Turbojet Dessign Point Simulator

Sergio Fernandez Bertolin

Thursday, June 04, 2015

Turbojet Dessign Point Simulator

Program a numerical simulator to predict turbojet design point performances.

Use any convenient programming language to write a code that is capable of predicting turbojet performances given a set of design parameters and component efficiency targets.

The design parameters will be:

- Flight mach number (M_0 [2]).
- Flight altitude (h [11000 m]) or, equivalently, atmospheric ambient conditions (T_0 [216.5 K] and p_0 [22630]).
- Turbine entry temperature (T_{t4} [1373 K]).
- Compressor temperature ratio ($\pi_c = T_{t3}/T_{t2}$ [2]).

The component efficiency targets are:

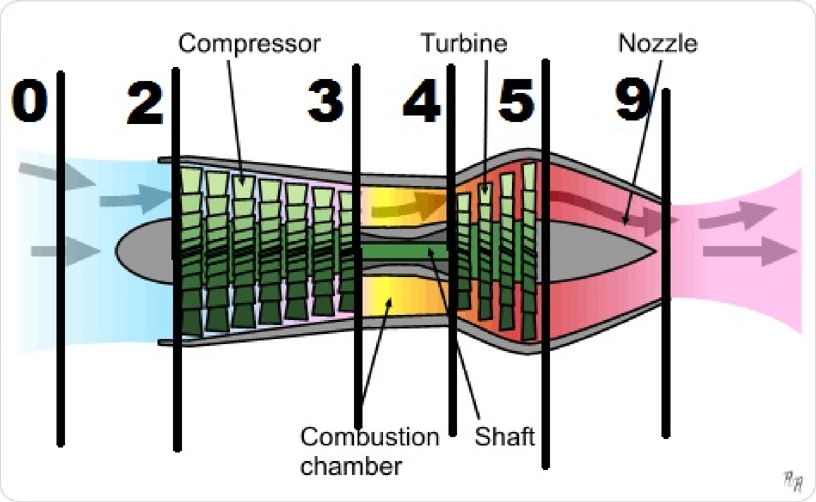
- Intake (ϵ_i [0.075]), burner (ϵ_b [0.06]) and turbine-nozzle duct (ϵ_n [0.02]) pressure loss coefficients.
- Compressor (η_{cp} [0.88]) and turbine (η_{tp} [0.93]) politropic efficiencies.
- Nozzle velocity coefficient (ϕ [0.98])
- Turbine cooling bleed (x [0.07]).

The simulator should be able to compute:

- Specific thrust (Ψ).
- Thermal (η_{th}), propulsive (η_{pr}) and overall (η_0) efficiency.
- Thrust specific fuel consumption (CTS).

The numbers in brackets are reference values to be used for code debugging.

Engine diagram



List of hypotheses made and solver diagram

Stage 0 -> Free-Stream

We assume γ=1.4 and calorically perfect gas to compute the V_0 velocity

```
##First of all, we calculate V_0
V_0 <- M_0*sqrt(gamma*r*T_0)
```

With T_0, our equations and V_0, we now obtain h_0 and h_{t0}

```
##Then, h_0 and ht_0
h_0 <- r*h_air(T_0)
h_t0 <- h_0 + (V_0^2)/2
```

T_{t0}, δ_{t0} and p_{t0} computation

By equalling h_{t0} and the given equations, and using uniroot function (similar to Solver Matlab function) we are obtaining T_{t0}

```
##Now it's time to find Tt_0 by comparing ht_0 computed and
y = function(T){r*h_air(T)-h_t0}
T_t0 <- uniroot(y,interval=c(0,T_t4))$root
tau_c <- (delta_Tc/T_t0)+1
```

Then, computation of δ_{t0} and p_{t0} is made

```
##delta_0 and p_t0 computation
delta_0 <- exp((fi_air(T_t0))-(fi_air(T_0)))
p_t0 <- p_0*delta_0
```

