700kg \ , \ W_{empty} = 64600kg \ , \ W_{payload} = 21150kg \ \& \ Area_{wing} = 280m^2$$

It is required to select the engine design parameters and size that maximize the aircraft range using the simplified method analysis. Results should contain representative figures of the engine performance at each segment and comparisons between different scenarios should be included. Comparisons between optimal engine for the cruise and optimal engine for the overall mission are also required.

Methodology

Flow Chart


Entering the Initial Values and Specs
Evaluating the weights at the start and end of each segment
Evaluate M climb
Assuming Tt4 (iterate till reach min SFC)
Iterate on Pi_c form 8 to 30
Evaluate the turbojet parameters (design analysis)
Evaluating the min SFC and the corresponding Pi_c & Tt4
Evaluate the required thrust
Get mdot_3 and the frontal area
Plot SFC vs Pi_c & Range vs Pi_C
Evaluating mdot_off_takeoff
Evaluating the off design takeoff Tt4_off & Pi_c_off
Evaluating the off design nozzle
if Tt4_off < Tt4_max
No
Yes
Evaluating the required thrust
Evaluating the available thrust
if F_a > F_r
Yes
No
decrease Tt4 and repeate
Evaluating mdot_off_climb
Evaluating the off design climb Tt4_off & Pi_c_off
Evaluating the off design nozzle
if Tt4_off < Tt4_max
No
Yes
Evaluating the required thrust
Evaluating the available thrust
if F_a > W*sin(theta) + F_r
Yes
No
decrease Tt4 and repeate
Design Complete

Cruise optimum Tt4 & Pi_c

To achieve the maximum range and as indicated by the range formula, the Specific Fuel Consumption (SFC) must be minimum. So the optimization goal of the cruise is to minimize SFC.

$$R = 2\sqrt{\frac{2}{\rho_\infty S}}\frac{1}{c_t}\frac{C_L^{1/2}}{C_D}\left(W_0^{1/2} - W_1^{1/2}\right)$$

Start by taking Taurine Inlet Temperature (TIT) by its maximum value of 1170 K, then evaluate the design turbojet engine cycle analysis for a vector of compressor pressure ratio that ranges from 8 to 30 by a step of 0.01. After getting SFC, minimum SFC is extracted with its corresponding compressor pressure ratio which is the optimum cruise compressor pressure ratio now for that TIT.

Using the specific thrust and the obtained required thrust (form the aerodynamic balance of forces), and as in the case of cruise the required thrust equals the available thrust, the mass flow rate of the engine can be evaluated. Then the frontal area is evaluated from the obtained mass flow rate. Finally, SFC and range are plotted versus the various compressor pressure ratios.

Off Design Takeoff

Starting by calculating the mass flow rate of the takeoff using the relation

$$\dot{m} = \rho A V$$

With the air density at sea level and the velocity of the mid way of the takeoff (corresponding to $M = 0.286$). Then the ram effect is evaluated for the takeoff, and using the simplified off design relations for fixed nozzle area, the new off design TIT and π_c are evaluated.

$$\left(\frac{\dot{m}_2\sqrt{T_{t2}}}{P_{t2}}\right)_{Rel} = \frac{(\pi_c)_{Rel}}{\sqrt{\left(\frac{T_{t4}}{T_{t2}}\right)_{Rel}}}$$

$$(\tau_c - 1)_{Rel} = \left(\frac{T_{t4}}{T_{t2}}\right)_{Rel}$$

Then the new nozzle off-design conditions are evaluated too and checked whether it is choked or unchoked. If the takeoff off design TIT value were higher than the engine max TIT then the engine is re-designed on a lower TIT value from the design state (cruise) again.

Finally and after evaluating the whole off-design cycle during the takeoff, the available and required thrust are calculated and checked if the available thrust is slightly higher than the required thrust. If otherwise, the engine is re-designed on a lower TIT value from the design state (cruise) again.

