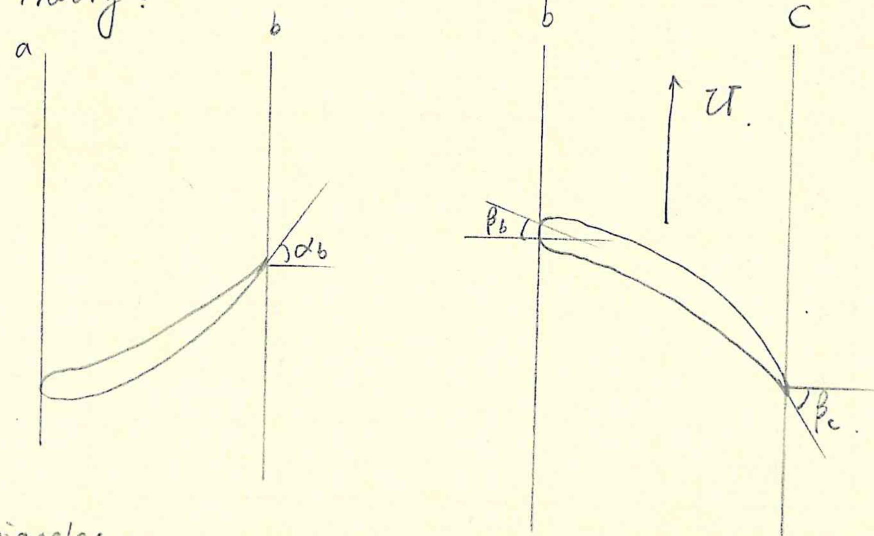
Part 1. Turbine Stator - Rotor Analysis

(a) Background Theory:



Velocity triangles:

(b) $u_{\theta,b} = u_x \tan\alpha_b$

$u_{\theta,c} = U - u_x \tan\beta_c$.

So $\Delta u_\theta = u_{\theta,b} - u_{\theta,c} = u_x(\tan\alpha_b + \tan\beta_c) - U$

Work: $\Delta h_0 = h_{0a} - h_{0c} = \underbrace{h_{0a} - h_{0b}}_{0} + \underbrace{h_{0b} - h_{0c}}_{U\Delta u_\theta} = U\Delta u_\theta$

$$= U\left[u_x(\tan\alpha_b + \tan\beta_c) - U\right]$$

$$\Rightarrow \boxed{\Delta h_0 = U^2\left[\frac{u_x}{U}(\tan\alpha_b + \tan\beta_c) - 1\right]}$$

Going further from last equation:

$$h_{0a} - h_{0c} = c_p(T_{0a} - T_{0c}) = U\left[u_x(\tan\alpha_b + \tan\beta_c) - U\right]$$

$$1 - \frac{T_{0c}}{T_{0a}} = \frac{U}{c_p T_{0a}}\left[u_x(\tan\alpha_b + \tan\beta_c) - U\right]$$

$$\Rightarrow 1-\frac{T_{0c}}{T_{0a}} = \frac{U}{\underbrace{\frac{\gamma}{\gamma-1}R}_{c_p} T_{0a}}\left[u_x(\tan\alpha_b+\tan\beta_c)-U\right]$$

$$=(\gamma-1)\frac{U}{\gamma R T_{0a}}\left[u_x(\tan\alpha_b+\tan\beta_c)-U\right]$$

Therefore, $$\boxed{1-\frac{T_{0c}}{T_{0a}}=(\gamma-1)\frac{U}{\sqrt{\gamma R T_{0a}}}\left[\frac{u_x}{\sqrt{\gamma R T_{0a}}}(\tan\alpha_b+\tan\beta_c)-\frac{U}{\sqrt{\gamma R T_{0a}}}\right]}.$$

(c) Stage efficiency:

$$\eta_{st}=\frac{h_{0a}-h_{0c}}{h_{0a}-h_{0cs}}=\frac{\frac{T_{0c}}{T_{0a}}-1}{\frac{T_{0cs}}{T_{0a}}-1} \Rightarrow \frac{T_{0cs}}{T_{0a}}=1+\frac{1}{\eta_{st}}\left(\frac{T_{0c}}{T_{0a}}-1\right)$$

pressure ratio:

$$\frac{P_{0c}}{P_{0a}}=\frac{P_{0cs}}{P_{0a}}=\left(\frac{T_{0cs}}{T_{0a}}\right)^{\frac{\gamma}{\gamma-1}}=\left[1+\frac{1}{\eta_{st}}\left(\frac{T_{0c}}{T_{0a}}-1\right)\right]^{\frac{\gamma}{\gamma-1}}$$

Extending to multi-stage turbine:

①: Stage 1: $$\frac{T_{02s}}{T_{01}}=1-\frac{1}{\eta_{st}}\frac{U\Delta u_\theta}{c_pT_{01}} \quad (1a), \qquad T_{02}-T_{01}=-\frac{U\Delta u_\theta}{c_p} \quad (1b)$$

②: Stage 2: $$\frac{T_{03s}}{T_{02}}=1-\frac{1}{\eta_{st}}\frac{U\Delta u_\theta}{c_pT_{02}}$$

$$=1-\frac{1}{\eta_{st}}\frac{U\Delta u_\theta}{c_pT_{01}\left(1-\frac{U\Delta u_\theta}{c_pT_{01}}\right)} \quad (2a), \qquad T_{03}-T_{02}=-\frac{U\Delta u_\theta}{c_p} \quad (2b)$$