Stage 2 -> Intake

Intake conservation temperature $T_{t2} = T_{t0}$ $h_{t2} = h_{t0}$

```
##T_intake conservation
h_t2 <- h_t0
T_t2 <- T_t0
p_t2 <- p_t0*(1-epsilon_i)
```

Stage 3 -> Compressor

T_{t3},π_c, p_{t2} and p_{t3} computation

Using equations given and η_{cp} (compressor polytropic efficiency) passed as a parameter

```
T_t3 <- T_t2*tau_c
pi_c <- (exp(eta_cp*(fi_air(T_t3)-fi_air(T_t2))))
p_t3 <- pi_c*p_t2
h_t3 <- r*h_air(T_t3)
```

Stage 4 -> Combustion chamber

Then, computation of h_{t3} and h_{t4} is made

For h_t3 computation we only have air. Since fuel is added on stage 4, we have to compute the α parameter to obtain the real enthalpies from now on.

```
##Compute h_t4_air and p_t4
h_t4_air <- r*h_air(T_t4)
hf_Tt4 <- r*hf(T_t4)
p_t4 <- p_t3*(1-epsilon_b)

##Compute alpha, alpha_prima and h_t4
alpha <- (h_t4_air - h_t3)/hf_Tt4
alpha_prima <- (1-x)*alpha
h_t4 <- r*h(T_t4,alpha_prima)
```

Stage 5 -> Turbine

By equalling h_{t5} and the given equations, and using uniroot function we are obtaining T_{t5}

```
##Computing h_t5 and pi_t

h_t5 <- h_t4 - ((h_t3 - h_t2)/((1+alpha)*(1-x)))
y = function(T){r*h(T,alpha_prima)-h_t5}
T_t5 <- uniroot(y,interval=c(0,T_t4))$root
```

π_t and p_{t5} computation

Using equations given and η_{tp} (turbine polytropic efficiency) passed as a parameter (with alpha_prima, considering fuel)

```
pi_t <- (exp((1/eta_tp)*(fi(T_t5,alpha_prima)-fi(T_t4,alpha
p_t5 <- p_t4*pi_t
```

Stage 9 -> Nozzle

We are not considering an afterburner stage

Added to this, we are supposing that our nozzle is adapted, so:

$p_9 = p_0 \;\; \pi_9 = \delta_9$

And we can compute δ_9 from previous values and the given losses

```
##Adapted nozzle (p_9 = p_0)
delta_9 <- (1-epsilon_i)*(1-epsilon_b)*(1-epsilon_n)*delta_
p_t9 <- delta_9*p_0
pi_9 <- delta_9
```

Compute T_{t9} and T_{9is} (First by considering the process isentropic)

First we compute V_9 for an isentropic process inside the nozzle, so: $T_{t9} = T_{t5}$

We are solving the following equation:

$\log(\pi_9) = \Phi_{T\,t9}$ - $\Phi_{T\,9is}$

```
##Compute Tt9 and T9_is
##Computing first as if we have an isentropic process Tt9 =
T_t9_is <- T_t5
fi_T9_is <- fi(T_t9_is,alpha_prima)-log(pi_9)
y = function(T){fi(T,alpha_prima)-fi_T9_is}
T_9_is <- uniroot(y,interval=c(10,T_t4))$root
```

Obtaining V_9 and V_{9is}

We are solving the following equation:

$V_{9is} =$

$$\sqrt{2(h_T t9is - h_T 9is)}$$

```
h_t9_is <- r*h(T_t9_is,alpha_prima)
h_9_is <- r*h(T_9_is,alpha_prima)
V_9_is <- sqrt(2*(h_t9_is-h_9_is))
V_9 <- V_9_is*nozzle_v_coeff
```

Performance Results

Here we have a list of the formulas used

```
Spec_Thrust <- (1+alpha_prima)*V_9 - V_0
eff_thermal <- ((1+alpha_prima)*(V_9^2) - (V_0^2))/(2*alpha
eff_propulsive <- (2*Spec_Thrust*V_0)/((1+alpha_prima)*(V_9
eff_overall <- eff_thermal*eff_propulsive
CTS <- alpha_prima/Spec_Thrust
```