Off Design Climb

The analysis of the climb-off-design cycle is analogous to that of the takeoff except for the last checking of the available thrust that is replaced by checking that the excess power is slightly higher than **the climb rate** as

$$(P_a - P_r)/W > RC$$

$$(F_a V - F_r V)/W > V sin\theta$$

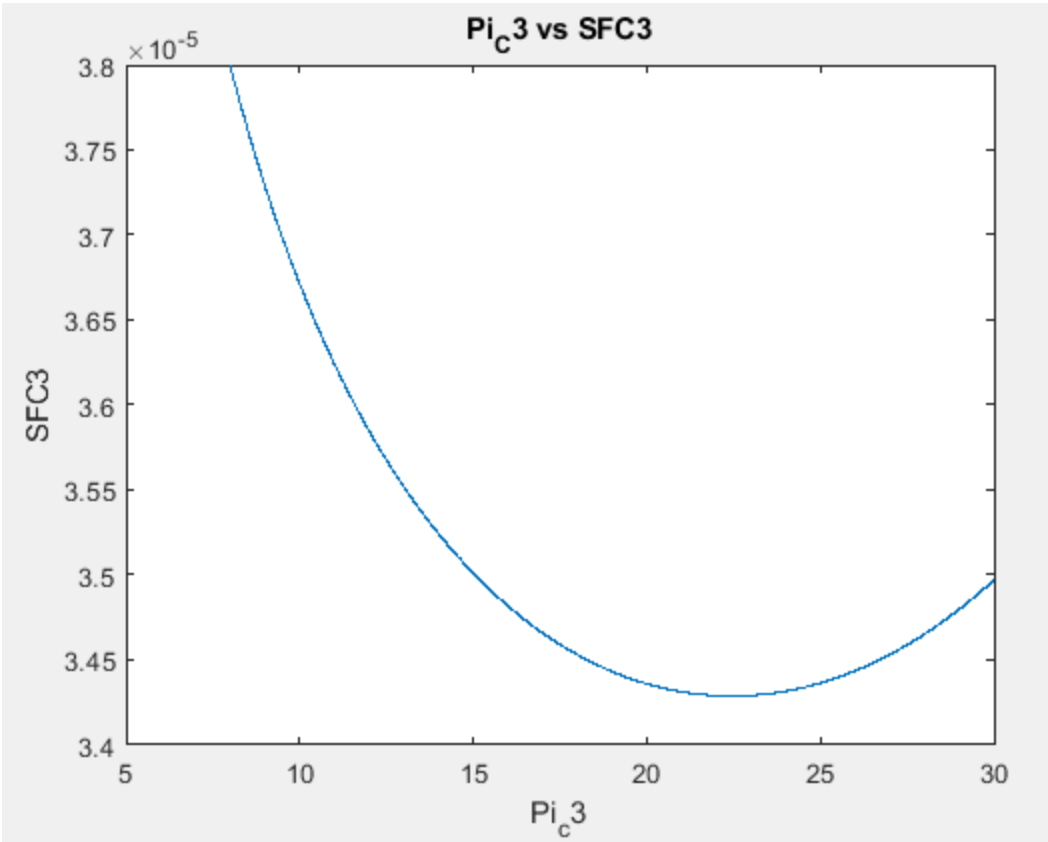
$$(F_a - F_r) > W sin\theta$$

$$F_a > W sin\theta + F_r$$

If the resulting available thrust is not satisfying the constitution, the engine is re-designed on a lower TIT value from the design state (cruise) again.