③: Stage 3: $$\frac{T_{04s}}{T_{03}}=1-\frac{1}{\eta_{st}}\frac{U\Delta u_\theta}{c_pT_{03}}$$

$$=1-\frac{1}{\eta_{st}}\frac{U\Delta u_\theta}{c_pT_{01}\left(1-2\frac{U\Delta u_\theta}{c_pT_{01}}\right)} \quad (3a), \qquad T_{04}-T_{03}=-\frac{U\Delta u_\theta}{c_p} \quad (3b)$$

$\vdots$

Con't

⋮

(n): Stage n :

$$\frac{T_{0,n+1,s}}{T_{0n}} = 1 - \frac{1}{\eta_{st}} \cdot \frac{U\Delta U_\theta}{c_p T_{01}\left[1-(n-1)\frac{U\Delta U_\theta}{c_p T_{01}}\right]} \quad (na), \quad T_{0n+1} - T_{0n} = -\frac{U\Delta U_\theta}{c_p} \quad (nb).$$

Adding (1b) + (2b) + (3b) + ⋯ + (nb) :

$$T_{0n+1} - T_{01} = -n \cdot \frac{U\Delta U_\theta}{c_p} \Rightarrow \frac{T_{0n+1}}{T_{01}} - 1 = -n \cdot \frac{U\Delta U_\theta}{c_p T_{01}} \quad (*)$$

Now pressure ratio:

$$\left(\frac{P_{0n+1}}{P_{01}}\right)^{\frac{\gamma-1}{\gamma}} = \frac{T_{0n+1,s}}{T_{01}} = (1a) \times (2a) \times (3a) \times \cdots \times (na)$$

$$\Rightarrow \boxed{\frac{T_{0n+1,s}}{T_{01}} = \left(1 - \frac{1}{\eta_{st}} \frac{U\Delta U_\theta}{c_p T_{01}}\right)\left[1 - \frac{1}{\eta_{st}} \frac{U\Delta U_\theta}{c_p T_{01}\left(1 - \frac{U\Delta U_\theta}{c_p T_{01}}\right)}\right] \cdots \left\{1 - \frac{1}{\eta_{st}} \frac{U\Delta U_\theta}{c_p T_{01}\left[1-(n-1)\frac{U\Delta U_\theta}{c_p T_{01}}\right]}\right\}} \quad (**)$$

Turbine efficiency :

$$\eta_T = \frac{h_{01} - h_{0n+1}}{h_{01} - h_{0n+1s}} = \frac{1 - \frac{T_{0n+1}}{T_{01}} \;(*)}{1 - \frac{T_{0n+1s}}{T_{01}} \;(**)} =$$

$$\Rightarrow \boxed{\eta_T = \frac{n \cdot \frac{U\Delta U_\theta}{c_p T_{01}}}{1 - \left(1 - \frac{1}{\eta_{st}} \frac{U\Delta U_\theta}{c_p T_{01}}\right)\left[1 - \frac{1}{\eta_{st}} \frac{U\Delta U_\theta}{c_p T_{01}\left(1 + \frac{U\Delta U_\theta}{c_p T_{01}}\right)}\right] \cdots \left\{1 - \frac{1}{\eta_{st}} \frac{U\Delta U_\theta}{c_p T_{01}\left[1+(n-1)\frac{U\Delta U_\theta}{c_p T_{01}}\right]}\right\}}}$$

(d) The MATLAB code is in Appendix 1 on Page 13.

Outputs from MATLAB: $\boxed{\text{P\_ratio} = 0.1120;\ \text{eta\_t} = 0.9179,\ \text{T\_out} = 917.0994\,K,\ \text{Work\_out} = 6.8632 \times 10^5\ J/kg}$

which are correct.

Part 2. Turbine Design

(e) The mechanical constraint is that the angular velocity ω of compressor and turbine should be the same.

From compressor: $\omega = \dfrac{U_c}{\bar{r}_c} = \dfrac{310\,m/s}{0.115\,m} = 2695.65\ s^{-1}$

Then for the turbine:

$$\boxed{U_t = \omega \bar{r}_t = 2695.65\,s^{-1} \times 0.08\,m = 215.65\ m/s}$$

(f) Ignoring the fuel-air ratio when using mass balance between HPC and HPT, we get:

$$\dot{m}_t \cong \dot{m}_{core} = 2.825\ kg/s.$$

When choked at turbine inlet:

$$f(M_4) = f(M_4=1) = \frac{A_4}{A_4^*} = \frac{1}{1}\underbrace{\left(\frac{\gamma+1}{2}\right)^{-\left(\frac{\gamma+1}{2(\gamma-1)}\right)}\left(\frac{\gamma+1}{2}\right)^{\left(\frac{\gamma+1}{2(\gamma-1)}\right)}}_{1}$$

$$= 1.$$

$$P_{04} = P_{03} = \pi_c P_{02.5} = 12 P_{02.5} = 988.8\ kPa$$

Therefore, using the following equation:

$$\dot{m}_t = \gamma\left(\frac{\gamma+1}{2}\right)^{\frac{-(\gamma+1)}{2(\gamma-1)}} \frac{P_{04} A_4 f(M_4=1)}{\sqrt{\gamma R T_{04}}}.$$

we can get A_4:

$$A_4 = \dot{m}_t \cdot \frac{\sqrt{\gamma R T_{04}}}{P_{04} f(M_4)} \cdot \frac{1}{\gamma}\left(\frac{\gamma+1}{2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

$$A_4 = 2.875\,kg/s \times \frac{\sqrt{1.4\times 287 J/kg\cdot K \times 1600K}}{988.8\times10^3\,Pa\times 1} \times \frac{1}{1.4} \times \left(\frac{1.4+1}{2}\right)^{\frac{1.4+1}{2(1.4-1)}}$$

$$\Rightarrow \boxed{A_4 = 2.827\times10^{-3}\,m^2}$$

Continue:

$$A_4 = \pi(r_{tip}^2 - r_{hub}^2) = 2.827\times10^{-3}\,m^2 \quad (1)$$

$$\bar{r} = \frac{1}{2}(r_{tip} + r_{hub}) = 0.08\,m \quad (2)$$

Combine (1) and (2), we can get:

$$\begin{cases} r_{tip} = 0.082812\,m \\ r_{hub} = 0.077188\,m \end{cases} \Rightarrow \boxed{\frac{r_{hub}}{r_{tip}} = 0.932}$$

Other than α_6 and β_5, we also need to know the axial velocity U_x in the turbine, the stage efficiency in the turbine, and the number of stages in the turbine.

