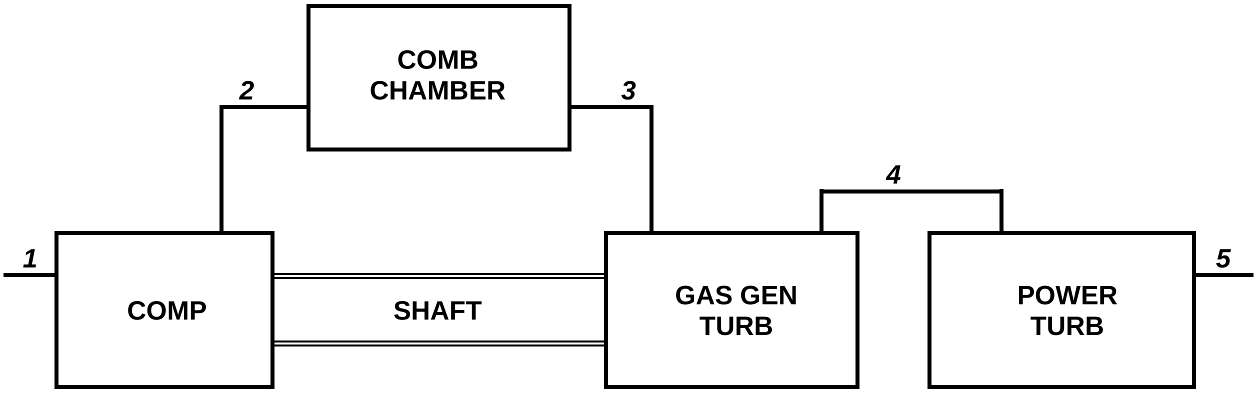
June Kwon

Gas-Turbine Engine System Modeling & Analysis



"A Brayton-cycle gas-turbine system is operating under the following ***standard working condition****. The system is to be compared with few different working conditions. It is a company's interest to find how each working condition compares to the standard working condition, and which of them yields the best option to maximize the net-system efficiency."*

[Standard Working Condition]

- A gas-turbine uses liquid n-octane at 537 R as fuel. The air inlet conditions to the compressor are 14.7 psia and 560 R with a mass flow rate of 50 lbm/sec. The compressor pressure ratio is 21.809, and the actual air exit temperature of the compressor is 1450 R.

  There is a pressure drop through the combustion chamber which is 10% of the compressor discharge pressure. The exit temperature from the combustion chamber is 2800 R.

  The exit pressure of the gas generator turbine is 51.31 psia. There are mechanical losses (due to bearing friction, lube oil pump requirements, etc.) equal to 9.47% of the compressor power requirement. These losses are also supplied by the gas generator turbine. The power turbine exhaust pressure is 15.28 psia and the actual exit temperature is 1500 R

  Assume the working substance flowing through the entire system (including the turbines) is air; assume no changes in kinetic or potential energy.

- From the standard working condition above,

Compressor Inlet Temperature	:	$T_1 = 560\ R$
Compressor Outlet Temperature	:	$T_{2a} = 1450\ R$
Combustor Outlet Temperature	:	$T_{3a} = 2800\ R$
Power Turbine Outlet Temperature	:	$T_{5a} = 1500\ R$
Compressor Inlet Pressure	:	$P_1 = 14.7\ psia$
Gas Generator Turbine Outlet Pressure	:	$P_{4a} = 51.31\ psia$
Power Turbine Outlet Pressure	:	$P_{5a} = 15.28\ psia$
Mass Flow Rate	:	$\dot{m} = 50\ lbm/sec$
Compressor Pressure Ratio	:	$r_{pc} = 21.809$

- Also, find :

1) the fuel/air mass flow ratio required for the combustion conditions, using the LVH approach, f' (lbm_{fuel} / lbm_{air}).
2) the mass flow rate leaving the combustion chamber, $\dot{m}_3$ (lbm/s)
3) the actual power required by the compressor, $\dot{W}_c$ (Btu/s)
4) the compressor efficiency, η_c (%)
5) the actual power which must be produced by the gas-generator turbine to drive the compressor and overcome the losses, $\dot{W}_{ggt}$ (Btu/s)
6) the actual enthalpy at the exit of the gas-generator turbine, h_{4a} (Btu/lbm)
7) the energy added in the combustion chamber, $\dot{Q}_a$ (Btu/s)
8) the efficiency of the gas generator turbine, η_{ggt} (%)
9) the net power produced by the overall gas-turbine system, $\dot{W}_{net}$ (Btu/s)
10) the net-cycle efficiency, η_{net} (%)

First, to find the computational model of the gas-turbine engine system, the operation with "Standard Working Condition" is first to be analysed.

- June Kwon

$$r_{pc} = 21.809 \rightarrow P_2 = P_1 r_{pc} = 14.7(21.809) = 320.5923$$

① $T_1 = 560\ R$ $\rightarrow$ $h_1 = 3.9$ Btu/lbm

$P_1 = 14.7$ psia $\qquad P_{r1} = 1.5781$

(2a) $T_{2a} = 1450\ R$ $\rightarrow$ $h_{2a} = 226.2$ Btu/lbm

$P_{2a} = 320.5923$ psia $\qquad P_{r2a} = 49.188$

(2s) $P_{r2s} = r_p(P_{r1}) = 21.809(1.5781) = 34.4168$

$\downarrow$

$h_{2s} = 192.2994$ Btu/lbm

$T_{2s} = 1319.998\ R$

$P_{2s} = 320.5923$ psia

Thus,

$$\eta_c = \frac{h_{2s} - h_1}{h_{2a} - h_1} = \frac{192.2994 - 3.9}{226.2 - 3.9} = 0.8475 \rightarrow 84.75\%$$

compressor efficiency

③ $\overset{\swarrow 10\%\ \text{Drop}}{}$

$$P_3 = 0.9\,P_2 = 0.9(320.5923) = 288.533 \text{ psia}$$

$$T_3 = 2800\,R$$

$$\downarrow$$

$$h_3 = 602.5 \text{ Btu/lbm}$$

$$P_{r3} = 702.2$$

ok, first, let's find fuel flow... $\dot{m}_{air} = 50$ lbm/sec

$$\dot{m}_3 h_3 = \dot{m}_f (LHV_{C_8H_{18}}) + \dot{m}_2 h_{2a}$$

$$(\dot{m}_f + 50)\,602.5 = \dot{m}_f(19098) + 50(226.2)$$

$$\rightarrow \left[\dot{m}_f = 1.01727 \text{ lbm/sec}\right]$$

Then, we know

$$\dot{W}_{ggt} = \dot{W}_c + \overset{9.47\%}{Loss} = 1.0947\,\dot{W}_c$$

$$\dot{m}_{ggt}(h_3 - h_{4a}) = (1.0947)\,\dot{m}_2(h_{2a} - h_1)$$

$$(50 + 1.01727)(602.5 - h_{4a}) = (1.0947)(50)(226.2 - 3.9)$$

$$\rightarrow \left[h_{4a} = 364 \text{ Btu/lbm}\right]_{\shortparallel}$$

Thus...

4a $h_{4a} = 364$ Btu/lbm

$\downarrow$

$T_{4a} = 1960$ R

$P_{r4a} = 160.84$

$P_{4a} = 51.31$ psia (given)

Next, we know...

$$r_{pggt} = \frac{P_{r3}}{P_{r4s}} = \frac{P_3}{P_4}$$

Thus, $\longrightarrow P_{r4s} = P_{r3}\left(\frac{P_4}{P_3}\right)$

$$= 702.2\left(\frac{51.31}{288.533}\right) = \left[124.8726\right]$$

Thus...

4s $P_{r4s} = 124.8726$

$\downarrow$

$T_{4s} = 1840$ R

$P_{4s} = 51.31$ psia

$h_{4s} = 331.0$ Btu/lbm

Therefore, the efficiency of Gas Generator Turbine is...

$$\eta_{ggt} = \frac{h_3 - h_{4a}}{h_3 - h_{4s}} = \frac{602.5 - 364}{602.5 - 331} = 0.8784$$

$$= \boxed{87.84\ \%}$$

Next...

5a

$T_{5a} = 1500\ R$

$\downarrow$

$h_{5a} = 239.4$ Btu/lbm

$P_{r5a} = 56.03$

$P_{5a} = 15.28$ psia

Then, we know...

$$r_{ppt} = \frac{P_{r4a}}{P_{r5s}} = \frac{P_4}{P_5} \longrightarrow P_{r5s} = P_{r4a}\left(\frac{P_5}{P_4}\right) = 160.84\left(\frac{15.28}{51.31}\right)$$

$$\longrightarrow P_{r5s} = \boxed{47.8978}$$

Thus,

5s

$P_{r5s} = 47.8978 \longrightarrow$

$h_{5s} = 223.5913$ Btu/lbm

$T_{5s} = 1439.97\ R$

$P_{5s} = 15.28$ psia

Thus ...

$$\eta_{pt} = \frac{h_{4a} - h_{5a}}{h_{4a} - h_{5s}} = \frac{364 - 239.4}{364 - 223.5913} = 0.8874$$

$$= \boxed{88.74\%}$$

Therefore ....

$$W_c = h_{2a} - h_1 = 226.2 - 3.9 = 222.3 \text{ Btu/lbm}$$

$$\dot{W}_c = \dot{m}_c W_c = 50(222.3) = 11115 \text{ Btu/sec}$$

$$W_{ggt} = h_3 - h_{4a} = 602.5 - 364 = 238.5 \text{ Btu/lbm}$$

$$\dot{W}_{ggt} = \dot{m}_{ggt} W_{ggt} = 51.01727(238.5) = 12167.6189 \frac{\text{Btu}}{\text{sec}}$$

$$W_{net} = W_{pt} = h_{4a} - h_{5a} = 364 - 239.4 = 124.6 \text{ Btu/lbm}$$

$$\dot{W}_{net} = \dot{W}_{pt} = \dot{m}_{pt} W_{pt} = 51.01727(124.6)$$

$$= 6356.752 \text{ Btu/sec}$$

$$\dot{Q}_a = \dot{m}_3 h_3 - \dot{m}_2 h_{2a} = 51.01727(602.5) - 50(226.2)$$

$$= 19427.905 \text{ Btu/sec}$$

$$\eta_{cycle} = \frac{\dot{W}_{net}}{\dot{Q}_a} = \frac{\dot{W}_{pt}}{\dot{Q}_a} = \frac{6356.752}{19427.905} = 0.3272$$

$$= \boxed{32.72\%}$$

Thus....

① Fuel/Air mass ratio?

$$f = \frac{F}{A} = \frac{\dot{m}_{fuel}}{\dot{m}_{air}} = \frac{1.01727 \text{ lbm/sec}}{50 \text{ lbm/sec}} = \boxed{0.02034 \frac{\text{lbm fuel}}{\text{lbm air}}}$$

② $\dot{m}_3$?

$$\dot{m}_3 = \dot{m}_{fuel} + \dot{m}_{air} = 1.01727 + 50 = \boxed{51.01727 \text{ lbm/sec}}''$$

③ $\dot{W}_c$?

Found in the back... $\dot{W}_c = \boxed{11115 \text{ Btu/sec}}''$

④ η_c?

Found in the back... $\eta_c = \boxed{84.75\%}''$

⑤ $\dot{W}_{ggt}$?

Found in the back... $\dot{W}_{ggt} = \boxed{12167.6189 \text{ Btu/sec}}''$

⑥ h_{4a}?

$h_{4a} = \boxed{364 \text{ Btu/lbm}}$ Found in the back...

⑦ $\dot{Q}_a$?

Found in the back... $\dot{Q}_a = \boxed{19427.905 \text{ Btu/sec}}$

⑧ η_{ggt}?