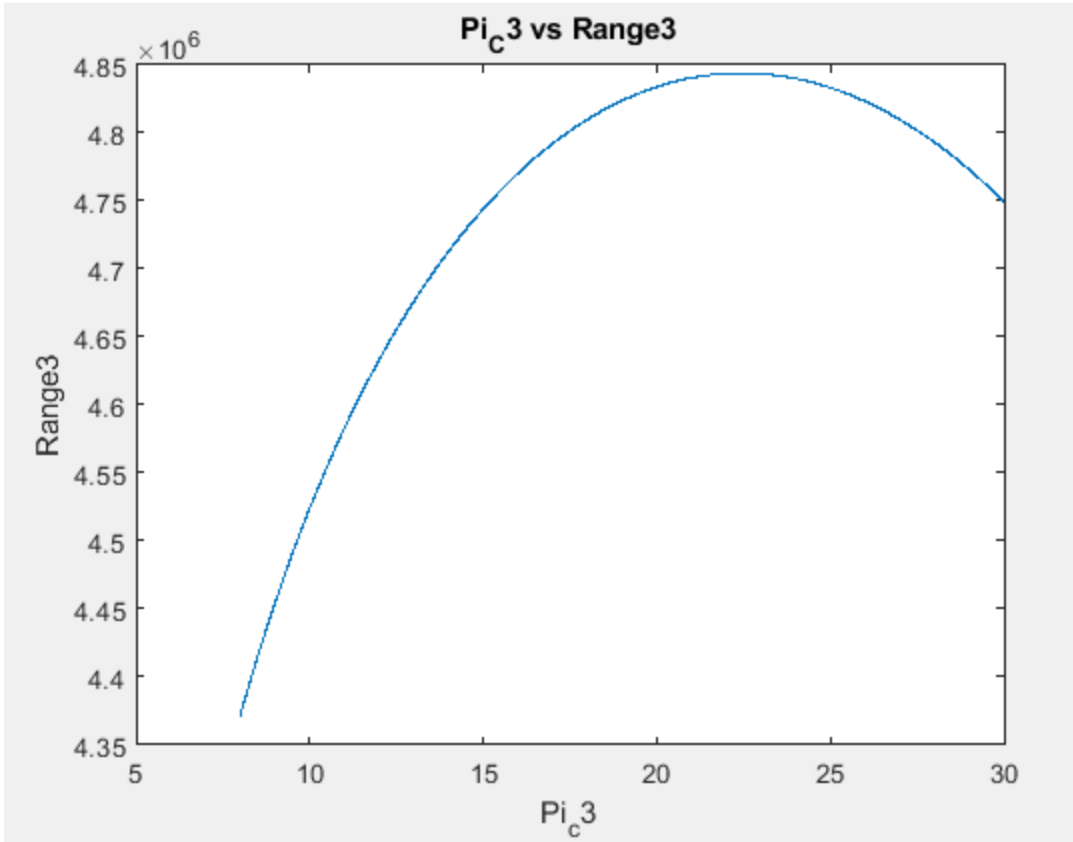
Results

The SFC variation versus the corresponding compressor's different pressure ratios



Which indicates the minimum SFC that could be achieved is **3.43e-05** $(kg/s)/N$ that corresponds to an optimum compressor pressure ratio of **22.4500**

The range variation versus the corresponding compressor different pressure ratios



Indicating a maximum range of **4844.1** km for that optimum compressor pressure ratio

And for the off-design case

```
Tt4_3 =

        1170


Tt4_off_2 =

        1070


Tt4_off_1 =

        1070
```

```
pi_cBest =

   28.8200


pi_c_off_2 =

   25.6055


pi_c_off_1 =

   25.9358
```