(g) We know that:

$$h_0 = h + \frac{1}{2}u^2$$

$$\Rightarrow T_0 = T + \frac{u^2}{2c_p}$$, with $c_p = \frac{\gamma}{\gamma-1}R$, $\gamma RT = a^2$ and $M = \frac{u}{a}$.

we have:

$$\frac{T_0}{T} = 1 + \left(\frac{\gamma-1}{2}\right)M^2 \quad \text{or} \quad \frac{T}{T_0} = \left[1+\left(\frac{\gamma-1}{2}\right)M^2\right]^{-1}$$

Since $u = Ma = M\sqrt{\gamma RT}$

then $$\boxed{u = M\sqrt{\gamma R T_0\left(1+\frac{\gamma-1}{2}M^2\right)^{-1}}} \quad (*)$$

Reversely, we know:

$$u = M\sqrt{\gamma R T_0 \left(1+\frac{\gamma-1}{2}M^2\right)^{-1}}$$

$$\Rightarrow \frac{u^2}{\gamma R T_0} = \frac{M^2}{1+\frac{\gamma-1}{2}M^2} \Rightarrow \frac{u^2}{\gamma R T_0} + \frac{u^2}{\gamma R T_0}\frac{\gamma-1}{2}M^2 = M^2$$

$$\Rightarrow M^2 = \frac{u^2/\gamma R T_0}{1-\frac{\gamma-1}{2}\cdot\frac{u^2}{\gamma R T_0}} = \frac{1}{\frac{\gamma R T_0}{u^2}-\frac{\gamma-1}{2}}$$

$$\Rightarrow \boxed{M = \sqrt{\frac{1}{\frac{\gamma R T_0}{u^2}-\frac{\gamma-1}{2}}}} \quad (**)$$

Specifically when turbine inlet is choked, M=1. Using (*), we have:

$$u_x = 1\sqrt{\gamma R T_{04}\left(1+\frac{\gamma-1}{2}\right)^{-1}}$$

$$= \sqrt{1.4\times 287\,J/kg\cdot K \times 1600\,K \times \left(1+\frac{1.4-1}{2}\right)^{-1}}$$

$$= 731.94\ m/s$$

(h) From HW4, the design/target work from HPc is:

$$W = n U \Delta u_\theta$$

$$= nU\left(U - u_x(\tan\alpha_a + \tan\beta_b)\right)$$

$$= 4\times 307\,m/s\left[307\,m/s - 75\,m/s\times(\tan 30^\circ + \tan 30^\circ)\right]$$

$$= \boxed{2.71\times 10^5\ J/kg}$$

The contour plot is shown in Figure 1 on Page 7. The design/target work contour is also shown in Figure 1. The MATLAB code is in Appendix 2 on Pages 13-14.

Part 2. Turbine Design

(h). Specific work contour plot

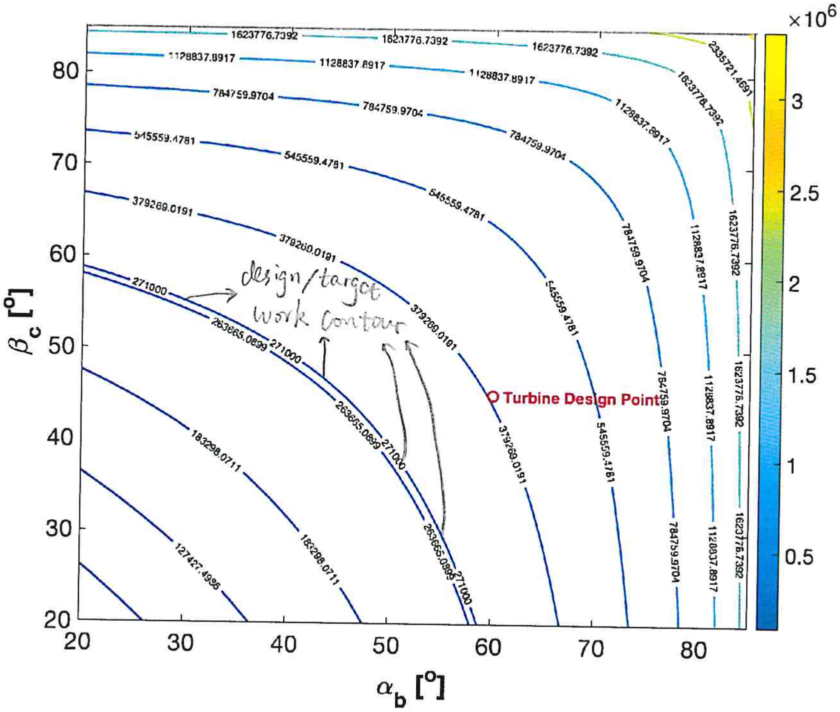


Figure 1.

(i).