Found in the back... $\eta_{ggt} = \boxed{87.84\%}$

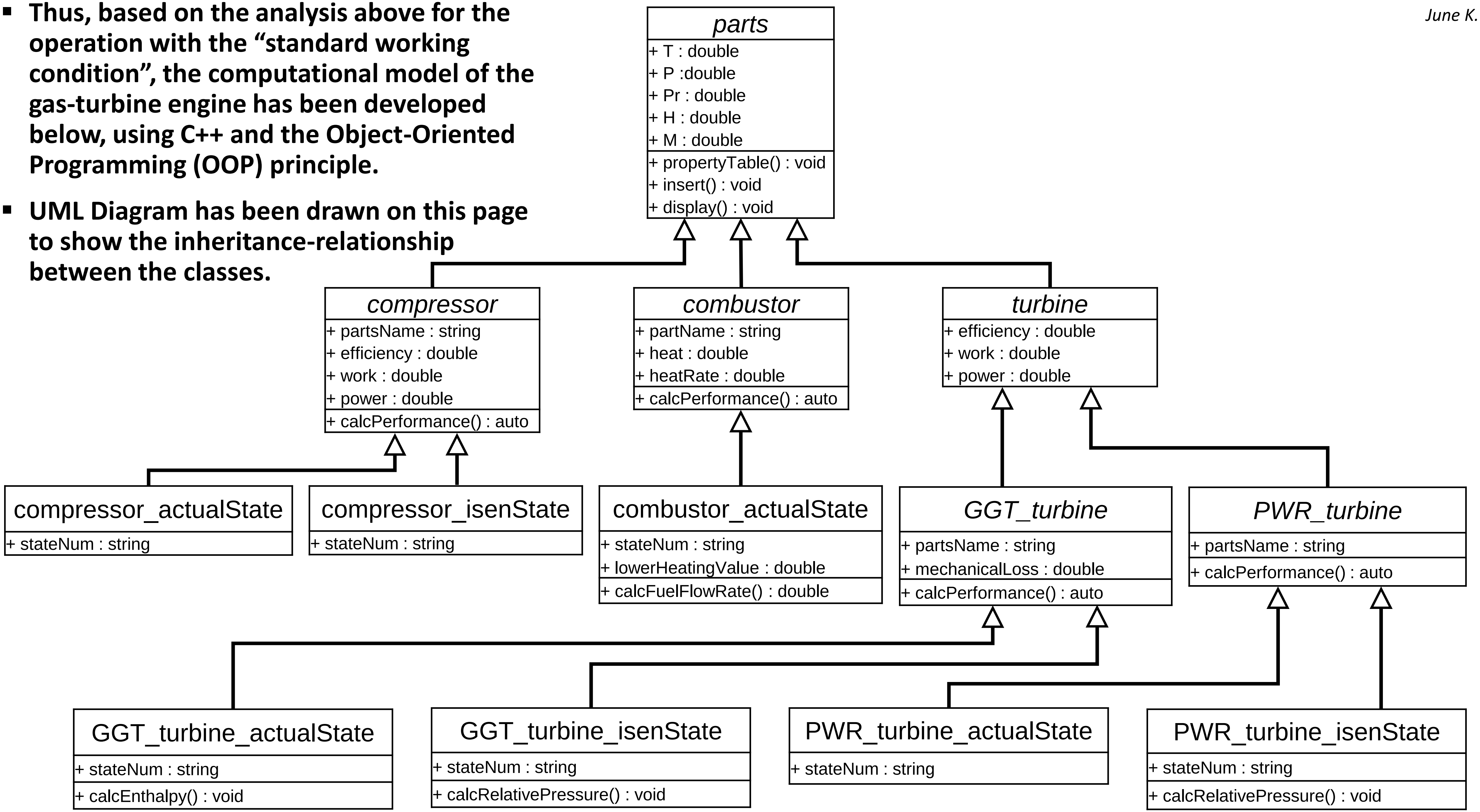
⑨ $\dot{W}_{net} = \dot{W}_{pt} = ?$

Found in the back... $\dot{W}_{pt} = \boxed{6356.752 \text{ Btu/sec}} = \dot{W}_{net}$

⑩ η_{cycle}?

Found in the back... $\eta_{cycle} = \boxed{32.72\%}$

- **Thus, based on the analysis above for the operation with the "standard working condition", the computational model of the gas-turbine engine has been developed below, using C++ and the Object-Oriented Programming (OOP) principle.**
- **UML Diagram has been drawn on this page to show the inheritance-relationship between the classes.**



[Please Note]

- **Below is the C++ code for this computational model.**
- **It is 20 pages long. (I've put it here just in case you lose the main file.)**
- **There is no need to read the whole code with details in this document.**
- **Therefore, with the UML diagram shown above, please skim through the code.**
- **You can also run and check my code at Replit (link below) :**