Code

```
clc; clear; close all;
%% Initial Values and Spacs
AR = 8; %aspect ratio
b =  280; %wing area in m2
S = b^2/AR; %wing span
gammac = 1.4;   gammat = 1.3;
cpc = 1004.5;   cpt = 1148.86;
Rc = 287;       Rt = cpt*(gammat-1)/gammat;
pi_d = 0.96;    pi_b = 0.96;    pi_n = 0.96;
eta_c = 0.88;   eta_b = 0.99;   eta_t = 0.9;    eta_m = 0.98;
hpr = 4.28e7;
W_max = 141700;
W_empty = 64600;
W_payload = 21150;
W_f = W_max - W_empty - W_payload;
W_endClimb = W_max - 0.1*W_f;   % 10% fuel comsumed
W_endCruise = W_max - 0.9*W_f;   % 90% fuel comsumed
% take-off(start mid), climb(start mid), & cruise(start mid end)
W = [W_max 140301.25 138902.5 137503.75 W_endClimb 113725 W_endCruise]*9.81; % N
H = [0 3 8.7]*10^3; % m
RC = [6.35 10]; % rate of climb
V = [241 216];

%% Evaluating M4
To3 = 288 - 6.5e-3*H(1);
To5 = 288 - 6.5e-3*(2*H(2));
M3 = V(1)/sqrt(gammac*Rc*To3);
M5 = V(2)/sqrt(gammac*Rc*To5);
M4 = 0.5*(M3+M5); % M mid climb
M = [0.286  M4 0.89];
```

```
theta = asin(RC./V); % rad
theta_avg = 0.5*(theta(1)+theta(2));
To = 288 - 6.5e-3*H;    % K
Po = (101.325*(1-(2.257e-5)*H).^5.256); % Pa
pho = Po./(0.2869*To);  % kg/m3
V = M.*sqrt(gammac*Rc*To); % update the V vector with the mid points speeds

%% Finding the min Tt4 & Pi_c
% Assumptions
Tt4_3 = 1170;

% cruise @ 6
% ram effect
tau_r_3 = 1 + (gammac - 1)/2 * M(3)^2;
pi_r_3 = tau_r_3^(gammac/(gammac-1));
Tt2_3 = tau_r_3 * To(3);
Pt2_3 = pi_r_3 * Po(3);
```

```
%% Finding the min Tt4 & Pi_c @ cruise
% ram effect
tau_r_3 = 1 + (gammac - 1)/2 * M(3)^2;
pi_r_3 = tau_r_3^(gammac/(gammac-1));
Tt2_3 = tau_r_3 * To(3);
Pt2_3 = pi_r_3 * Po(3);
% comb.
pi_c_3 = 8:0.01:30;
tau_c_3 = (pi_c_3.^((gammac-1)/gammac)-1)./eta_c + 1;
% c.c.
tau_lambda_3 = cpt*Tt4_3/(cpc*To(3));
f_3 = (tau_lambda_3-tau_r_3.*tau_c_3)./(((eta_b*hpr)/(cpc*To(3)))-tau_lambda_3);
% turb.
tau_t_3 = 1 - tau_r_3.*(tau_c_3-1)./((eta_m.*(1+f_3))*tau_lambda_3);
pi_t_3 = (1-(1-tau_t_3)./eta_t).^(gammat/(gammat-1));
% nozzle
pt9p9_cr_3 = (0.5*gammat+0.5)^(gammat/(gammat-1));
pt9p0_3 = pi_r_3.*pi_d.*pi_c_3.*pi_b.*pi_t_3.*pi_n;
```

```
for i = 1:length(pt9p0_3)
    if pt9p0_3(i) < pt9p9_cr_3 % unchoked nozzle
        disp('unchoked nozzle');
        p9p0_3(i) = pt9p0_3(i)/pt9p9_cr_3;
        M9_3(i) = 2/(gammat-1)*((p9p0_3(i))^((gammat-1)/gammat)-1);
        T9_3(i) = Tt4_3*tau_t_3(i);
        V9_3(i) = M9_3(i) * sqrt(gammat*Rt*T9_3(i));