From the desired work contour, we see that the range of angles is quite reasonable. Theoretically, we can achieve higher turning angles in turbines than the angles in compressors. Because in compressors, the working fluid is moving in adverse pressure gradient. Flow separations would occur if we increase the turning angle in compressor, to a certain extent. In contrast, the working fluid is moving in favorable pressure gradient, when flow separation is less possible to happen. Therefore, one stage is enough for the HPT design.

(j).

The design point of $\alpha_b = 60^o$ and $\beta_c = 45^o$ is shown also in Figure 1, as a red circle. The turbine work is higher than the target work from HPC ($2.17\times10^5 J/kg$). Therefore, the angles proposed here is a satisfactory design.

Part 3. Turbine Map

(a). Pseudo code

```
function [pt_ratio, fM4_rev, eta_t, T_out, Work_out] = turbineMap(cp, gamma, T, eta_st, u_x, U, n, alpha_b, beta_c)
1)  Get [pt_ratio, eta_t, T_out, Work_out] from turbfn function we wrote in Part 1(d);
2)  Get M04 using the equation derived in Part 2(g), M04 = sqrt(1 / (gamma * R * T/u_x^2 - (gamma - 1)/2));
3)  Get f(M04), fM04 = (1/M4) * (((gamma + 1)/2)^(-(gamma + 1)/2/(gamma - 1))) * ((1 + (gamma - 1)/2 *
    M4^2)^((gamma + 1)/2/(gamma - 1)));
4)  Calculate fM4_rev = 1 / fM4;
end
```

(b)

The initial turbine map is shown below in Figure 2.

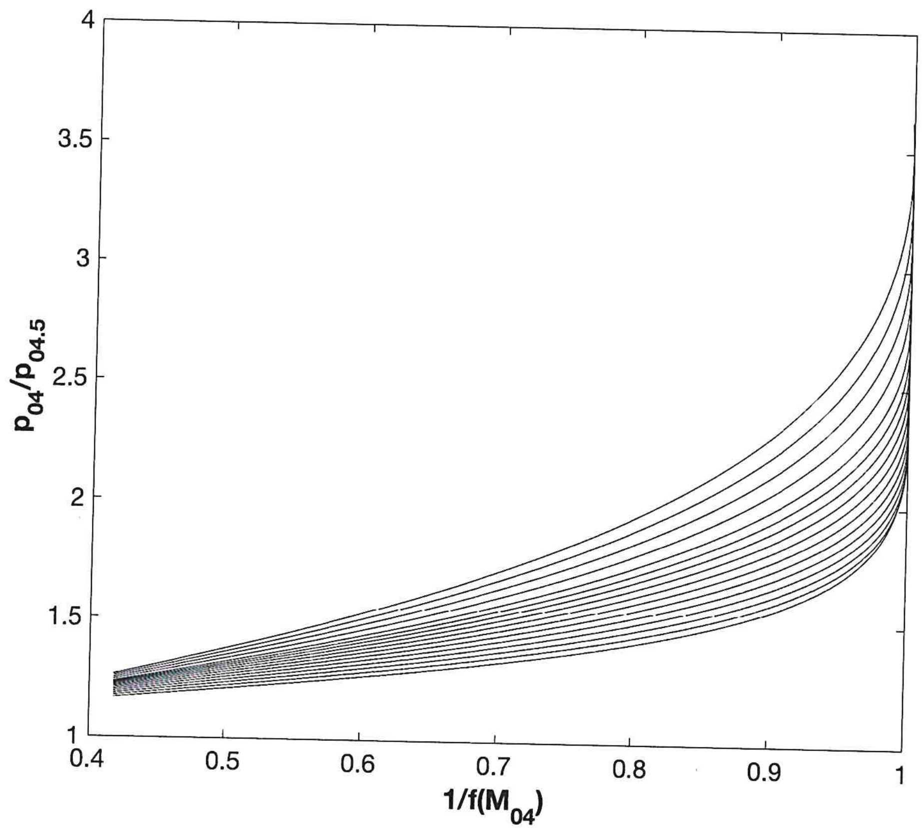


Figure 2.

$f(M_{04}) = \frac{A}{A^*}$, consistent with HW4 formula summary, but different from the $f(M) = \frac{A^*}{A}$ defined in the Project document from Prof. Ihme.

8

(c). Consolidated turbine map

The consolidated turbine map is shown in Figure 3. The MATLAB code of (b) and (c) is in Appendix3 on Pages 14-15 .