gasTurbine - Replit

Thank you very much,

June Kwon

```cpp
/** [Computational Model of Gas Turbine Engine]
    : This program allows simulation of the general gas-turbine
    engine to evaluate its performance at various conditions.
    (it requires "AirTable.txt" for property-table.)
    @file     main.cpp
    @author   June Kwon
    @version  1.2
*/

#include <iostream>
#include <fstream>
using namespace std;
// #include <iomanip>
// #include <string>
// #include <stdio.h>
// #include <math.h>


//++++++++++++++++++++++++++++[Super Class : "parts"]++++++++++++++++++++++++++++

/** [parts]
    : "parts" is the base class for various components throughout
    the entire program. It has thermodynamic properties such as
    temperature, pressure, relative pressure, enthalpy, etc.
*/
class parts
{
    public:
        double T;                   // Temperature       (R)
        double P;                   // Pressure          (psia)
        double Pr;                  // Relative Pressure
        double H;                   // Enthalpy          (Btu/lbm)
        double M;                   // Mass Flow Rate    (lbm/sec)
        void propertyTable(char);   // Look Up Property Table
        void assign(char,double);   // Assign Values
        void display();             // Display Information
};

/** [propertyTable]
    : "propertyTable" method accesses the file "AirTable.txt"
    to find the matching values for the unknown properties by
    specifying the known property
    @param - Known Property (EX: T, H, R)
*/
```

```
void parts::propertyTable(char knownProperty)
{
    // Initialization
    string line, vec[3];
    double table_T, table_H, table_Pr;
    int i, pos;

    // Access "AirTable.txt" & File Check
    ifstream infile;
    infile.open("AirTable.txt");
    if (!infile)
    {
        cout << "Cannot Open File!\n";
        exit(1);
    }

    // Find Values from Table
    while (getline(infile, line))
    {
        for  (i = 0; i < 3; i++)
        {
            pos = line.find(',');
            vec[i] = line.substr(0, pos);
            line = line.substr(pos + 1);
        }
        table_T  = atof(vec[0].c_str());
        table_H  = atof(vec[1].c_str());
        table_Pr = atof(vec[2].c_str());

        // If T is known, then H and Pr will be found from table
        if (knownProperty == 'T' && this->T <= table_T)
        {
            H = table_H;
            Pr = table_Pr;
            break;
        }

        // If H is known, then T and Pr will be found from table
        if (knownProperty == 'H' && this->H <= table_H)
        {
            T = table_T;
            Pr = table_Pr;
            break;
        }
```

```
        // If R is known, then T and H will be found from table
        if (knownProperty == 'R' && this->Pr <= table_Pr)
        {
            T = table_T;
            H = table_H;
            break;
        }
    }
    infile.close();
}

/** [assign]
    : "assign" method assigns the value for the known property
    by specifying the known property with its value
    @param - Known Property (EX: T, H, R)
    @param - Value for the known property
*/
void parts::assign(char knownProperty, double value)
{
    switch (knownProperty)
    {
        // If the known property is temperature (T),
        // the user specified value will be assigned to T
        case 'T':
            T = value;
            break;

        // If the known property is pressure (P),
        // the user specified value will be assigned to P
        case 'P':
            P = value;
            break;

        // If the known property is relative pressure (Pr),
        // the user specified value will be assigned to Pr
        case 'R':
            Pr = value;
            break;

        // If the known property is enthalpy (H),
        // the user specified value will be assigned to H
        case 'H':
            H = value;
```

```
            break;

        // If the known property is mass flow rate (M),
        // the user specified value will be assigned to M
        case 'M':
            M = value;
    }
}

/** [display]
    : "parts::display" prints the entire thermodynamic properties
    (T, P, Pr, H, M) of the object.
*/
void parts::display()
{
    cout << "\nTemperature:\t\t" << T << " \t\t(R)";
    cout << "\nPressure:\t\t\t" << P << "\t\t(psia)";
    cout << "\nRelative Pressure:\t" << Pr;
    cout << "\nEnthalpy:\t\t\t" << H << " \t\t(Btu/lbm)";
    cout << "\nMass Flow Rate:\t\t" << M << "  \t\t(lbm/sec)";
    cout << endl;
}

//++++++++++++++++++++++++++[Sub Class : "compressor"]++++++++++++++++++++++++++

// Forward Declaration
class compressor_actualState;
class compressor_isenState;

/** [compressor]
    : "compressor" is a derived class that inherits from "parts" class.
    In addition to what is derived from "parts" class, "compressor"
    class has efficiency, work, power, and partsName.
*/
class compressor : public parts
{
    public:
        string partsName = "Compressor";
        double efficiency;
        double work;
        double power;
        auto calcPerformance(compressor_actualState, compressor_actualState,
compressor_isenState);
};
```

```
//++++++++++++++++++[Sub Sub Class : "compressor_actualState"]++++++++++++++++++

/** [compressor_actualState]
    : "compressor_actualState" is a derived class that inherits from
    "compressor" class. This class is to analyze the performance of
    the compressor at its actual state.
    (In addition to "compressor" class, this class has stateNum.)
*/
class compressor_actualState : public compressor
{
    public:
        compressor_actualState(string num) // Constructor
        {
            T  = 0;
            P  = 0;
            Pr = 0;
            H  = 0;
            M  = 0;
            stateNum = num;
        }
        string stateNum;
        void display();
        friend auto compressor::calcPerformance(compressor_actualState,
compressor_actualState, compressor_isenState);
};

/** [display]
    : "compressor_actualState::display" overrides the "parts::display".
    In addition, partsName and stateNum are printed.
*/
void compressor_actualState::display()
{
    cout << "\n----------------[State " << stateNum << "]----------------";
    cout << "\nPart Name:\t\t\t" << partsName;
    cout << "\nState #:\t\t\t" << stateNum << "\t\t\t(Actual)";
    parts::display(); // Override
}

//++++++++++++++++++[Sub Sub Class : "compressor_isenState"]++++++++++++++++++

/** [compressor_isenState]
    : "compressor_isenState" is a derived class that inherits from
    "compressor" class. This class is to analyze the performance of
```

```
    the compressor at its isentropic state.
    (In addition to "compressor" class, this class has stateNum.)
*/
class compressor_isenState : public compressor
{
    public:
        compressor_isenState(string num) // Constructor
        {
            T  = 0;
            P  = 0;
            Pr = 0;
            H  = 0;
            M  = 0;
            stateNum = num;
        }
        string stateNum;
        void display();
        friend auto calcPerformance(compressor_actualState,
compressor_actualState, compressor_isenState);
};

/** [display]
    : "compressor_isenState::display" overrides the "parts::display".
    In addition, partsName and stateNum are printed.
*/
void compressor_isenState::display()
{
    cout << "\n-----------------[State " << stateNum << "]-----------------";
    cout << "\nPart Name:\t\t\t" << partsName;
    cout << "\nState #:\t\t\t" << stateNum << "\t\t\t(Isentropic)";
    parts::display(); // Override
}

/** [calcPerformance]
    : "calcPerformance" calculates the performance of the compressor
    (efficiency, work, power) and print the result.
    @param - Actual Compressor Inlet State
    @param - Actual Compressor Outlet State
    @param - Isentropic Compressor Outlet State
    @return - Efficiency, Work, Power of Compressor
*/
auto compressor::calcPerformance(compressor_actualState actPtr1,
compressor_actualState actPtr2, compressor_isenState isnPtr2)
{
```

```cpp
    struct result
    {
        double value_efficiency;
        double value_work;
        double value_power;
    };

    // Performance Calculation
    efficiency = (isnPtr2.H - actPtr1.H) / (actPtr2.H - actPtr1.H) * 100;
    work = (actPtr2.H - actPtr1.H);
    power = (M * work);

    // Print
    cout << "\n------------[Performance of Compressor]------------";
    cout << "\nEfficiency of Compressor:\t\t" << efficiency << "\t(%)";
    cout << "\nWork Done on Compressor:\t\t" << work << "\t(Btu/lbm)";
    cout << "\nPower Required by Compressor:\t" << power << "\t(Btu/sec)";
    cout << endl;

    return result {efficiency, work, power};
}

//+++++++++++++++++++++++++++[Sub Class : "combustor"]+++++++++++++++++++++++++++

// Forward Declaration
class combustor_actualState;

/** [combustor]
    : "combustor" is a derived class that inherits from "parts" class.
    In addition to what is derived from "parts" class, "combustor"
    class has heat, heatRate and partsName.
*/
class combustor : public parts
{
    public:
        string partsName = "Combustor";
        double heat;
        double heatRate;
        auto calcPerformance(combustor_actualState, compressor_actualState);
};

//+++++++++++++++++++[Sub Sub Class : "combustor_actualState"]+++++++++++++++++++

/** [combustor_actualState]
```

```
    : "combustor_actualState" is a derived class that inherits from
    "combustor" class. This class is to analyze the performance of
    the combustor at its actual state.
    (In addition to "combustor" class, this class has stateNum and
    lowerHeatingValue.)
*/
class combustor_actualState : public combustor
{
    public:
        combustor_actualState(string num1, double num2) // Constructor
        {
            T  = 0;
            P  = 0;
            Pr = 0;
            H  = 0;
            M  = 0;
            stateNum = num1;
            lowerHeatingValue = num2;
        }
        string stateNum;
        double lowerHeatingValue;
        double calcFuelFlowRate(combustor_actualState, compressor_actualState);
        void display();
        friend auto combustor::calcPerformance(combustor_actualState,
compressor_actualState);
};

/** [calcFuelFlowRate]
    : "calcFuelFlowRate" calculates the fuel flow rate into the combustor,
    required to satisfy the compressor power requirement and returns
    the result.
    @param - Actual Combustor State
    @param - Actual Compressor Outlet State
    @return - Fuel Flow Rate into Combustor
*/
double combustor_actualState::calcFuelFlowRate(combustor_actualState cmbPtr,
compressor_actualState cprPtr)
{
    double fuelFlowRate = (((cprPtr.M * cprPtr.H) - (cmbPtr.M * cmbPtr.H)) /
(cmbPtr.H - cmbPtr.lowerHeatingValue));
    this->M = fuelFlowRate + this->M; // Update Current Fuel Flow Rate
    return fuelFlowRate;
}
```

```cpp
/** [display]
    : "combustor_actualState::display" overrides the "parts::display".
    In addition, partsName and stateNum are printed.
*/
void combustor_actualState::display()
{
    cout << "\n-----------------[State " << stateNum << "]-----------------";
    cout << "\nPart Name:\t\t\t" << partsName;
    cout << "\nState #:\t\t\t" << stateNum << "\t\t\t(Actual)";
    parts::display(); // Override
}


/** [calcPerformance]
    : "calcPerformance" calculates the performance of the combustor
    (heat, heatRate) and print the result.
    @param - Actual Combustor State
    @param - Actual Compressor Outlet State
    @return - Heat and Heat Rate of Compressor
*/
auto combustor::calcPerformance(combustor_actualState cmbPtr,
compressor_actualState cprPtr)
{
    struct result
    {
        double value_heat;
        double value_heatRate;
    };

    // Performance Calculation
    heat = (cmbPtr.H - cprPtr.H);
    heatRate = (cmbPtr.M * cmbPtr.H) - (cprPtr.M * cprPtr.H);

    // Print
    cout << "\n------------[Performance of Combustor]------------";
    cout << "\nFuel Flow Rate into Combustor:\t" << cmbPtr.M << "\t(lbm/sec)";
    cout << "\nHeat Added to Combustor:\t\t" << heat << "\t(Btu/lbm)";
    cout << "\nHeat Rate Added to Combustor:\t" << heatRate << "\t(Btu/sec)";
    cout << endl;

    return result {heat, heatRate};
}
```

```cpp
//+++++++++++++++++++++++++++++[Sub Class : "turbine"]+++++++++++++++++++++++++++++

/** [turbine]
    : "turbine" is a derived class that inherits from "parts" class.
    In addition to what is derived from "parts" class, "turbine"
    class has efficiency, work, and power.
*/
class turbine : public parts
{
    public:
        double efficiency;
        double work;
        double power;
};

//+++++++++++++++++++++++++[Sub Sub Class : "GGT_turbine"]+++++++++++++++++++++++++

// Forward Declaration
class GGT_turbine_actualState;
class GGT_turbine_isenState;

/** [GGT_turbine]
    : "GGT_turbine" is a derived class that inherits from "turbine" class.
    (In addition to "turbine", this class has partsName and mechanicalLoass.)
    (GGT stands for Gas Generator Turbine.)
*/
class GGT_turbine : public turbine
{
    public:
        string partsName = "Gas Generator Turbine";
        double mechanicalLoss;
        auto calcPerformance(GGT_turbine_actualState, GGT_turbine_isenState,
combustor_actualState);
};

//+++++++++++++++[Sub Sub Sub Class : "GGT_turbine_actualState"]++++++++++++++++

/** [GGT_turbine_actualState]
    : "GGT_turbine_actualState" is a derived class that inherits from
    "GGT_turbine" class. This class is to analyze the performance of
    the gas generator turbine at its actual state.
    (In addition to "GGT_turbine" class, this class has stateNum.)
    (GGT stands for Gas Generator Turbine.)
*/
```

```cpp
class GGT_turbine_actualState : public GGT_turbine
{
    public:
        GGT_turbine_actualState(string num1, double num2) // Constructor
        {
            T  = 0;
            P  = 0;
            Pr = 0;
            H  = 0;
            M  = 0;
            stateNum = num1;
            mechanicalLoss = num2;
        }
        string stateNum;
        void calcEnthalpy(GGT_turbine_actualState, combustor_actualState,
compressor_actualState, compressor_actualState);
        void display();
        friend auto GGT_turbine::calcPerformance(GGT_turbine_actualState,
GGT_turbine_isenState, combustor_actualState);
};


/** [calcEnthalpy]
    : "calcEnthalpy" calculates the enthalpy of the gas generator turbine.
    This calculation involves mechanical losses.
    @param - Actual Gas Generator Turbine State
    @param - Actual Combustor State
    @param - Actual Compressor Inlet State
    @param - Actual Compressor Outlet State
*/
void GGT_turbine_actualState::calcEnthalpy(GGT_turbine_actualState ggtPtr,
combustor_actualState cmbPtr, compressor_actualState cprPtr1,
compressor_actualState cprPtr2)
{
    // Enthalpy Calculation
    this->H = (((1 + ggtPtr.mechanicalLoss) * (cprPtr2.M) * (cprPtr1.H -
cprPtr2.H)) / ggtPtr.M) + cmbPtr.H;
}


/** [display]
    : "GGT_turbine_actualState::display" overrides the "parts::display".
    In addition, partsName and stateNum are printed.
*/
```

```cpp
void GGT_turbine_actualState::display()
{
    cout << "\n-----------------[State " << stateNum << "]----------------";
    cout << "\nPart Name:\t\t\t" << partsName;
    cout << "\nState #:\t\t\t" << stateNum << "\t\t\t(Actual)";
    parts::display(); // Override
}

//+++++++++++++++++[Sub Sub Sub Class : "GGT_turbine_isenState"]+++++++++++++++++

/** [GGT_turbine_isenState]
    : "GGT_turbine_isenState" is a derived class that inherits from
    "GGT_turbine" class. This class is to analyze the performance of
    the gas generator turbine at its isentropic state.
    (In addition to "GGT_turbine" class, this class has stateNum.)
    (GGT stands for Gas Generator Turbine.)
*/
class GGT_turbine_isenState : public GGT_turbine
{
    public:
        GGT_turbine_isenState(string num1, double num2) // Constructor
        {
            T  = 0;
            P  = 0;
            Pr = 0;
            H  = 0;
            M  = 0;
            stateNum = num1;
            mechanicalLoss = num2;
        }
        string stateNum;
        void calcRelativePressure(GGT_turbine_isenState, combustor_actualState);
        void display();
        friend auto GGT_turbine::calcPerformance(GGT_turbine_actualState,
GGT_turbine_isenState, combustor_actualState);
};

/** [calcRelativePressure]
    : "calcRelativePressure" calculates the relative pressure of the gas
    generator turbine. This calculation involves the pressure and the
    relative pressure of the combustor.
    @param - Isentropic Gas Generator Turbine State
    @param - Actual Combustor State
*/
```

```cpp
void GGT_turbine_isenState::calcRelativePressure(GGT_turbine_isenState ggtPtr,
combustor_actualState cmbPtr)
{
    this->Pr = cmbPtr.Pr * (ggtPtr.P / cmbPtr.P);
}

/** [display]
    : "GGT_turbine_isenState::display" overrides the "parts::display".
    In addition, partsName and stateNum are printed.
*/
void GGT_turbine_isenState::display()
{
    cout << "\n-----------------[State " << stateNum << "]-----------------";
    cout << "\nPart Name:\t\t\t" << partsName;
    cout << "\nState #:\t\t\t" << stateNum << "\t\t\t(Isentropic)";
    parts::display(); // Override
}

/** [calcPerformance]
    : "calcPerformance" calculates the performance of the gas generator
    turbine (efficiency, work, power) and print the result.
    @param - Actual Gas Generator Turbine State
    @param - Isentropic Gas Generator Turbine State
    @param - Actual Combustor State
    @return - Efficiency, Work, Power of Gas Generator Turbine
*/
auto GGT_turbine::calcPerformance(GGT_turbine_actualState actPtr,
GGT_turbine_isenState isnPtr, combustor_actualState cmbPtr)
{
    struct result
    {
        double value_efficiency;
        double value_work;
        double value_power;
    };

    // Performance Calculation
    efficiency = (cmbPtr.H - actPtr.H) / (cmbPtr.H - isnPtr.H) * 100;
    work = (cmbPtr.H - actPtr.H);
    power = (M * work);

    // Print
    cout << "\n-------[Performance of Gas Generator Turbine]-------";
    cout << "\nEfficiency of GGT Turbine:\t\t" << efficiency << "\t(%)";
```

```cpp
    cout << "\nWork Done by GGT Turbine:\t\t" << work << "\t(Btu/lbm)";
    cout << "\nPower Generated by GGT Turbine:\t" << power << "\t(Btu/sec)";
    cout << endl;

    return result {efficiency,work,power};
}

//++++++++++++++++++++++++[Sub Sub Class : "PWR_turbine"]++++++++++++++++++++++++

// Forward Declaration
class PWR_turbine_actualState;
class PWR_turbine_isenState;

/** [PWR_turbine]
    : "PWR_turbine" is a derived class that inherits from "turbine" class.
    (In addition to "turbine", this class has partsName.)
    (PWR stands for Power Turbine.)
*/
class PWR_turbine : public turbine
{
    public:
        string partsName = "Power Turbine";
        auto calcPerformance(PWR_turbine_actualState, PWR_turbine_isenState,
GGT_turbine_actualState);
};

//++++++++++++++++[Sub Sub Sub Class : "PWR_turbine_actualState"]++++++++++++++++

/** [PWR_turbine_actualState]
    : "PWR_turbine_actualState" is a derived class that inherits from
    "PWR_turbine" class. This class is to analyze the performance of
    the power turbine at its actual state.
    (In addition to "PWR_turbine" class, this class has stateNum.)
    (PWR stands for Power Turbine.)
*/
class PWR_turbine_actualState : public PWR_turbine
{
    public:
        PWR_turbine_actualState(string num) // Constructor
        {
            T  = 0;
            P  = 0;
            Pr = 0;
            H  = 0;
```

```
            M  = 0;
            stateNum = num;
        }
        string stateNum;
        void display();
        friend auto PWR_turbine::calcPerformance(PWR_turbine_actualState,
PWR_turbine_isenState, GGT_turbine_actualState);
};

/** [display]
    : "PWR_turbine_actualState::display" overrides the "parts::display".
    In addition, partsName and stateNum are printed.
*/
void PWR_turbine_actualState::display()
{
    cout << "\n----------------[State " << stateNum << "]----------------";
    cout << "\nPart Name:\t\t\t" << partsName;
    cout << "\nState #:\t\t\t" << stateNum << "\t\t\t(Actual)";
    parts::display(); // Override
}

//+++++++++++++++++[Sub Sub Sub Class : "PWR_turbine_isenState"]+++++++++++++++++

/** [PWR_turbine_isenState]
    : "PWR_turbine_isenState" is a derived class that inherits from
    "PWR_turbine" class. This class is to analyze the performance of
    the power turbine at its isentropic state.
    (In addition to "PWR_turbine" class, this class has stateNum.)
    (PWR stands for Gas Generator Turbine.)
*/
class PWR_turbine_isenState : public PWR_turbine
{
    public:
        PWR_turbine_isenState(string num) // Constructor
        {
            T  = 0;
            P  = 0;
            Pr = 0;
            H  = 0;
            M  = 0;
            stateNum = num;
        }
        string stateNum;
        void calcRelativePressure(PWR_turbine_isenState, GGT_turbine_actualState);
```

```
        void display();
        friend auto PWR_turbine::calcPerformance(PWR_turbine_actualState,
PWR_turbine_isenState, GGT_turbine_actualState);
};

/** [calcRelativePressure]
    : "calcRelativePressure" calculates the relative pressure of the
    power turbine. This calculation involves the pressure and the
    relative pressure of the gas generator turbine.
    @param - Isentropic Power Turbine State
    @param - Actual Gas Generator Turbine State
*/
void PWR_turbine_isenState::calcRelativePressure(PWR_turbine_isenState pwrPtr,
GGT_turbine_actualState ggtPtr)
{
    this->Pr = ggtPtr.Pr * (pwrPtr.P / ggtPtr.P);
}

/** [display]
    : "PWR_turbine_isenState::display" overrides the "parts::display".
    In addition, partsName and stateNum are printed.
*/
void PWR_turbine_isenState::display()
{
    cout << "\n----------------[State " << stateNum << "]----------------";
    cout << "\nPart Name:\t\t\t" << partsName;
    cout << "\nState #:\t\t\t" << stateNum << "\t\t\t(Isentropic)";
    parts::display(); // Override
}

/** [calcPerformance]
    : "calcPerformance" calculates the performance of the power
    turbine (efficiency, work, power) and print the result.
    @param - Actual Power Turbine State
    @param - Isentropic Power Turbine State
    @param - Actual Gas Generator Turbine State
    @return - Efficiency, Work, Power of Power Turbine
*/
auto PWR_turbine::calcPerformance(PWR_turbine_actualState actPtr,
PWR_turbine_isenState isnPtr, GGT_turbine_actualState ggtPtr)
{
    struct result
    {
        double value_efficiency;
```