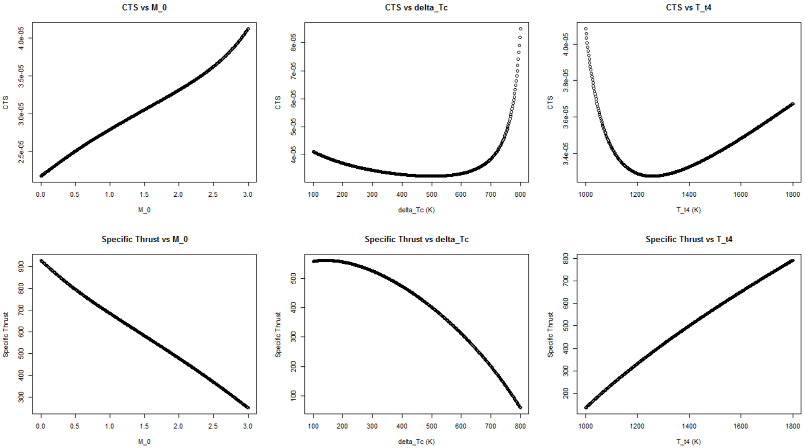
Now we compute those in real-time

Comparison between calculator and hand-made computations

Parameter	Simulator result	Hand-made computation	Desviation
δ_0	7,82	7,82	0,05%
δ_9	15,62	16,05	2,69%
π_c	9,13	8,46	7,94%
π_t	0,26	0,28	9,80%
V_9	1052,29	1018,76	3,29%
η_{th}	0,61	0,58	5,50%
η_{pr}	0,73	0,73	0,75%
η_0	0,44	0,42	4,70%
ψ	478,94	429,15	11,60%
CTS	3,31E-05	3,46E-05	4,42%

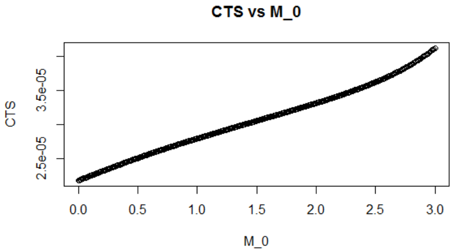
Results

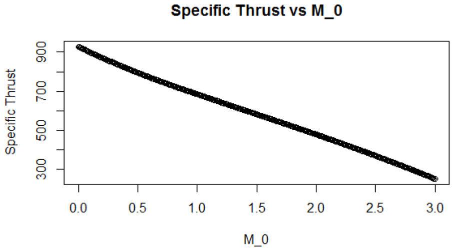
Now we show the plots of the CTS and the Specific Thrust. We have fixed 2 out of the 3 parameters considered.



CTS and Specific Thrust with M_0 variation

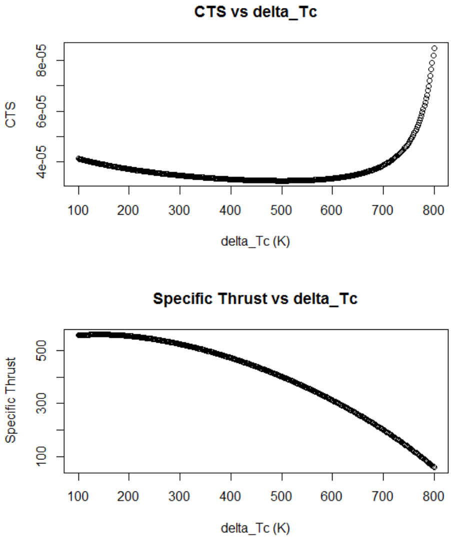
In this case Δ $T_c = 389.697$ K to have $\pi_c = 2$ and $T_{t4} = 1373$ K





CTS and Specific Thrust with Δ T_c variation

In this case $M_0 = 2$ and $T_{t4} = 1373$ K



CTS and Specific Thrust with T_{t4} variation

In this case Δ $T_c = 389.697$ K to have $\pi_c = 2$ and $M_0 = 2$