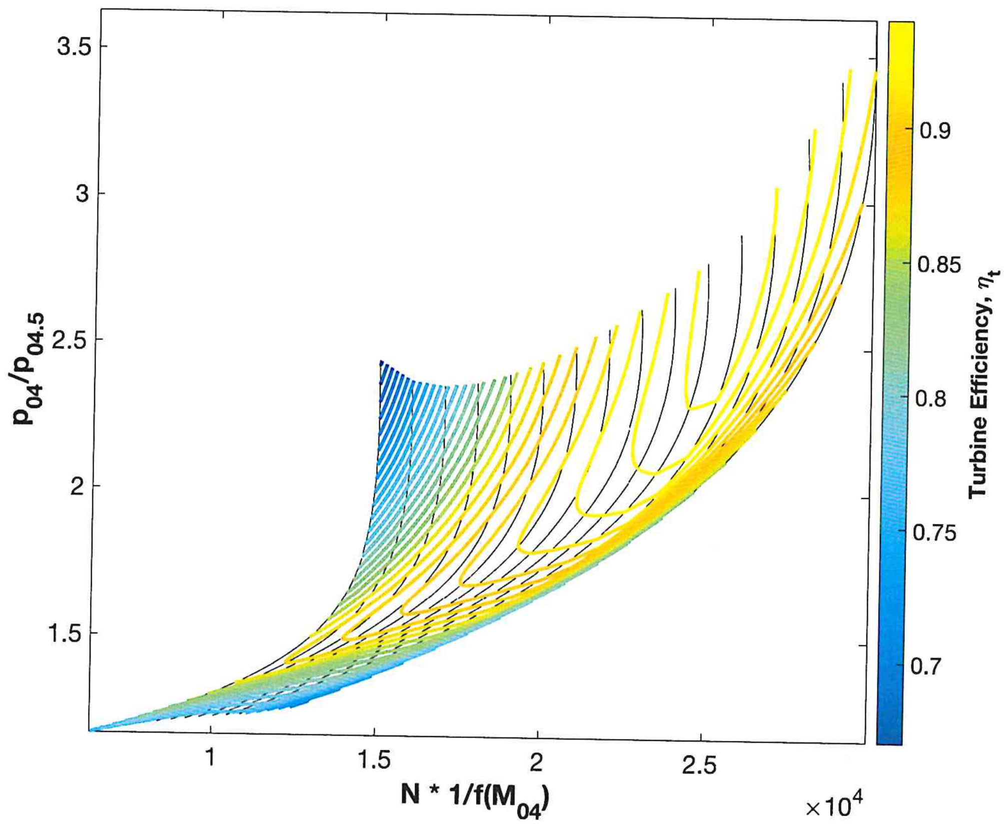


Figure 3.

$$f(M_{04}) = \frac{A}{A^*}.$$

Part 4. Compressor-Turbine Matching and Operating Line

(a). Approach.

The computation approach is shown as following steps:

1) Compute compressor map;
2) Compute turbine map;
3) Select/guess a compressor/turbine rotating speed N, then get the U values;
4) Guess an axial velocity in compressor, i.e. $u_{x,c}$;
5) Get pressure ratio, $f(M_{02.5})$, and compressor work from the compressor map function which has been written in HW4;
6) Calculate p_{03} and p_{04} using the compressor pressure ratio;
7) Calculate mass flow rate in compressor, $\dot{m}_{compressor}$;
8) Equate turbine work to compressor work, and calculate the corresponding axial velocity in turbine, i.e. $u_{x,t}$;
9) Using the chocking condition ($M_4 = 1$) to calculate turbine inlet temperature T_{04};
10) Get pressure ratio, $f(M_{04})$, and from the turbine map function which has been written in Part 3;
11) Calculate mass flow rate in turbine, $\dot{m}_{turbine}$;
12) Calculate fuel to mass ratio f;
13) Check whether $\dot{m}_{compressor}(1+f)$ is equal to $\dot{m}_{turbine}$. if yes, march to the next rotating speed N; if not, guess another axial velocity in compressor $u_{x,c}$.

The variables for iteration is the axial velocity in compressor $u_{x,c}$. The objective function is the difference between $\dot{m}_{compressor}(1+f)$ and $\dot{m}_{turbine}$.

(b). Pseudo Code

```
Compute compressor map;
Compute turbine map;
for N = 15000:1000:30000
        U_comp = N / 60 * 2 * pi * r_mean_compressor;
        U_turb = N / 60 * 2 * pi * r_mean_turbine;
        for ux_c varies
                [pc_ratio, fM025_rev, eta_c, T03_guess, Work_in] = compressorMap(inputs);
                Calculate p03 and p04 using the compressor pressure ratio;
                Calculate mass flow rate in compressor, ṁ_compressor;
                wc = Work_in;
                wt = wc;
                Get ux_t from wt;
                Using the chocking condition (M4 = 1) to calculate turbine inlet temperature T04;
                [pt_ratio, fM04_rev, eta_t, T05_guess, Work_out] = turbineMap(inputs);
                Calculate mass flow rate in turbine, ṁ_turbine;
                Calculate fuel to mass ratio f;
                func = ṁ_turbine − ṁ_compressor(1 + f);
                if (abs(func) < 0.01)
                    break;
                end
        end
end
```

(c). Write Code.

The MATLAB code is shown in Appendix 4 on Pages 15-17.

(d). Operating Lines.

The compressor and turbine maps showing the operating lines are in Figures 4 and 5 on the next page.

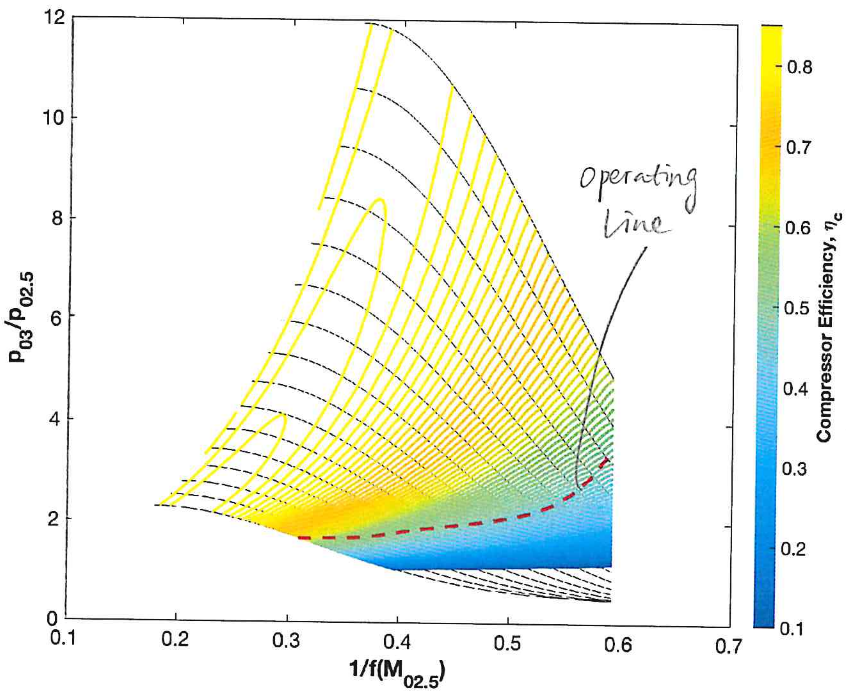


Figure 4.