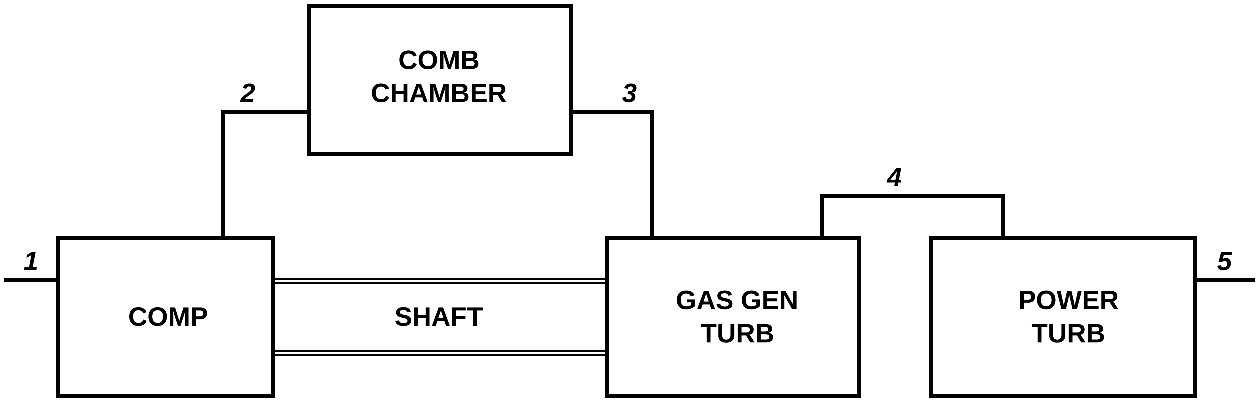
```
        double value_work;
        double value_power;
    };

    // Performance Calculation
    efficiency = (ggtPtr.H - actPtr.H) / (ggtPtr.H - isnPtr.H) * 100;
    work = (ggtPtr.H - actPtr.H);
    power = (M * work);

    // Print
    cout << "\n----------[Performance of Power Turbine]----------";
    cout << "\nEfficiency of PWR Turbine:\t\t" << efficiency << "\t(%)";
    cout << "\nWork Done by PWR Turbine:\t\t" << work << "\t(Btu/lbm)";
    cout << "\nPower Generated by PWR Turbine:\t" << power << "\t(Btu/sec)";
    cout << endl;

    return result {efficiency, work, power};
}

/** [calcNETPerformance]
    : "calcNETPerformance" calculates the performance (efficiency, work, power)
    of the entire gas turbine as a net system and print the result.
    @param - Isentropic Power Turbine State
    @param - Actual Combustor State
    @param - Mass Flow Rate
    @return - Efficiency, Work, Power of Net Gas Turbine System
*/
auto calcNETPerformance(PWR_turbine_isenState isnPtr, combustor_actualState
cmbPtr, double MFR)
{
    struct result
    {
        double value_efficiency;
        double value_work;
        double value_power;
        double value_ratio;
    };

    // Performance Calculation
    double netEfficiency = (isnPtr.power / cmbPtr.heatRate) * 100;
    double netWork = isnPtr.work;
    double netPower = isnPtr.power;
    double netRatio = ((cmbPtr.M - MFR) / MFR);
```

```cpp
    // Print
    cout << "\n---[Net Performance of Entire Gas Turbine System]---";
    cout << "\nEfficiency of Net Cycle System:\t" << netEfficiency << "\t(%)";
    cout << "\nWork Done by Net Cycle System:\t" << netWork << "\t(Btu/lbm)";
    cout << "\nPower Generated by Net System:\t" << netPower << "\t(Btu/sec)";
    cout << "\nFuel/Air Mass Flow Ratio:\t\t" << netRatio;
    cout << endl;

    return result {netEfficiency, netWork, netPower, netRatio};
}

//+++++++++++++++++++++++++++++++++[Main Script]+++++++++++++++++++++++++++++++++

int main()
{
    // Given Conditions
    double RPC = 21.8090;       // Compressor Pressure Ratio
    double MFR = 50.0000;       // Mass Flow Rate (lbm/sec)
    double FFR = 0.00000;       // Fuel Flow Rate (lbm/sec)
    double DRP = 0.10000;       // 10% Pressure Drop
    double MLS = 0.09470;       // 9.47% Mechanical Losses
    double LHV = 19098.0;       // Lower Heating Value of n-Octane (Btu/lbm)
    double T1  = 560.000;       // Compressor Inlet Temperature (R)
    double T2A = 1450.00;       // Compressor Outlet Temperature (R)
    double T3A = 2800.00;       // Combustor Outlet Temperature (R)
    double T5A = 1500.00;       // Power Turbine Outlet Temperature (R)
    double P1  = 14.7000;       // Compressor Inlet Pressure (psia)
    double P4A = 51.3100;       // Gas Generator Turbine Outlet Pressure (psia)
    double P5A = 15.2800;       // Power Turbine Outlet Pressure (psia)

    // 1. Compressor Inlet +++++++++++++++++++++++++++++++++++++++++++++++++++++

    compressor_actualState state1("1");     // Define State 1
    state1.assign('M',MFR);                 // Assign Known Properties at State 1
    state1.assign('T',T1);
    state1.assign('P',P1);
    state1.propertyTable('T');              // Find other unknown properties
    state1.display();                       // Display all properties at State 1

    // 2. Compressor Outlet ++++++++++++++++++++++++++++++++++++++++++++++++++++

    compressor_actualState state2A("2A");   // Define State 2A (Actual)
    state2A.assign('M',MFR);                // Assign Known Properties at State 2A
    state2A.assign('T',T2A);
```

```
    state2A.assign('P',RPC*state1.P);
    state2A.propertyTable('T');                // Find other unknown properties
    state2A.display();                         // Display all properties at State 2A

    compressor_isenState state2S("2S");        // Define State 2S (Isentropic)
    state2S.assign('M',MFR);                   // Assign Known Properties at State 2S
    state2S.assign('P',state2A.P);
    state2S.assign('R',RPC*state1.Pr);
    state2S.propertyTable('R');                // Find other unknown properties
    state2S.display();                         // Display all properties at State 2S

    // 3. Combustor ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

    combustor_actualState state3("3",LHV);     // Define State 3 (Actual)
    state3.assign('M',MFR);                    // Assign Known Properties at State 3
    state3.assign('T',T3A);
    state3.assign('P',(1-DRP)*state2A.P);
    state3.propertyTable('T');                 // Find other unknown properties
    FFR = state3.calcFuelFlowRate(state3,state2A);
    state3.display();                          // Display all properties at State 3

    // 4. Gas Generator Turbine ++++++++++++++++++++++++++++++++++++++++++++++++

    GGT_turbine_actualState state4A("4A",MLS); // Define State 4A (Actual)
    state4A.assign('M',MFR+FFR);               // Assign Known Properties at State 4A
    state4A.assign('P',P4A);
    state4A.calcEnthalpy(state4A,state3,state1,state2A);
    state4A.propertyTable('H');                // Find other unknown properties
    state4A.display();                         // Display all properties at State 4A

    GGT_turbine_isenState state4S("4S",MLS);// Define State 4S (Isentropic)
    state4S.assign('M',MFR+FFR);               // Assign Known Properties at State 4S
    state4S.assign('P',P4A);
    state4S.calcRelativePressure(state4S,state3);
    state4S.propertyTable('R');                // Find other unknown properties
    state4S.display();                         // Display all properties at State 4S

    // 5. Power Turbine ++++++++++++++++++++++++++++++++++++++++++++++++++++++++

    PWR_turbine_actualState state5A("5A");     // Define State 5A (Actual)
    state5A.assign('M',MFR+FFR);               // Assign Known Properties at State 5A
    state5A.assign('T',T5A);
    state5A.assign('P',P5A);
    state5A.propertyTable('T');                // Find other unknown properties
```

```
    state5A.display();                          // Display all properties at State 5A

    PWR_turbine_isenState state5S("5S");        // Define State 5S (Isentropic)
    state5S.assign('M',MFR+FFR);                // Assign Known Properties at State 5S
    state5S.assign('P',P5A);
    state5S.calcRelativePressure(state5S, state4A);
    state5S.propertyTable('R');                 // Find other unknown properties
    state5S.display();                          // Display all properties at State 5S

    // 6. NET Performance Evaluation ++++++++++++++++++++++++++++++++++++++++++++

    state2S.calcPerformance(state1,state2A,state2S);     // Compressor Performance
    state3.calcPerformance(state3,state2A);              // Combustor Performance
    state4S.calcPerformance(state4A,state4S,state3);     // GGT Turbine Performance
    state5S.calcPerformance(state5A,state5S,state4A);    // PWR Turbine Performance
    calcNETPerformance(state5S,state3,MFR);              // NET System Performance

    // End of Program
    // June Kwon
}
```

- **Next, once the engine was modelled using C++ as shown above, the model was then simulated with the "standard working condition", and the following engine performance-result for <u>each state</u> and for <u>each component</u> was obtained as shown below.**



```
>> ./main

-----------------[State 1]-----------------
Part Name:          Compressor
State #:            1            (Actual)
Temperature:        560          (R)
Pressure:           14.7         (psia)
Relative Pressure:  1.5781
Enthalpy:           3.9          (Btu/lbm)
Mass Flow Rate:     50           (lbm/sec)

-----------------[State 2A]-----------------
Part Name:          Compressor
State #:            2A           (Actual)
Temperature:        1450         (R)
Pressure:           320.592      (psia)
Relative Pressure:  49.188
Enthalpy:           226.2        (Btu/lbm)
Mass Flow Rate:     50           (lbm/sec)

-----------------[State 2S]-----------------
Part Name:          Compressor
State #:            2S           (Isentropic)
Temperature:        1320         (R)
Pressure:           320.592      (psia)
Relative Pressure:  34.4168
Enthalpy:           192.3        (Btu/lbm)
Mass Flow Rate:     50           (lbm/sec)

-----------------[State 3]-----------------
Part Name:          Combustor
State #:            3            (Actual)
Temperature:        2800         (R)
Pressure:           288.533      (psia)
Relative Pressure:  702.2
Enthalpy:           602.5        (Btu/lbm)
Mass Flow Rate:     51.0173      (lbm/sec)

-----------------[State 4A]-----------------
Part Name:          Gas Generator Turbine
State #:            4A           (Actual)
Temperature:        1960         (R)
Pressure:           51.31        (psia)
Relative Pressure:  160.84
```

```
Enthalpy:             364.001    (Btu/lbm)
Mass Flow Rate:       51.0173    (lbm/sec)

-----------------[State 4S]-----------------
Part Name:            Gas Generator Turbine
State #:              4S         (Isentropic)
Temperature:          1840       (R)
Pressure:             51.31      (psia)
Relative Pressure:    124.873
Enthalpy:             331        (Btu/lbm)
Mass Flow Rate:       51.0173    (lbm/sec)

-----------------[State 5A]-----------------
Part Name:            Power Turbine
State #:              5A         (Actual)
Temperature:          1500       (R)
Pressure:             15.28      (psia)
Relative Pressure:    56.03
Enthalpy:             239.4      (Btu/lbm)
Mass Flow Rate:       51.0173    (lbm/sec)

-----------------[State 5S]-----------------
Part Name:            Power Turbine
State #:              5S         (Isentropic)
Temperature:          1440       (R)
Pressure:             15.28      (psia)
Relative Pressure:    47.8978
Enthalpy:             223.6      (Btu/lbm)
Mass Flow Rate:       51.0173    (lbm/sec)

------------[Performance of Compressor]------------
Efficiency of Compressor:       84.7503 (%)
Work Done on Compressor:        222.3   (Btu/lbm)
Power Required by Compressor:   11115   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:  51.0173 (lbm/sec)
Heat Added to Combustor:        376.3   (Btu/lbm)
Heat Rate Added to Combustor:   19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:      87.8451 (%)
Work Done by GGT Turbine:       238.499 (Btu/lbm)
Power Generated by GGT Turbine: 12167.6 (Btu/sec)
```

```
-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:      88.7465 (%)
Work Done by PWR Turbine:       124.601 (Btu/lbm)
Power Generated by PWR Turbine: 6356.78 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System: 32.7198 (%)
Work Done by Net Cycle System:  124.601 (Btu/lbm)
Power Generated by Net System:  6356.78 (Btu/sec)
Fuel/Air Mass Flow Ratio:       0.0203455
```

- **Thus, as shown above, the computational model successfully simulated & generated the same performance-result data as the manual analysis. Now, this model will allow me to run multiple simulations and analyze the system at a much faster pace than manual analysis.**

- **Using this model, the method to increase the net-system efficiency will be explored (in a relatively simple manner).**

- **Currently, the net-system efficiency is 32.7198%.**

- **To explore the ways to increase this efficiency, several tests have been run with "different working conditions" to observe and evaluate the engine performance and how it compares to the "standard working condition" operation.**

  1. **Decrease the inlet temperature of the compressor by 45R.**
     $(\mathrm{T}_1 = 560\mathrm{R} \rightarrow \mathrm{T}_1 = 515\mathrm{R}\ )$

  2. **Increase the inlet temperature of the compressor by 45R.**
     $(\mathrm{T}_1 = 560\mathrm{R} \rightarrow \mathrm{T}_1 = 605\mathrm{R}\ )$

3. Decrease the outlet temperature of the power turbine by 45R.
$(\mathrm{T_{5A}} = 1500\mathrm{R} \rightarrow \mathrm{T_{5A}} = 1455\mathrm{R}\ )$

4. Increase the outlet temperature of the power turbine by 45R.
$(\mathrm{T_{5A}} = 1500\mathrm{R} \rightarrow \mathrm{T_{5A}} = 1545\mathrm{R}\ )$

5. Decrease the outlet pressure of the gas-generator turbine by 10%.
$(P_{4A} = 51.31\, psia \rightarrow P_{4A} = 46.179\, psia)$

6. Increase the outlet pressure of the gas-generator turbine by 10%.
$(P_{4A} = 51.31\, psia \rightarrow P_{4A} = 56.441\, psia)$