    else % choked nozzle
        disp('choked nozzle');
        p9p0_3(i) = pt9p0_3(i)/pt9p9_cr_3;
        M9_3(i) = 1;
        T9_3(i) = Tt4_3*tau_t_3(i)/(0.5+0.5*gammat);
        V9_3(i) = sqrt(gammat*Rt*T9_3(i));
    end
end
V9e_3 = V9_3 + V9_3./(gammat.*M9_3.^2).*(1-1./p9p0_3).*(1+f_3);

% preformance
Vo_3 = M(3) .* sqrt(gammac*Rc*To(3));
Fmo = (1+f_3).*V9e_3 - Vo_3;
SFC = f_3./Fmo;
```

```
plot(pi_c_3,SFC);
xlabel('Pi_c3'); ylabel('SFC3'); title('Pi_C3 vs SFC3');

%% Thrust Required
% cruise
a3 = 0.5*pho(3)*V9e_3(best)^2*b;
CL(3) = W(6)/a3;
CD(3) = (0.01 + 0.065*CL(3)^2);
F_r(3) = 1/4*(a3*CD(3)); % thrust required for one engine

%% geting mo3 & A9
mo_3 = (F_r(3)/Fmo(best));
F_a(3) = Fmo(best)*mo_3;
R3 = 2.*sqrt(2/(pho(3)*S)).*1./SFC.*CL(3).^0.5/CD(3).*(W(5)^0.5 - W(7)^0.5);
[RMax, bestR] = max(R3);
R(3) = RMax;

figure
plot(pi_c_3,R3);
xlabel('Pi_c3'); ylabel('Range3'); title('Pi_C3 vs Range3');
```

```
%% off-design procedure
F_a(1) = 0; F_r(1) = 1; F_a(2) = 0; F_r(2) = 1;
mo_ref_req = mo_3;
while F_a(1) < F_r(1)
    mo_3 = mo_ref_req;
    Tt2_ref = Tt2_3;
    Tt4_ref = Tt4_3;
    tau_c_ref = tau_c_3(best);
    pi_c_ref = pi_cBest;
    Pt2_ref = Po(1)*pi_r_3;
    mo_ref = mo_3;

    %take off
    % ram effect
    tau_r_off_1 = 1+((gammac-1)/2)*M(1)^2;
    pi_r_off_1 = (1+((gammac-1)/2)*M(1)^2)^(1.4/0.4);
    Tt2_off_1 = tau_r_off_1*To(1);
    Tt4_off_1 = Tt4_3 - 100; % same as Tt4_3
    sqrt_Tt2_rel_1 = sqrt(Tt2_off_1)/sqrt(Tt2_ref);
    sqrt_Tt4_rel_1 = sqrt(Tt4_off_1)/sqrt(Tt4_ref);

    syms tau_c_off_1
    eqn2 = ((tau_c_off_1-1)/(tau_c_ref-1))==((Tt4_off_1/Tt2_off_1)/(Tt4_ref/Tt2_ref));
    tau_c_off_1 = double(solve(eqn2,tau_c_off_1));
    pi_c_off_1 = tau_c_off_1^(gammac/(gammac-1));
    pi_c_rel_1 = pi_c_off_1/pi_c_ref;
```

```
syms mo_rel_1
Pt2_off_1 = pi_r_off_1*Po(1);
Pt2_rel_1 = Pt2_off_1/Pt2_ref;
eqn1 = mo_rel_1*sqrt_Tt2_rel_1/Pt2_rel_1==pi_c_rel_1/(sqrt_Tt4_rel_1/sqrt_Tt2_rel_1);
mo_rel_1 = double(solve(eqn1,mo_rel_1));
mo_off_1 = mo_rel_1*mo_ref;

% available thrust
M9_off_1 = 1;
Tt9_off_1 = tau_t_3(best)*Tt4_off_1;
T9_off_1 = Tt9_off_1/(1+((gammat-1)/2));
V9_off_1 = sqrt(gammat*Rt*T9_off_1);
Pt9P9_cr_off = (0.5*gammat+0.5)^(gammat/(gammat-1));
Pt9P0_off_1 = pi_r_off_1*pi_d*pi_c_off_1*pi_b*pi_t_3(best)*pi_n;
P0P9_off_1 = Pt9P9_cr_off/Pt9P0_off_1;