$f(M) = \frac{A}{A^*}$.

Different from the definition of $f(M) = \frac{A^*}{A}$ in this Project document from Prof. Ihme. But the document asks us to plot $f(M) = \frac{A^*}{A}$. I am plotting my $\frac{1}{f(M)} = \frac{A^*}{A}$, which is consistent!

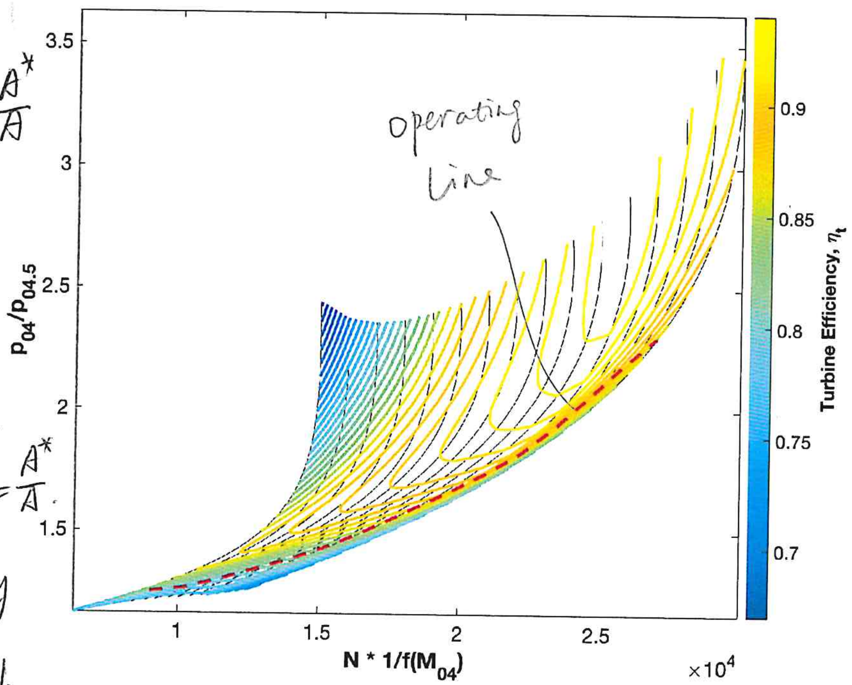


Figure 5.

Appendix 1. MATLAB code for Part 1(d).

```
function [p_ratio, eta_t, T_out, Work_out] = ...
    turbfn(cp, gamma, T, eta_st, u_x, U, n, alpha_b, beta_c)

dU_theta = u_x * (tan(abs(alpha_b)) + tan(abs(beta_c))) - U;
UdU_theta = U * dU_theta;

T_out = T - n * UdU_theta/cp;
Work_out = n * UdU_theta;

p_ratio = 1;
for i = 1:1:n
    p_ratio = p_ratio * ((1 - 1/eta_st * UdU_theta/...
        (cp * T - (i - 1) * UdU_theta))^(gamma/(gamma-1)));
end

eta_t = (n * UdU_theta/cp/T) / (1 - p_ratio^((gamma - 1)/gamma));

end
```

Appendix 2. MATLAB code for Part 2(h).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  ME357 Spring 2017
%  Final Project
%  2. Turbine Design (h)
%
%  Rui Xu (ruixu@stanford.edu)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc

% Given properties
cp = 1005;
gamma = 1.4;
T = 1600;
eta_st = 0.95;
u_x = 731.94;
U = 215.65;
n = 1;

% Turbine geometries
alpha_bd = 20:1:85;
beta_cd = 20:1:85;
alpha_b = alpha_bd / 180 * pi;
beta_c  = beta_cd / 180 * pi;
lb = length(alpha_b);
lc = length(beta_c);

% Calculate contours
work = zeros(lc,lb);
for i = 1:lc
    for j = 1:lb
        [p_ratio, eta_t, T_out, Work_out] =...
            turbfn(cp, gamma, T, eta_st, u_x, U, n, alpha_b(j), beta_c(i));
        work(i,j) = Work_out;
    end
end
work_values = logspace(4,7,20);
work_values = [2.71e5 work_values];

% Plot contours
figure(1)
co = [1 0 0;0 1 0;0 0 1];
contour(alpha_bd, beta_cd, work, work_values,'ShowText','on','LineWidth',2.0)
hold on
colorbar
xlabel('\alpha_b [^o]','FontSize',25,'FontWeight','bold');
ylabel('\beta_c [^o]','FontSize',25,'FontWeight','bold');
axesh = findobj('Type', 'axes');
```