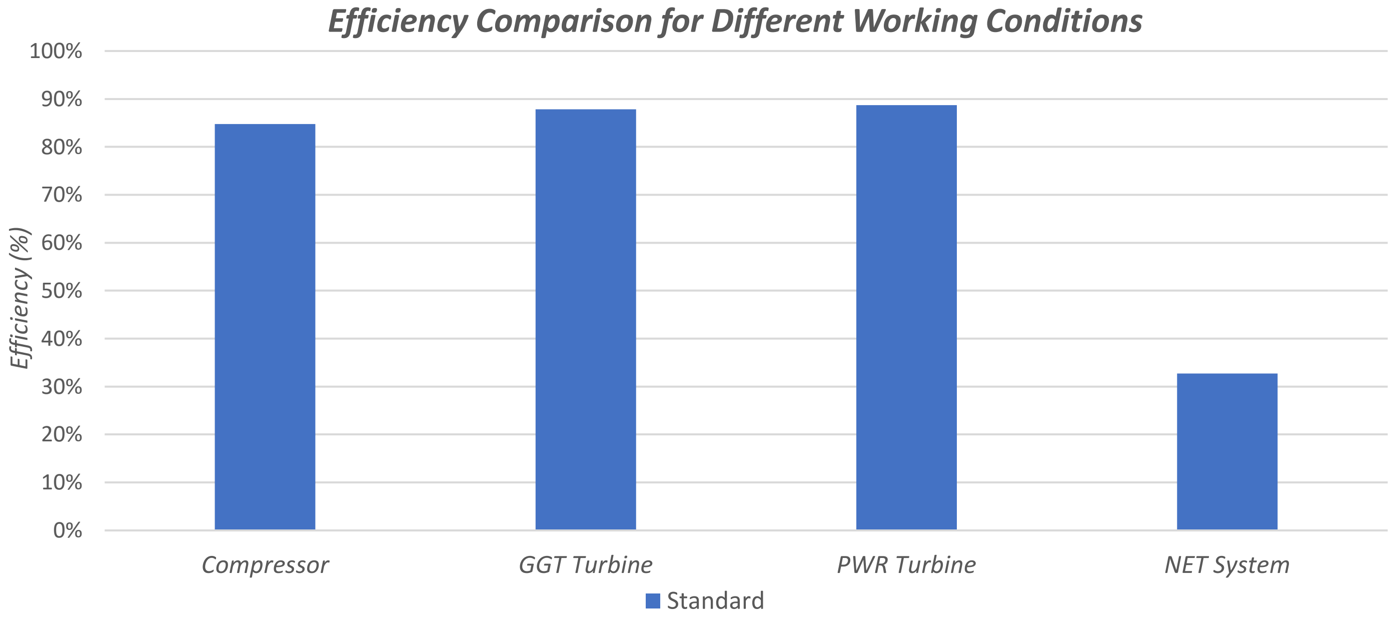
Then, from the above 6 tests, the behavior & performance of the engine will be analyzed, and the combinations of the above different working conditions will be chosen to further increase the net-system efficiency.

[Standard Working Condition]

- First, the model was run with the standard working condition.
- The performance-result is shown on the right, and the efficiencies of each component (Compressor, GGT Turbine, PWR Turbine) and the net-system efficiency are plotted below.
- The efficiency equations for each component are also provided below to indicate the changes in the efficiency equations when the different working conditions are used.

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}\ , \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}\ , \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}\ , \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$



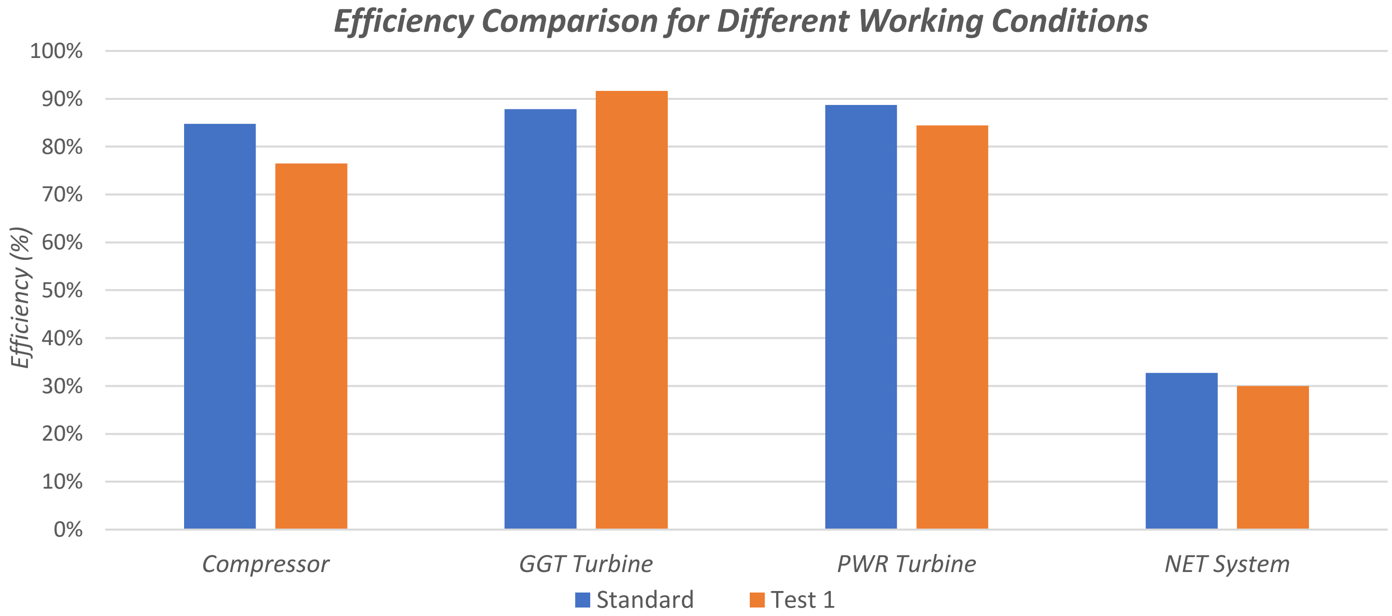

```
------------[Performance of Compressor]------------
Efficiency of Compressor:        84.7503 (%)
Work Done on Compressor:         222.3   (Btu/lbm)
Power Required by Compressor:    11115   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:   51.0173 (lbm/sec)
Heat Added to Combustor:         376.3   (Btu/lbm)
Heat Rate Added to Combustor:    19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:       87.8451 (%)
Work Done by GGT Turbine:        238.499 (Btu/lbm)
Power Generated by GGT Turbine:  12167.6 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:       88.7465 (%)
Work Done by PWR Turbine:        124.601 (Btu/lbm)
Power Generated by PWR Turbine:  6356.78 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System:  32.7198 (%)
Work Done by Net Cycle System:   124.601 (Btu/lbm)
Power Generated by Net System:   6356.78 (Btu/sec)
Fuel/Air Mass Flow Ratio:        0.0203455
```

[Test 1: Decrease Compressor Inlet Temperature]

- As shown below, the efficiencies of the compressor, the PWR turbine, and the net-system decreased, whereas the efficiency of the GGT turbine increased.
- This makes sense because the compressor needs to require more power to compensate for the decreased inlet temperature. This requires the GGT turbine to generate more power. When the GGT turbine uses more energy, leaving its outlet air with lower pressure and lower temperature, then, the PWR turbine will have lower energy to generate the power, losing its efficiency.

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}, \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}, \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}, \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$



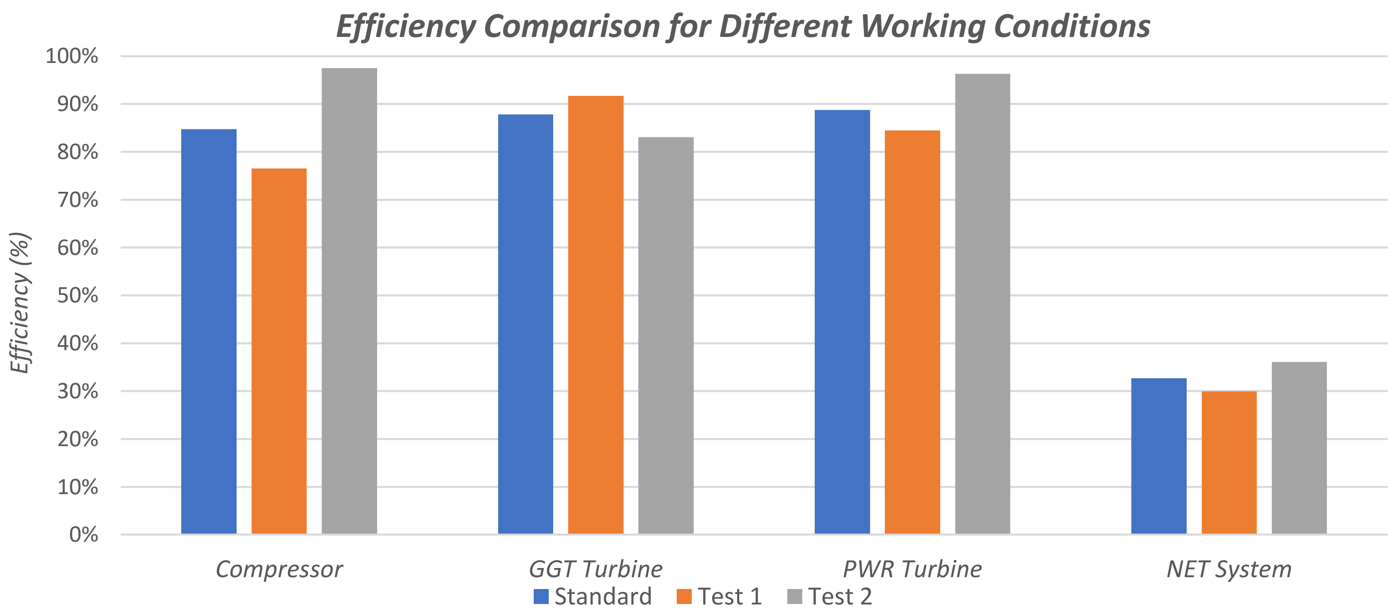

```
------------[Performance of Compressor]------------
Efficiency of Compressor:         76.5086 (%)
Work Done on Compressor:          232     (Btu/lbm)
Power Required by Compressor:     11600   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:    51.0173 (lbm/sec)
Heat Added to Combustor:          376.3   (Btu/lbm)
Heat Rate Added to Combustor:     19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:        91.6782 (%)
Work Done by GGT Turbine:         248.906 (Btu/lbm)
Power Generated by GGT Turbine:   12698.5 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:        84.4667 (%)
Work Done by PWR Turbine:         114.194 (Btu/lbm)
Power Generated by PWR Turbine:   5825.85 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System:   29.987  (%)
Work Done by Net Cycle System:    114.194 (Btu/lbm)
Power Generated by Net System:    5825.85 (Btu/sec)
Fuel/Air Mass Flow Ratio:         0.0203455
```

[Test 2: Increase Compressor Inlet Temperature]

- As shown below, the efficiencies of the compressor, the PWR turbine, and the net-system increased, whereas the efficiency of the GGT turbine decreased.
- This makes sense because now the compressor does not need more power because inlet temperature is already increased. This requires the GGT turbine to generate less power. When the GGT turbine uses less energy, leaving its outlet air with higher pressure and higher temperature, then, the PWR turbine will have more energy to generate more power, increasing its efficiency.

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1},\ \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}},\ \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}},\ \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$



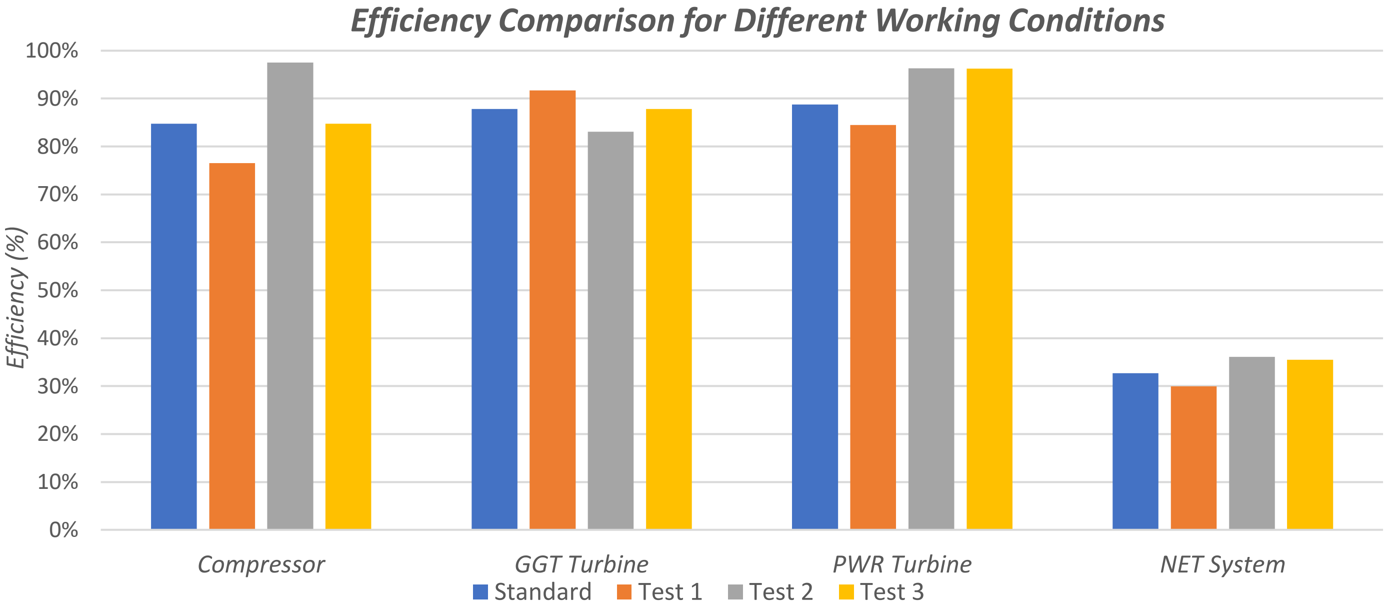

```
------------[Performance of Compressor]------------
Efficiency of Compressor:        97.5273 (%)
Work Done on Compressor:         210.3   (Btu/lbm)
Power Required by Compressor:    10515   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:   51.0173 (lbm/sec)
Heat Added to Combustor:         376.3   (Btu/lbm)
Heat Rate Added to Combustor:    19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:       83.1031 (%)
Work Done by GGT Turbine:        225.625 (Btu/lbm)
Power Generated by GGT Turbine: 11510.8 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:       96.2879 (%)
Work Done by PWR Turbine:        137.475 (Btu/lbm)
Power Generated by PWR Turbine: 7013.6   (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System: 36.1007 (%)
Work Done by Net Cycle System:  137.475 (Btu/lbm)
Power Generated by Net System:  7013.6   (Btu/sec)
Fuel/Air Mass Flow Ratio:        0.0203455
```

[Test 3: Decrease PWR Turbine Outlet Temperature]

- As shown below, the efficiencies of the compressor and the GGT turbine remained the same. However, the efficiencies of the PWR turbine and the net-system increased.
- This makes sense because decreasing the PWR turbine outlet temperature means that the turbine is extracting more energy from the air, leaving its outlet air with lower temperature. More energy extraction means the higher efficiency for the turbine, and the higher efficiency in the PWR turbine means increase in the net-system efficiency.

