Contents

UT Power Plant Gas Turbine Analysis Base Case (Phase 1)

assumptions: RH0 = 0%, fuel = natural gas, entire turbine is adiabatic

sb => s bar hb => h bar

```
function [nT1, nT2, Vdot1] = phase1_basecase(TSplot)
```

Set Up

```
% general constants
Rbar = 8.314; % kJ/kmol*K
M = 28.02*.79 + 31.999*.21; % kg/kmol

% conversion values
psi2kPa = 6.895;
R2K = 5/9;
ft2m = 1/3.281;
inH2O2kPa = 1/4.019;
lb2kg = 1/2.205;
BTUlb2kJkg = 2.326;
hr2s = 3600;
```

Input Operating Parameters

```
% ambient air conditions
p1 = 14.417 * psi2kPa; % kPa
T1 = (65 + 459.67) * R2K; % K

% altitude (not used in phase 1)
ALT = 530 * ft2m; % m

% inlet air mass flow rate
mdotin = 189.7 * lb2kg; % kg/s

% inlet pressure loss
delPin = 4 * inH2O2kPa; % kPa

% exhaust pressure loss
delPex = 10 * inH2O2kPa ; % kPa
```

```
% lower heating value
LHV = 19000 * BTUlb2kJkg; % kJ/kg

% fuel mass flow rate
mdotf = 14585 * lb2kg * 1/hr2s; % kg/s
```

Input Design Parameters

```
% compressor pressure ratios
rLPC = 6;
rHPC = 4;

% bypass air mass flow percent
mBP = 0;

% HP compressor parasitic bleed air mass flow percent
mPB = 0;

% compressor efficiencies (percent)
nLPC = .82;
nHPC = .84;

% turbine efficiencies (percent)
% nHPT =
% nLPT =

% generator efficiency (percent)
nGEN = .977;

% base case only
p48 = 71 * psi2kPa; % kPa
WdotELEC = 30.607 * 1000; % kW
```

Analysis

```
% state 1
v1 = (Rbar/M)*T1/p1;
Vdot1 = mdotin*v1;
sb1 = sbarcalc(T1);

% state 1 -> 2: guide vanes
p2 = p1 - delPin;
T2 = T1;
hb2 = hbarcalc(T2);
h2 = hb2 / M;

% state 2 -> 25: LP compressor
p25 = p2 * rLPC;
sb2 = sbarcalc(T2);
sb25s = sb2 + Rbar*log(p25/p2);
T25s = TcalcS(sb25s);
hb25s = hbarcalc(T25s);
h25s = hb25s / M;
h25 = h2 - (h2 - h25s)/nLPC;
hb25 = h25 * M;
T25 = TcalcH(hb25);
```

```
WdotC1 = mdotin*(h25-h2);

% state 25 -> 3: HP compressor
p3 = p25 * rHPC;
sb25 = sbarcalc(T25);
sb3s = sb25 + Rbar*log(p3/p25);
T3s = TcalcS(sb3s);
hb3s = hbarcalc(T3s);
h3s = hb3s / M;
h3 = h25 - (h25 - h3s)/nHPC;
hb3 = h3 * M;
T3 = TcalcH(hb3);
sb3 = sbarcalc(T3);
WdotC2 = mdotin*(h3-h25);

% state 3 -> 4: combustor
mdot4 = mdotin + mdotf;
Qdot = mdotf * LHV;
h4 = (Qdot + mdotin*h3)/mdot4;
hb4 = h4 * M;
T4 = TcalcH(hb4);
sb4 = sbarcalc(T4);
p4 = p3;

% state 4 -> 48: HP turbine
WdotT1 = WdotC1 + WdotC2;
h48 = h4 - (WdotT1/mdot4);
hb48 = h48 * M;
T48 = TcalcH(hb48);
sb48 = sbarcalc(T48);
sb48s = sb4 + Rbar*log(p48/p4);
T48s = TcalcS(sb48s);
hb48s = hbarcalc(T48s);
h48s = hb48s / M;
nT1 = (h4-h48)/(h4-h48s);

% state 48 -> 5: LP turbine
h5 = h48 - (WdotELEC/nGEN)/mdot4;
hb5 = h5 * M;
T5 = TcalcH(hb5);
sb5 = sbarcalc(T5);
p5 = p1 + delPex;
sb5s = sb48 + Rbar*log(p5/p48);
T5s = TcalcS(sb5s);
hb5s = hbarcalc(T5s);
h5s = hb5s / M;
nT2 = (h48-h5)/(h48-h5s);

% state 6: after nozzle
p6 = p1;
T6 = T5;
sb6 = sbarcalc(T6);
```

Output Performance Parameters

table of T and p at each station, vol. flow rate at inlet,

```
states = [1 2 25 3 4 48 5 6]';
allT = 1/R2K*[T1 T2 T25 T3 T4 T48 T5 T6]';
allp = 1/psi2kPa*[p1 p2 p25 p3 p4 p48 p5 p6]';

data = [states allT, allp];

BaseCaseStates = array2table(data, "VariableNames", ["state", "temp (K)", "pressure (kPa)"])

% turbine efficiencies
nT1
nT2
```