```
set(axesh, 'Box','on');
set(gca,'FontSize',23);
plot(60, 45, 'ro','MarkerSize',10.0, 'LineWidth',2.0)
text(61,45,'Turbine Design Point','Color','r','FontSize',15.0,'FontWeight','Bold')
```

Appendix 3. MATLAB code for Part 3.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  ME357 Spring 2017
%  Final Project
%  3. Turbine Map
%
%  Rui Xu (ruixu@stanford.edu)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc

% Given properties
cp = 1005;
gamma = 1.4;
T = 1600;
ux_ref = 732;
U_ref = 216;
n = 1;
alpha_b = 60 / 180 * pi;
beta_c  = 45 / 180 * pi;
eta_st = @(ux,uu) (0.95 - 0.03 * (ux/uu - ux_ref/U_ref).^2 -...
    0.2 * abs(uu - U_ref)/U_ref);
r_mean = 0.08;

% Preparation for the turbine map
N = 15000:1000:30000;
ux = 200:10:750;
lN = length(N);
lux = length(ux);
map_mDot = zeros(lux,1);
map_pRatio = zeros(lux,1);
map_etat = zeros(lux,1);

% Calculate the turbine map
for j = 1:lN
    U = N(j)/60 * 2 * pi * r_mean;
    for i = 1:lux
        [p_ratio, fM4_rev, eta_t, T_out, Work_out] =...
            turbineMap(cp, gamma, T, eta_st(ux(i),U), ux(i), U, n, alpha_b, beta_c);
        map_mDot(i) = fM4_rev;
        map_pRatio(i) = 1 / p_ratio;
        map_etat(i) = eta_t;

    end

    figure(1)
    plot(map_mDot,map_pRatio, 'k');
    hold on;

    figure(2)
    plot(map_mDot * N(j),map_pRatio, 'k');
    hold on;
    mDot_contour(j,:) = map_mDot * N(j);
    pRatio_contour(j,:) = map_pRatio;
    etat_contour(j,:) = map_etat;
end

% Plot turbine map
figure(1)
xlabel('1/f(M_{04})','FontSize',22,'FontWeight','bold');
ylabel('p_{04}/p_{04.5}','FontSize',22,'FontWeight','bold');
axesh = findobj('Type', 'axes');
set(axesh, 'Box','on');
set(gca,'FontSize',20);

figure(2)
eta_values = 0.5:0.01:0.98;
```

```
contour(mDot_contour, pRatio_contour, etat_contour, eta_values, 'LineWidth',3.0);
cbar = colorbar;
xlabel('N * 1/f(M_{04})','FontSize',22,'FontWeight','bold');
ylabel('p_{04}/p_{04.5}','FontSize',22,'FontWeight','bold');
ylabel(cbar,'Turbine Efficiency, \eta_t',...
    'FontSize',22,'FontWeight','bold');
axesh = findobj('Type', 'axes');
set(axesh, 'Box','on');
set(gca,'FontSize',20);


function [pt_ratio, fM4_rev, eta_t, T_out, Work_out] =...
    turbineMap(cp, gamma, T, eta_st, u_x, U, n, alpha_b, beta_c)

R = 287;
[pt_ratio, eta_t, T_out, Work_out] =...
    turbfn(cp, gamma, T, eta_st, u_x, U, n, alpha_b, beta_c);

M4 = sqrt(1 / (gamma * R * T/u_x^2 - (gamma - 1)/2));
fM4 = (1/M4) * (((gamma + 1)/2)^(-(gamma + 1)/2/(gamma - 1))) * ...
    ((1 + (gamma - 1)/2 * M4^2)^((gamma + 1)/2/(gamma - 1)));
fM4_rev = 1 / fM4;

end
```

Appendix 4. MATLAB code for Part 4.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  ME357 Spring 2017
%  Final Project
%  4. Compressor-Turbine Matching and Operating Line
%
%  Rui Xu (ruixu@stanford.edu)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc

% Fluid properties
cp = 1005;
gamma = 1.4;
R = 287;
LHV = 43e6;

%% Compressor Map
% Given for compressor
n_c = 4;
T025 = 311;
p025 = 82.4e3;
ux_c_ref = 75;
U_c_ref = 310;
mDot_c = 2.825;
r_mean_c = 0.115;
eta_c_st = @(x) (0.87 - 16 * (x - ux_c_ref/U_c_ref).^2);
alpha_a_comp = 30 / 180 * pi;
beta_b_comp  = 30 / 180 * pi;

% Preparation for compressor map
N = 15000:1000:30000;
ux_c = 20:1:130;
lN = length(N);
lux_c = length(ux_c);
Cmap_mDot = zeros(lux_c,1);
Cmap_pRatio = zeros(lux_c,1);
Cmap_etac = zeros(lux_c,1);

% Calculate and plot the compressor map
figure(1)
hold on
for j = 1:lN
    U_c = N(j)/60 * 2 * pi * r_mean_c;
    for i = 1:lux_c
        [pc_ratio, fM2_rev, eta_c, T_out, Work_in] =...
            compressorMap(cp, gamma, T025, eta_c_st(ux_c(i)/U_c),...
```

```
            ux_c(i), U_c, n_c, alpha_a_comp, beta_b_comp);
        Cmap_mDot(i) = fM2_rev;
        Cmap_pRatio(i) = pc_ratio;
        Cmap_etac(i) = eta_c;
    end

    % Cut off at surge line
    Cmap_etac = Cmap_etac((find(Cmap_pRatio == max(Cmap_pRatio))):end);
    Cmap_mDot = Cmap_mDot((find(Cmap_pRatio == max(Cmap_pRatio))):end);
    Cmap_pRatio = Cmap_pRatio((find(Cmap_pRatio == max(Cmap_pRatio))):end);
    plot(Cmap_mDot,Cmap_pRatio, 'k');

    % Prepare for compressor efficiency contour plot
    mDot_c_contour(j,:) = linspace(Cmap_mDot(1), Cmap_mDot(end), 100);
    pRatio_c_contour(j,:) = interp1( Cmap_mDot,Cmap_pRatio, mDot_c_contour(j,:) );
    etac_contour(j,:) = interp1( Cmap_mDot,Cmap_etac, mDot_c_contour(j,:) );
end