F_a(1) = mo_off_1*((1+f_3(best))*V9_off_1-V(1)+((1+f_3(best))*(V9_off_1/(gammat*(M9_off_1^2)))*(1-(P0P9_off_1))));
ct1 = f_3(best)/(F_a(1)/mo_off_1);
```

```
    % required thrust
    a1 = 0.5*pho(1)*V(1)^2*b;
    CL(1) = W(2)/a1;
    CD(1) = (0.01 + 0.065*CL(1)^2);
    F_r(1) = 1/4*(a1*CD(1)); % thrust required for one engine
    R(1) = 2*sqrt(2/(pho(1)*S))*1/ct1*CL(1)^0.5/CD(1)*(W(1)^0.5 - W(3)^0.5);

    if F_a(1) < F_r(1)
        mo_off_r_1 = F_r(1)/F_a(1)*mo_off_1;
        mo_ref_req = mo_off_r_1/mo_rel_1;
    end
end
```

```
% climb
while F_a(2) < F_r(2)

    % ram effect
    tau_r_off_2 = 1+((gammac-1)/2)*M(2)^2;
    pi_r_off_2 = (1+((gammac-1)/2)*M(2)^2)^(1.4/0.4);
    Tt2_off_2 = tau_r_off_2*To(2);
    Tt4_off_2 = Tt4_3 - 100;
    sqrt_Tt2_rel_2 = sqrt(Tt2_off_2)/sqrt(Tt2_ref);
    sqrt_Tt4_rel_2 = sqrt(Tt4_off_2)/sqrt(Tt4_ref);

    syms tau_c_off_2
    eqn2 = ((tau_c_off_2-1)/(tau_c_ref-1))==(Tt4_off_2/Tt2_off_2)/(Tt4_ref/Tt2_ref);
    tau_c_off_2 = double(solve(eqn2,tau_c_off_2));
    pi_c_off_2 = tau_c_off_2^(gammac/(gammac-1));
    pi_c_rel_2 = pi_c_off_2/pi_c_ref;

    syms mo_rel_2
    Pt2_off_2 = pi_r_off_2*Po(2);
    Pt2_rel_2 = Pt2_off_2/Pt2_ref;
    eqn1 = mo_rel_2*sqrt_Tt2_rel_2/Pt2_rel_2==pi_c_rel_2/(sqrt_Tt4_rel_2/sqrt_Tt2_rel_2);
    mo_rel_2 = double(solve(eqn1,mo_rel_2));
    mo_off_2 = mo_rel_2*mo_ref;
```

```
    % available thrust
    M9_off_2 = 1;
    Tt9_off_2 = tau_t_3(best)*Tt4_off_2;
    T9_off_2 = Tt9_off_2/(1+((gammat-1)/2));
    V9_off_2 = sqrt(gammat*Rt*T9_off_2);
    Pt9P0_off_2 = pi_r_off_2*pi_d*pi_c_off_2*pi_b*pi_t_3(best)*pi_n;
    P0P9_off_2 = (1/Pt9P0_off_2)*Pt9P9_cr_off;
    F_a(2) = mo_off_2*((1+f_3(best))*V9_off_2-V(2)+((1+f_3(best))*(V9_off_2/(gammat*(M9_off_2^2)))*(1-(P0P9_off_2))));
    ct2 = f_3(best)/(F_a(2)/mo_off_2);

    % required thrust
    a2 = 0.5*pho(2)*V(2)^2*b;
    CL(2) = W(4)*cos(theta_avg)/a2;
    CD(2) = (0.01 + 0.065*CL(2)^2);
    F_r(2) = 1/4*(a2*CD(2)+W(4)*sin(theta_avg));
    R(2) = 2*sqrt(2/(pho(2)*S))*1/ct2*CL(2)^0.5/CD(2)*(W(3)^0.5 - W(5)^0.5);

    if F_a(2) < (W(4)*sin(theta_avg) + F_r(2))
        mo_off_r_2 = F_r(2)/F_a(2)*mo_off_2;
        mo_ref_req = mo_off_r_2/mo_rel_2;
    end

end
```

References

1. *Earth Atmosphere Model - Metric Units. (n.d.).* The National Aeronautics and Space Administration. Retrieved June 23, 2022, from https://www.grc.nasa.gov/www/k-12/airplane/atmosmet.html