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}, \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}, \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}, \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$



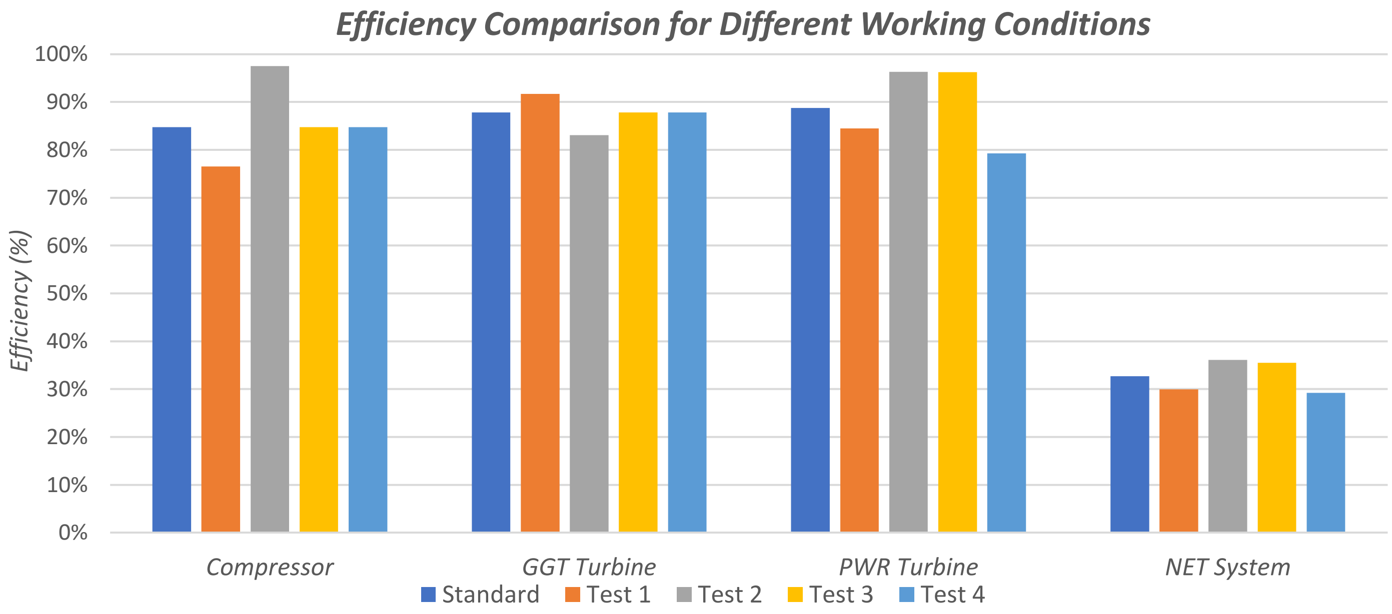

```
------------[Performance of Compressor]------------
Efficiency of Compressor:        84.7503 (%)
Work Done on Compressor:         222.3   (Btu/lbm)
Power Required by Compressor:    11115   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:   51.0173 (lbm/sec)
Heat Added to Combustor:         376.3   (Btu/lbm)
Heat Rate Added to Combustor:    19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:       87.8451 (%)
Work Done by GGT Turbine:        238.499 (Btu/lbm)
Power Generated by GGT Turbine:  12167.6 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:       96.2251 (%)
Work Done by PWR Turbine:        135.101 (Btu/lbm)
Power Generated by PWR Turbine:  6892.46 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System:  35.4771 (%)
Work Done by Net Cycle System:   135.101 (Btu/lbm)
Power Generated by Net System:   6892.46 (Btu/sec)
Fuel/Air Mass Flow Ratio:        0.0203455
```

[Test 4: Increase PWR Turbine Outlet Temperature]

- As shown below, the efficiencies of the compressor and the GGT turbine remained the same. However, the efficiencies of the PWR turbine and the net-system decreased.
- This makes sense because increasing the PWR turbine outlet temperature means that the turbine is extracting less energy from the air, leaving its outlet air with higher temperature. Less energy extraction means the lower efficiency for the turbine, and the lower efficiency in the PWR turbine means decrease in the net-system efficiency.

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}, \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}, \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}, \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$



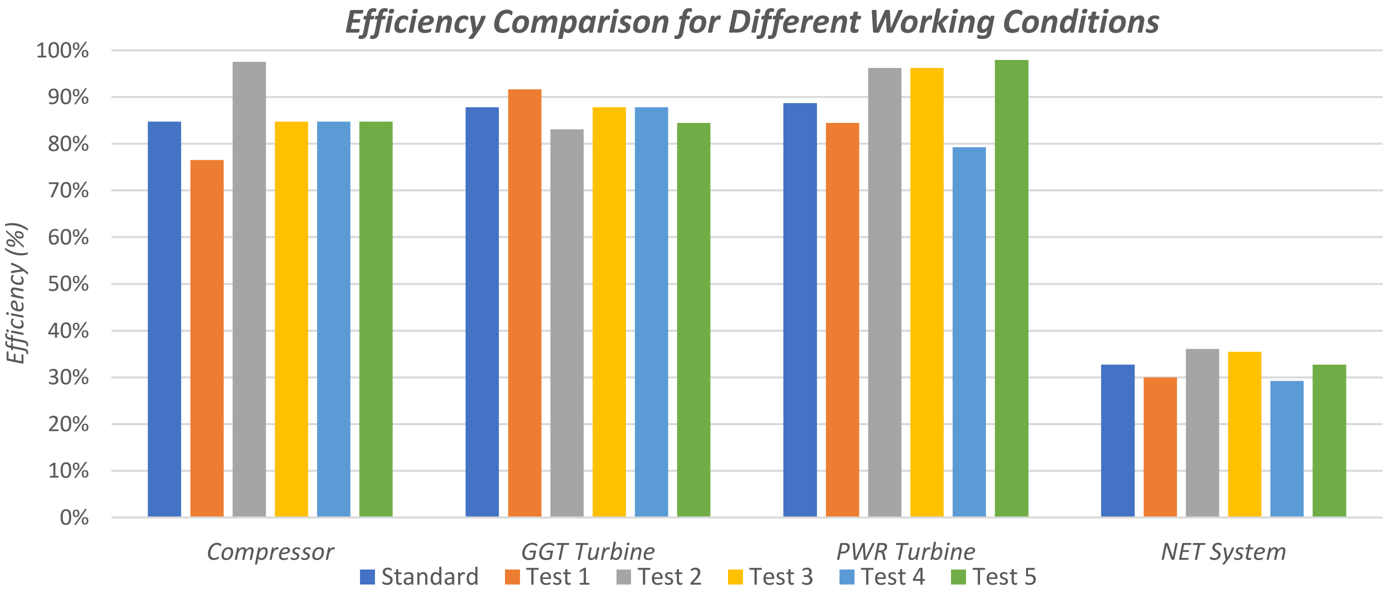

```
------------[Performance of Compressor]------------
Efficiency of Compressor:        84.7503 (%)
Work Done on Compressor:         222.3   (Btu/lbm)
Power Required by Compressor:    11115   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:   51.0173 (lbm/sec)
Heat Added to Combustor:         376.3   (Btu/lbm)
Heat Rate Added to Combustor:    19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:       87.8451 (%)
Work Done by GGT Turbine:        238.499 (Btu/lbm)
Power Generated by GGT Turbine:  12167.6 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:       79.2736 (%)
Work Done by PWR Turbine:        111.301 (Btu/lbm)
Power Generated by PWR Turbine:  5678.25 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System:  29.2273 (%)
Work Done by Net Cycle System:   111.301 (Btu/lbm)
Power Generated by Net System:   5678.25 (Btu/sec)
Fuel/Air Mass Flow Ratio:        0.0203455
```

[Test 5: Decrease GGT Turbine Outlet Pressure]

- As shown below, the efficiencies of the compressor and the net-system remained the same. However, the GGT turbine efficiency decreased, whereas the PWR turbine efficiency increased.
- For this test, as shown in below equation, decreasing the GGT turbine outlet pressure does not mean the turbine will extract more energy. Rather, it means that the actual process remains the same, but the isentropic process changes its energy margin without changing the net-system efficiency. For this test, a pressure-decrease in GGT turbine will increase the isentropic process energy margin in GGT turbine (thus, decreasing its efficiency) and decrease the isentropic process energy margin in PWR turbine (thus, increasing its efficiency).

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}, \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}, \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}, \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$



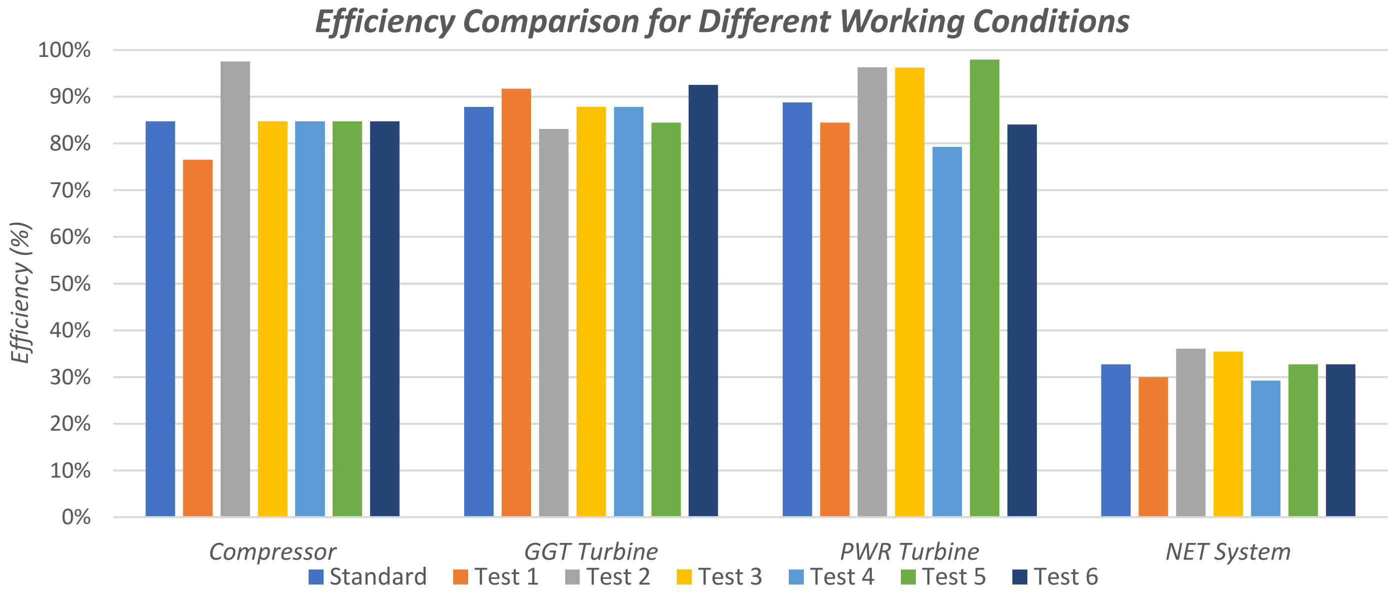

```
------------[Performance of Compressor]------------
Efficiency of Compressor:        84.7503 (%)
Work Done on Compressor:         222.3   (Btu/lbm)
Power Required by Compressor:    11115   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:   51.0173 (lbm/sec)
Heat Added to Combustor:         376.3   (Btu/lbm)
Heat Rate Added to Combustor:    19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:       84.4545 (%)
Work Done by GGT Turbine:        238.499 (Btu/lbm)
Power Generated by GGT Turbine: 12167.6 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:       97.956  (%)
Work Done by PWR Turbine:        124.601 (Btu/lbm)
Power Generated by PWR Turbine: 6356.78 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System: 32.7198 (%)
Work Done by Net Cycle System:   124.601 (Btu/lbm)
Power Generated by Net System:   6356.78 (Btu/sec)
Fuel/Air Mass Flow Ratio:        0.0203455
```

[Test 6: Increase GGT Turbine Outlet Pressure]

- As shown below, the efficiencies of the compressor and the net-system remained the same. However, the GGT turbine efficiency increased, whereas the PWR turbine efficiency decreased.
- For this test, as shown in below equation, increasing the GGT turbine outlet pressure does not mean the turbine will extract less energy. Rather, it means that the actual process remains the same, but the isentropic process changes its energy margin without changing the net-system efficiency. For this test, a pressure-increase in GGT turbine will decrease the isentropic process energy margin in GGT turbine (thus, increasing its efficiency) and increase the isentropic process energy margin in PWR turbine (thus, decreasing its efficiency).