% Plot efficiency contour
figure(1)
eta_values = 0.1:0.01:0.9;
contour(mDot_c_contour, pRatio_c_contour, etac_contour, eta_values, 'LineWidth',3.0)
cbar = colorbar;
scale = axis;
axis([0.1 0.7 0 12]);
xlabel('1/f(M_{02.5})','FontSize',22,'FontWeight','bold');
ylabel('p_{03}/p_{02.5}','FontSize',22,'FontWeight','bold');
ylabel(cbar,'Compressor Efficiency, \eta_c',...
    'FontSize',22,'FontWeight','bold');
axesh = findobj('Type', 'axes');
set(axesh, 'Box','on');
set(gca,'FontSize',20);

%% Turbine Map
% Given for turbine
T04_t = 1600;
ux_t_ref = 732;
U_t_ref = 216;
n_t = 1;
alpha_b_turb = 60 / 180 * pi;
beta_c_turb  = 45 / 180 * pi;
eta_t_st = @(ux,uu) (0.95 - 0.03 * (ux/uu - ux_t_ref/U_t_ref).^2 -...
    0.2 * abs(uu - U_t_ref)/U_t_ref);
r_mean_t = 0.08;

% Preparation for turbine map
ux_t = 200:10:750;
lux_t = length(ux_t);
Tmap_mDot = zeros(lux_t,1);
Tmap_pRatio = zeros(lux_t,1);
Tmap_etat = zeros(lux_t,1);

% Calculate and plot turbine map
for j = 1:lN
    U_t = N(j)/60 * 2 * pi * r_mean_t;
    for i = 1:lux_t
        [pt_ratio, fM4_rev, eta_t, T_out, Work_out] =...
            turbineMap(cp, gamma, T04_t, eta_t_st(ux_t(i),U_t), ...
            ux_t(i), U_t, n_t, alpha_b_turb, beta_c_turb);
        Tmap_mDot(i) = fM4_rev;
        Tmap_pRatio(i) = 1 / pt_ratio;
        Tmap_etat(i) = eta_t;
    end

    figure(2)
    hold on;
    plot(Tmap_mDot * N(j),Tmap_pRatio, 'k');
    mDot_t_contour(j,:) = Tmap_mDot * N(j);
    pRatio_t_contour(j,:) = Tmap_pRatio;
    etat_contour(j,:) = Tmap_etat;
end

% Plot efficiency contour
figure(2)
eta_values = 0.5:0.01:0.98;
contour(mDot_t_contour, pRatio_t_contour, etat_contour, eta_values, 'LineWidth',3.0);
cbar = colorbar;
xlabel('N * 1/f(M_{04})','FontSize',22,'FontWeight','bold');
```

```
ylabel('p_{04}/p_{04.5}','FontSize',22,'FontWeight','bold');
ylabel(cbar,'Turbine Efficiency, \eta_t',...
    'FontSize',22,'FontWeight','bold');
axesh = findobj('Type', 'axes');
set(axesh, 'Box','on');
set(gca,'FontSize',20);

%% Operating line
% Areas directly calculated from compressor and turbine geometries.
A025 = 0.04154756284;
A04 = 0.002826930733;

% Preparation for operating line calculation
Op_c_mDot = zeros(1, lN);
Op_c_pRatio = zeros(1, lN);
Op_t_mDot = zeros(1, lN);
Op_t_pRatio = zeros(1, lN);
for j = 1:lN
    U_c = N(j)/60 * 2 * pi * r_mean_c;
    U_t = N(j)/60 * 2 * pi * r_mean_t;
    fun = 1e6;
    tol = 0.01;

    % Guess a ux in compressor
    for ux_c_guess = 20:0.1:300

        % Get compressor map output
        [pc_ratio, fM025_rev, eta_c, T03_guess, Work_in] =...
            compressorMap(cp, gamma, T025, eta_c_st(ux_c_guess/U_c),...
            ux_c_guess, U_c, n_c, alpha_a_comp, beta_b_comp);
        p03_guess = pc_ratio * p025;
        p04_guess = p03_guess;
        mDot_c_guess = gamma * (((gamma + 1)/2)^(-(gamma + 1)/2/(gamma - 1))) * ...
            p025 * A025 * fM025_rev/sqrt(gamma * R * T025);

        % work balance
        wc = Work_in;
        wt = wc;
        ux_t_guess = (wt/U_t + U_t) / ...
            (tan(abs(alpha_b_turb)) + tan(abs(beta_c_turb)));
        T04_guess = ux_t_guess^2/R * (gamma + 1)/2/gamma;

        % Get turbine map output
        [pt_ratio, fM04_rev, eta_t, T05_guess, Work_out] =...
            turbineMap(cp, gamma, T04_guess, eta_t_st(ux_t_guess,U_t), ...
            ux_t_guess, U_t, n_t, alpha_b_turb, beta_c_turb);
        f_guess = (T04_guess/T03_guess - 1) / ...
            (LHV/(cp * T03_guess) - T04_guess/T03_guess);
        mDot_t_guess = gamma * (((gamma + 1)/2)^(-(gamma + 1)/2/(gamma - 1))) * ...
            p04_guess * A04 * fM04_rev/sqrt(gamma * R * T04_guess);

        % Check mass balance
        fun = mDot_t_guess - mDot_c_guess * (1 + f_guess);
        if (abs(fun) < tol)
            break;
        end
    end

    Op_c_mDot(j) = fM025_rev;
    Op_c_pRatio(j) = pc_ratio;
    Op_t_mDot(j) = fM04_rev;
    Op_t_pRatio(j) = 1 / pt_ratio;
end

% Plot operating lines on compressor and turbine maps
figure(1)
plot(Op_c_mDot, Op_c_pRatio,'r--','LineWidth',3.0)

figure(2)
plot(Op_t_mDot .* N, Op_t_pRatio,'r--','LineWidth',3.0)
```