T–s diagram

TSplot = 0;

```
if TSplot == 1
    % Reference for real entropy change (TS (not taylor swift yet though :( ) diagram)
    sbref = 0;

    % calculate specific entropies
    sb1r = sbref;
    sb2r = sb1r + sb2 - sb1 - Rbar*log(p2/p1);
    sb25r = sb2r + sb25 - sb2 - Rbar*log(p25/p2);
    sb25rs = sb2r + sb25s - sb2 - Rbar*log(p25/p2);
    sb3rs = sb25r + sb3s - sb25 - Rbar*log(p3/p25);
    sb3r = sb25r + sb3 - sb25 - Rbar*log(p3/p25);
    sb4r = sb3r + sb4 - sb3 - Rbar*log(p4/p3);
    sb48r = sb4r + sb48 - sb4 - Rbar*log(p48/p4);
    sb48rs = sb4r + sb48s - sb4 - Rbar*log(p48/p4);
    sb5r = sb48r + sb5 - sb48 - Rbar*log(p5/p48);
    sb5rs = sb48r;
    sb6r = sb5r + sb6 - sb5 - Rbar*log(p6/p5);


    % Collect specific entropies (kJ/kg-K). Divide s̄ (kJ/kmol-K) by M (kg/kmol).
    allS = [ ...
        sb1r/M
        sb2r/M
        sb25r/M
        sb3r/M
        sb4r/M
        sb48r/M
        sb5r/M
        sb6r/M];

    figure
    hold on

    % Plot 1–2–25–3
    plot(allS(1:4), [T1 T2 T25 T3], 'o-b', 'LineWidth', 1, 'MarkerSize', 6)

    % Plot 4–48–5–6 separately
    plot(allS(5:8), [T4 T48 T5 T6], 'o-b', 'LineWidth', 1, 'MarkerSize', 6)

    xlabel('Specific Entropy s (kJ/kg-K)')
```

```
ylabel('Temperature T (K)')
title('Gas Turbine T–s Diagram, Base Case')
grid on

% Label points
text(allS, [T1 T2 T25 T3 T4 T48 T5 T6], string(states), ...
    'VerticalAlignment','bottom','HorizontalAlignment','right')

% --- add isentropic points
S_iso = [sb25rs/M, sb3rs/M, sb48rs/M, sb5rs/M];
T_iso = [T25s,     T3s,     T48s,    T5s];
labels_iso = {'25s','3s','48s','5s'};

hold on
scatter(S_iso, T_iso, 60, [0.5 0.5 0.5], 'x', 'LineWidth', 1.5)

% annotate isentropic points
text(S_iso, T_iso, labels_iso, ...
    'VerticalAlignment','top', 'HorizontalAlignment','left', ...
    'Color', [0.5 0.5 0.5],'FontSize',6)

% --- define turbine inlets and connect to isentropic outlets
S_in   = [sb2r/M, sb25r/M, sb4r/M, sb48r/M];   % entropy at turbine inlets
T_in   = [T2,     T25,     T4,     T48];       % turbine inlet temperatures

for k = 1:length(S_iso)
    plot([S_iso(k) S_iso(k)], [T_in(k) T_iso(k)], ...
         '--', 'Color', [0.5 0.5 0.5], 'LineWidth', 0.8)
end

% Pressures to plot
% p_iso = [p1 p2 p25 p3 p4 p48 p5];
% colors = lines(length(p_iso));

% for k = 1:length(p_iso)
%     iso_curve = ts_isobar(p_iso(k), T1, 2000, p1, Rbar, T1);
%     plot(iso_curve(:,1)/M, iso_curve(:,2), '--', 'Color', [0.5 0.5 0.5], 'LineWidth', 1)
% end

iso_curve_25s_25 = ts_isobar(p25, T1, T25, p1, Rbar, T25s);
plot(iso_curve_25s_25(:,1)/M, iso_curve_25s_25(:,2), '--', 'Color', [0.5 0.5 0.5], 'LineWidth', 1)

iso_curve_3s_3 = ts_isobar(p3, T1, T3, p1, Rbar, T3s);
plot(iso_curve_3s_3(:,1)/M, iso_curve_3s_3(:,2), '--', 'Color', [0.5 0.5 0.5], 'LineWidth', 1)

iso_curve_3_4 = ts_isobar(p3, T1, T4, p1, Rbar, T3);
plot(iso_curve_3_4(:,1)/M, iso_curve_3_4(:,2), '-', 'Color', [0 0 1], 'LineWidth', 1)

iso_curve_48s_48 = ts_isobar(p48, T1, T48, p1, Rbar, T48s);
plot(iso_curve_48s_48(:,1)/M, iso_curve_48s_48(:,2), '--', 'Color', [0.5 0.5 0.5], 'LineWidth', 1)

iso_curve_5s_5 = ts_isobar(p5, T1, T5, p1, Rbar, T5s);
plot(iso_curve_5s_5(:,1)/M, iso_curve_5s_5(:,2), '--', 'Color', [0.5 0.5 0.5], 'LineWidth', 1)

iso_curve_1_6 = ts_isobar(p6, T1, T6, p1, Rbar, T1);
plot(iso_curve_1_6(:,1)/M, iso_curve_1_6(:,2), '--', 'Color', [0 0 1], 'LineWidth', 1)
```

```
        hold off
    end
```

```
end
```

Published with MATLAB® R2024b