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}, \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}, \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}, \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$

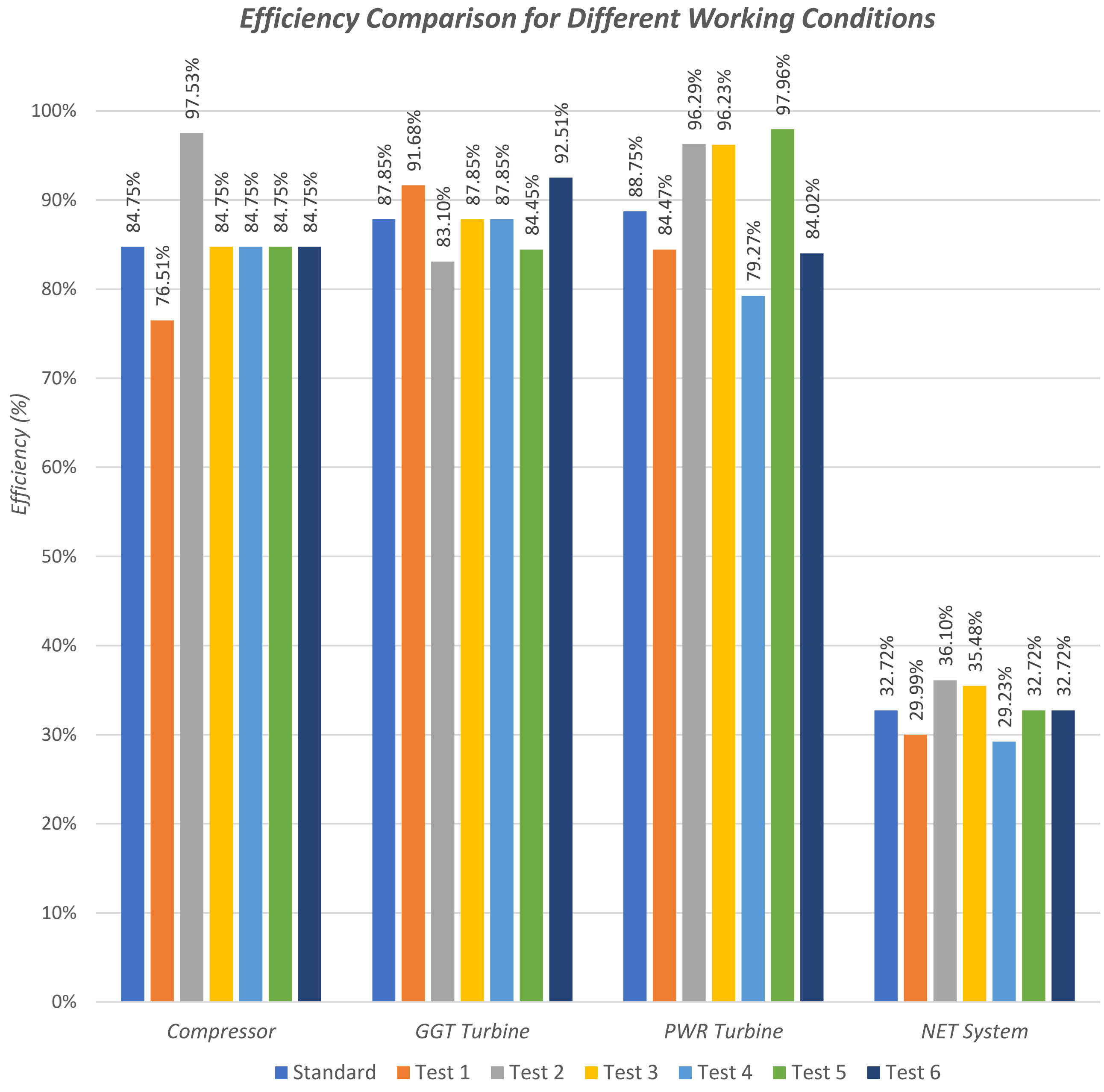



```
------------[Performance of Compressor]------------
Efficiency of Compressor:        84.7503 (%)
Work Done on Compressor:         222.3   (Btu/lbm)
Power Required by Compressor:    11115   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:   51.0173 (lbm/sec)
Heat Added to Combustor:         376.3   (Btu/lbm)
Heat Rate Added to Combustor:    19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:       92.5134 (%)
Work Done by GGT Turbine:        238.499 (Btu/lbm)
Power Generated by GGT Turbine:  12167.6 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:       84.0189 (%)
Work Done by PWR Turbine:        124.601 (Btu/lbm)
Power Generated by PWR Turbine:  6356.78 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System:  32.7198 (%)
Work Done by Net Cycle System:   124.601 (Btu/lbm)
Power Generated by Net System:   6356.78 (Btu/sec)
Fuel/Air Mass Flow Ratio:        0.0203455
```

- Thus, comparing all the different working conditions (Test 1 – 6) to the standard working condition, the following observations can be noted.

- It can be shown that **Test 2** and **Test 3** increase the net-system efficiency.
  - Test 2 (Increase Compressor Inlet Temperature) increases the net-system efficiency up to 36.10%
  - Test 3 (Decrease PWR Turbine Outlet Temperature) increases the net-system efficiency up to 35.48%

- **Test 5** and **Test 6** show that the pressure change in the GGT turbine outlet balances the efficiencies of the GGT turbine and the PWR turbine.
  - A pressure-decrease in GGT turbine (Test 5) decreases the GGT turbine efficiency and increases the PWR turbine efficiency.
  - A pressure-increase in GGT turbine (Test 6) increases the GGT turbine efficiency and decreases the PWR turbine efficiency.

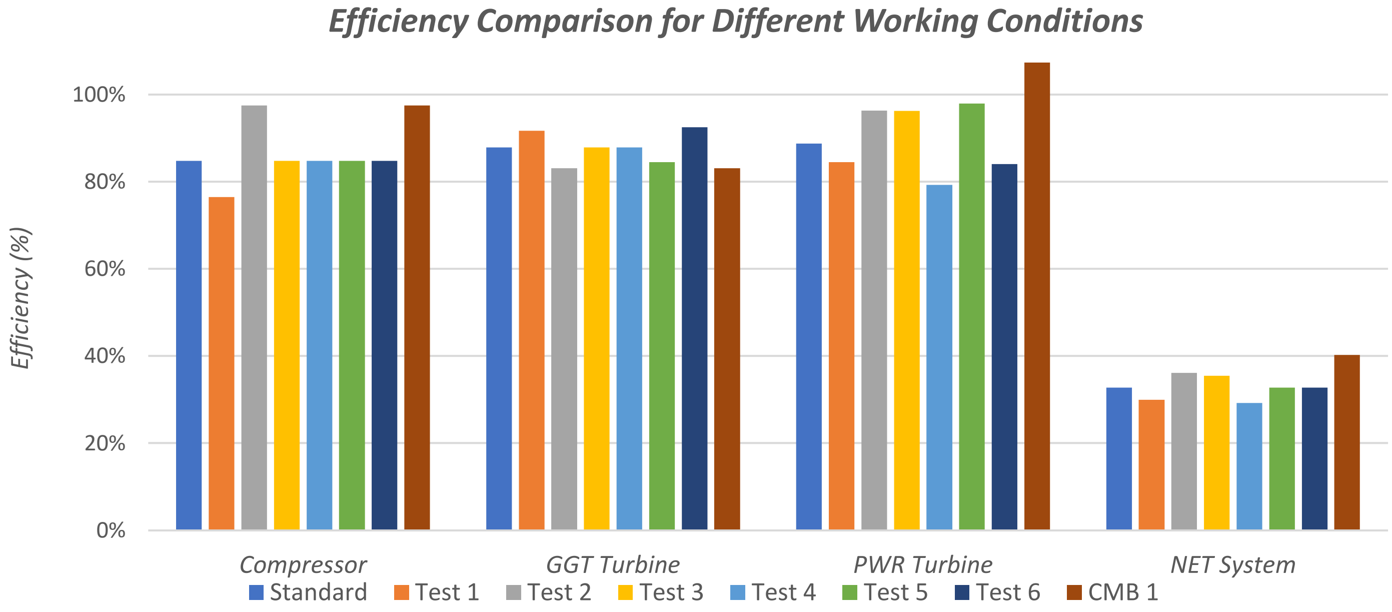


[CMB 1: Increase Compressor Inlet Temperature & Decrease PWR Turbine Outlet Temperature]

- As written above, the combination-test of Test 2 & Test 3 was designed and run. The new condition for the test can be shown below.
  - The compressor inlet temperature is increased by 50R $(T_1 = 560R \rightarrow T_1 = 610R)$
  - The PWR turbine outlet temperature is decreased by 65R $(T_{5A} = 1500R \rightarrow T_{5A} = 1435R)$
- The efficiencies of the compressor, the PWR turbine, and the net-system <u>increased</u>, and the efficiency of the GGT turbine <u>decreased</u>.
- However, the efficiency of PWR turbine is 107.354%. Such efficiency does not exist. Also, the efficiency difference between GGT turbine and PWR turbine is too great, this difference needs to be balanced.

*Blue: Decreased
*Red: Increased

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}, \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}, \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}, \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$




```
------------[Performance of Compressor]------------
Efficiency of Compressor:       97.5273 (%)
Work Done on Compressor:        210.3   (Btu/lbm)
Power Required by Compressor:   10515   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:  51.0173 (lbm/sec)
Heat Added to Combustor:        376.3   (Btu/lbm)
Heat Rate Added to Combustor:   19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:      83.1031 (%)
Work Done by GGT Turbine:       225.625 (Btu/lbm)
Power Generated by GGT Turbine: 11510.8 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:      107.354 (%)
Work Done by PWR Turbine:       153.275 (Btu/lbm)
Power Generated by PWR Turbine: 7819.67 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System: 40.2497 (%)
Work Done by Net Cycle System:  153.275 (Btu/lbm)
Power Generated by Net System:  7819.67 (Btu/sec)
Fuel/Air Mass Flow Ratio:       0.0203455
```

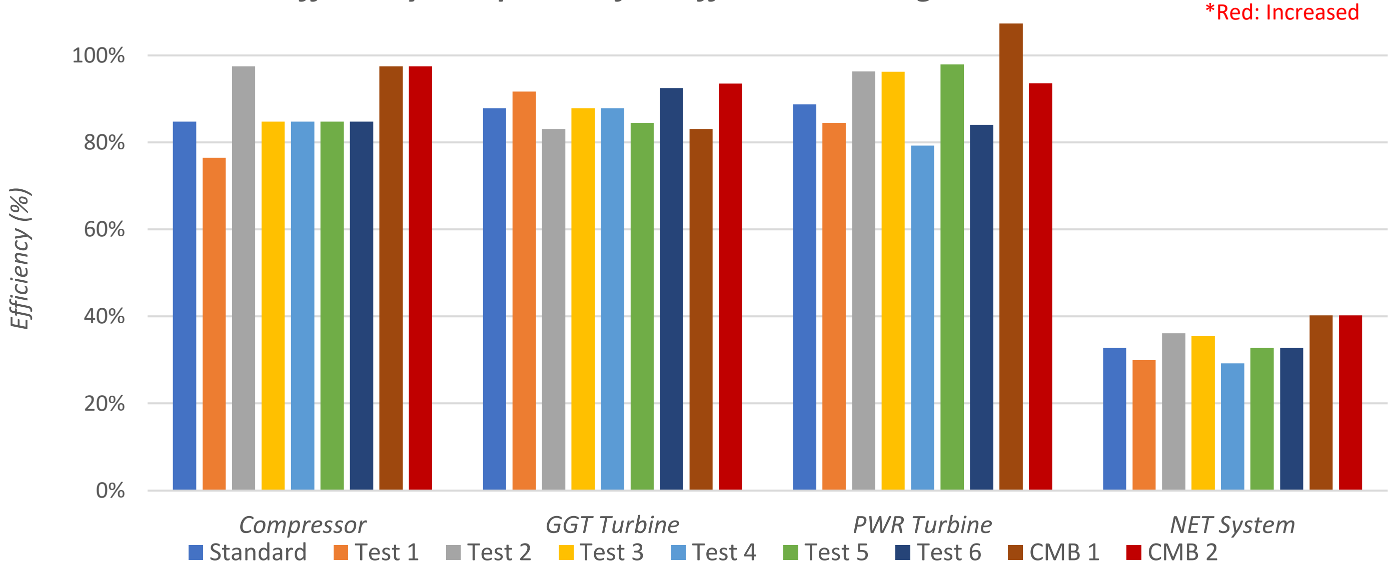
[CMB 2: Increase Compressor Inlet Temperature & Decrease PWR Turbine Outlet Temperature & Increase GGT Turbine Outlet Pressure]

- Thus, a new combination-test of Test 2 & Test 3 & & Test 6 was designed and run. The new condition for the test can be shown below.
  - The compressor inlet temperature is increased by 50R $(T_1 = 560R \rightarrow T_1 = 610R)$
  - The PWR turbine outlet temperature is decreased by 65R $(T_{5A} = 1500R \rightarrow T_{5A} = 1435R)$
  - The GGT turbine outlet pressure is increased by 25% $(P_{4A} = 51.31\, psia \rightarrow P_{4A} = 64.14\, psia)$
- Successfully, the efficiencies of the compressor, the GGT turbine, the PWR turbine, and the net-system all increased. The efficiency difference between GGT turbine and PWR turbine is also well-balanced. **This is a better working condition for the company to use to increase their gas-turbine efficiency.**

$$\eta_c = \frac{h_{2s}-h_1}{h_{2a}-h_1}, \eta_{ggt} = \frac{h_3-h_{4a}}{h_3-h_{4s}}, \eta_{pwr} = \frac{h_{4a}-h_{5a}}{h_{4a}-h_{5s}}, \eta_{net} = \frac{\dot{W}_{pt}}{\dot{Q}_a}$$

Efficiency Comparison for Different Working Conditions

*Blue: Decreased
*Red: Increased

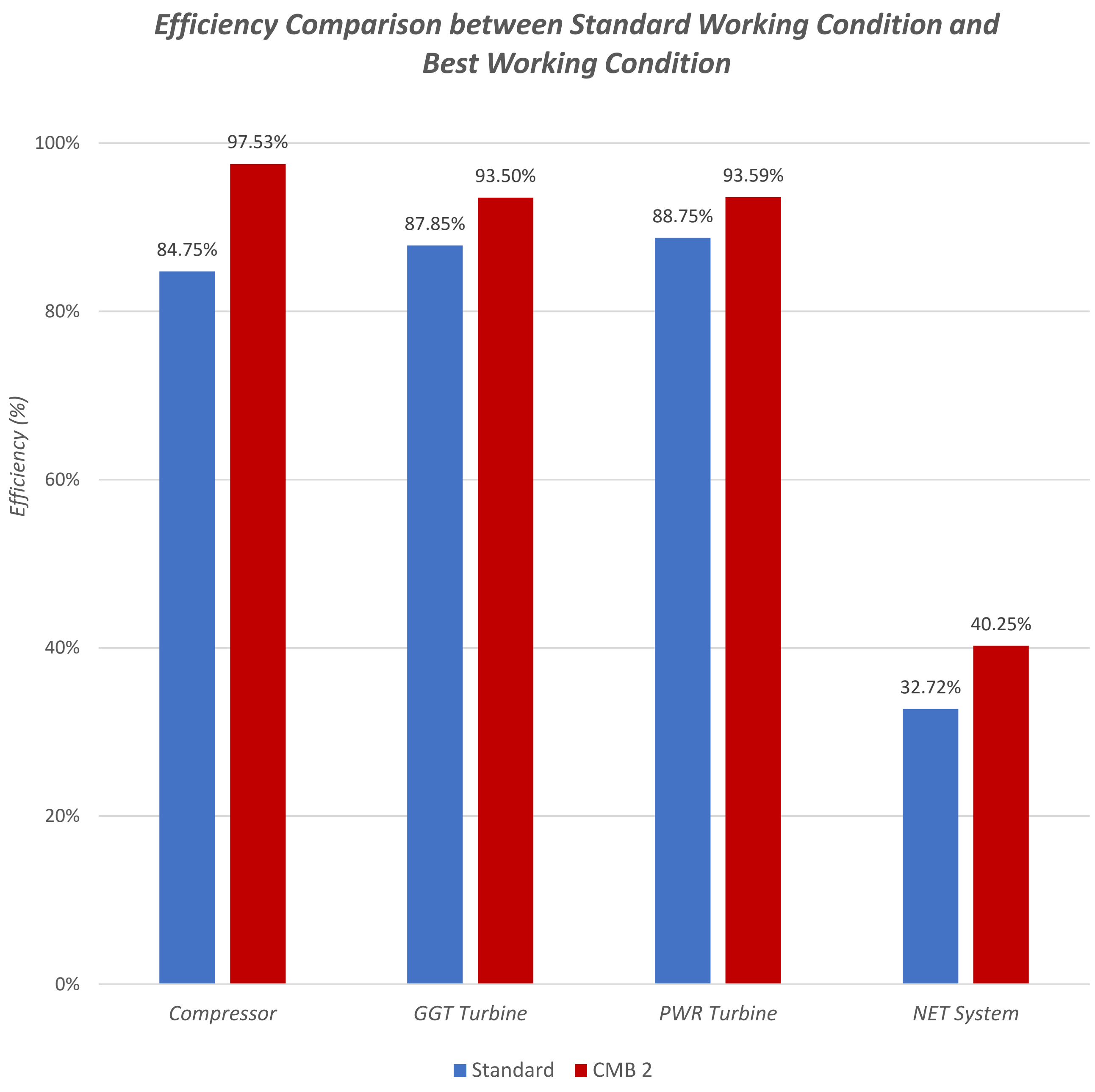



```
------------[Performance of Compressor]------------
Efficiency of Compressor:        97.5273 (%)
Work Done on Compressor:         210.3   (Btu/lbm)
Power Required by Compressor:    10515   (Btu/sec)

-------------[Performance of Combustor]-------------
Fuel Flow Rate into Combustor:   51.0173 (lbm/sec)
Heat Added to Combustor:         376.3   (Btu/lbm)
Heat Rate Added to Combustor:    19427.9 (Btu/sec)

-------[Performance of Gas Generator Turbine]-------
Efficiency of GGT Turbine:       93.5039 (%)
Work Done by GGT Turbine:        225.625 (Btu/lbm)
Power Generated by GGT Turbine:  11510.8 (Btu/sec)

-----------[Performance of Power Turbine]-----------
Efficiency of PWR Turbine:       93.5888 (%)
Work Done by PWR Turbine:        153.275 (Btu/lbm)
Power Generated by PWR Turbine:  7819.67 (Btu/sec)

---[Net Performance of Entire Gas Turbine System]---
Efficiency of Net Cycle System:  40.2497 (%)
Work Done by Net Cycle System:   153.275 (Btu/lbm)
Power Generated by Net System:   7819.67 (Btu/sec)
Fuel/Air Mass Flow Ratio:        0.0203455
```

- Therefore, finally, by analyzing all the different working conditions (Test 1 – 6), the behavior and the performance of the given gas-turbine engine were understood, and the best working condition that increases the engine efficiency was found.

- Best Working Condition:
  - Increase the compressor inlet temperature by 50R. $(T_1 = 560R \ \rightarrow T_1 = 610R)$
  - Decrease the PWR turbine outlet temperature by 65R. $(T_{5A} = 1500R \ \rightarrow T_{5A} = 1435R)$
  - Increase the GGT turbine outlet pressure by 25%. $(P_{4A} = 51.31\ psia \rightarrow P_{4A} = 64.14\ psia)$

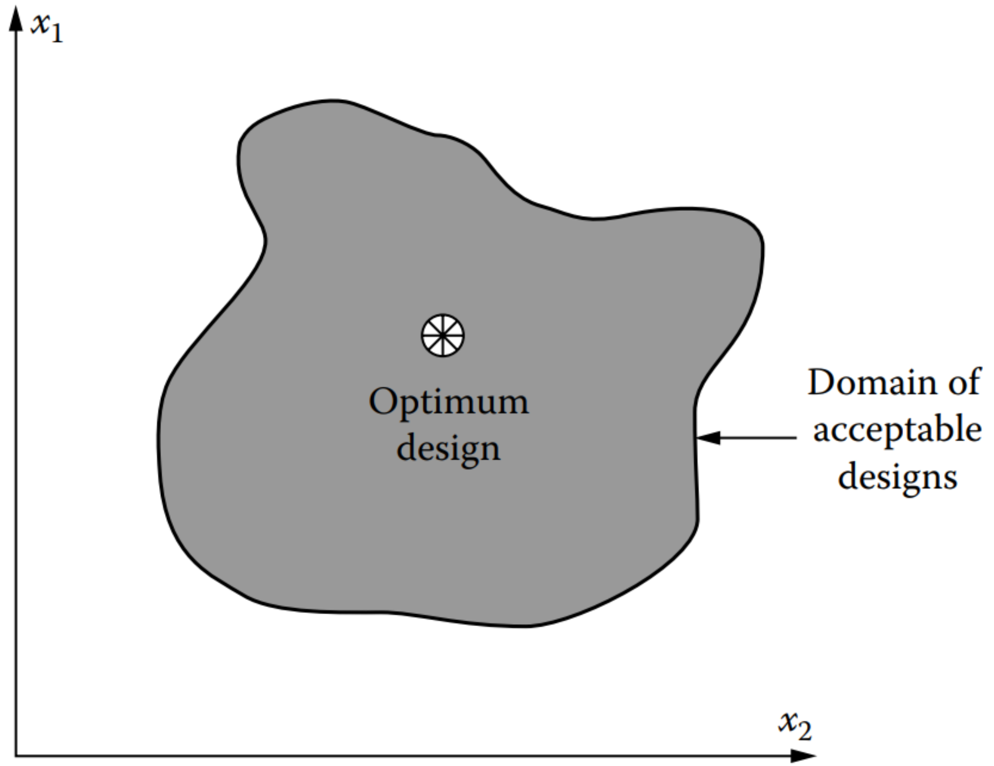
- Using this newly proposed best working condition, the engine can perform better than the standard working condition in all components. The following observations can be noted from the efficiency comparison plot shown right.
  - The compressor efficiency can be increased by 12.78%.
  - The GGT turbine efficiency can be increased by 5.65%.
  - The PWR turbine efficiency can be increased by 4.84%.
  - **The net-system efficiency can be increased by 7.53%.**



Discussion

- The best working condition that increases the net-system efficiency by 7.53% has been found by analyzing and understanding the dynamics of the gas-turbine engine.
- However, this best working condition does not mean the most **optimum** working condition for the given gas-turbine engine. This best working condition is only **locally** best among the given tested working conditions. To find the most optimum working condition, a more sophisticated mathematical analysis needs to be performed such as linear, dynamic, geometric programming, constrained/unconstrained optimization, simplex method, and other various search methods, etc.
- For this problem, many constraints existed. For example, the combustor outlet temperature was constrained to be $T_{3a} = 2800R$. Knowing these constraints, the best working condition was designed to make sure the system lies in the domain of acceptable designs. However, it is my interest to wonder, if these constraints change (For example, $T_{3a} \geq 2800R$), how would the best working condition change? Would it still be in the domain of acceptable designs? How different that best working condition would be compared to the current best working condition?

Image : Google



Discussion

- Moreover, it was stated that the best working condition requires,
  - Increase the compressor inlet temperature by 50R.
  - Decrease the PWR turbine outlet temperature by 65R.
  - Increase the GGT turbine outlet pressure by 25%.
- However, a question must be asked, **how much does it cost to implement the above conditions to the engine? Is the cost less than the benefit? How feasible it is to implement the above conditions?** Before implementing this condition, I highly recommend the company to consider such economic questions and compare if the benefit of this best working condition exceeds the cost.
- Lastly, it is my desire & interest to learn more about various methods to optimize the thermal system like this. This project has taught me a lot about the engine-operation and how to analyze the engine system. It was one little step towards the goal.

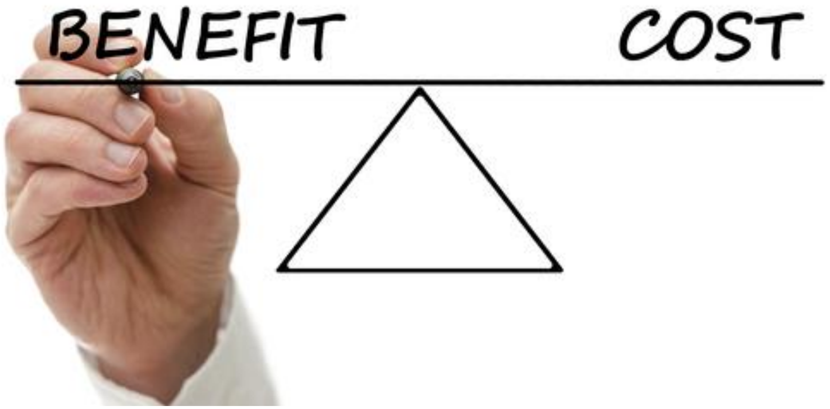


Image : Google

Conclusion

- The best working condition that increases the net-system efficiency was designed by utilizing the computational model of the gas-turbine engine.
- This computational model significantly improved the system-dynamics analysis and reduced the time & cost to design the best working condition for the company's gas-turbine engine.
- This best working condition increases all components' efficiencies, including the compressor, the gas-generator turbine, the power turbine, and the net-system.
- However, an economic analysis, feasibility of implementing the best working condition to the actual engine, and how well the computational model represents the actual system can be further analyzed to enhance the overall analysis for the company.
- It is my interest to learn more about how to better model the thermodynamics/thermal systems like this, and how to better utilize the computer to help model, analyze, and optimize such systems.

Thank you very much.

June K.

Image : Google