LPRES Library 1.3.6
Test Cases

School of Aeronautical and Space Engineering

Technical University of Madrid

In cooperation with Empresarios Agrupados

Pablo Sierra Heras

Supervisors:

Juan Manuel Tizón Pulido (Technical University of Madrid),
Javier Vilá Vacas (Empresarios Agrupados),
José Francisco Moral Moral (Empresarios Agrupados).

Contents

1 General overview

1.1 Purpose of this guide

This document is a guide for the engine examples and test cases done with the LPRES Library (version 1.3.6).

This guide will provide the following:

- Description of **some** test cases performed using the LPRES Library.
- Description of **some** engine examples performed using the LPRES Library.
- Comparison of results with the ESPSS toolkit or with measured values.
- Explanation of the particularities of each example or test.
- Enlightenment of the complex part of the component' models.

1.2 Introduction

The LPRES Library, which stands for **Liquid Propellant Rocket Engine Simulation**, contains components to predict the behaviour of the different configurations that a liquid propellant rocket engine can have.

Although the qualities of a thorough and in-depth analysis may be desirable in the study of specific single components and scenarios, the long computational times become insurmountable for complex, multi-component systems.

A good simplification allows the reduction of the 3-D governing partial differential equations to 1-D differential equations which no longer require complex solution methods thus allowing much faster computational times.

As it is a basic library, the components included in it, works in steady state. However, transient models are projected to be included in future developments of it. The library

works with simple models but its usefulness is to achieve the analysis of complex situations due to the system number of elements.

The LPRES Library is built to emulate the elements of the professional ESPSS toolkit. The LPRES Library employs simple models but it can address the simulation of real systems with many elements. The advantages that the LPRES Library presents are that it can be used with the Educational version of EcosimPro and, moreover, its learning curve is much faster than the learning curve of the ESPSS toolkit.

At least version 5.2.0 of EcosimPro must be installed in order to run the library.

All files in which the library is written use version 3.1 (at least) of the MATH Library, which is one of the libraries included with EcosimPro.

1.3 LPRES_EXAMPLES Library

The engine examples and test cases done with the LPRES Library are included in the LPRES_EXAMPLES Library. The examples and test cases were made to show the way to use the LPRES Library.

The test cases are examples in which few components are tested. They show in a clearer way the performances and capabilities of the components included in them, since this is difficult to observe in a most complicated example.

The engine examples show the capabilities of the LPRES Library to simulate complete engines.

To validate the results, a comparison of results with the ESPSS toolkit or with measured values has been made in some examples.

Note that the name of the test cases starts without a capital letter and ends with the word “test”. The name of the engine examples starts with a capital letter.

2 Test Cases

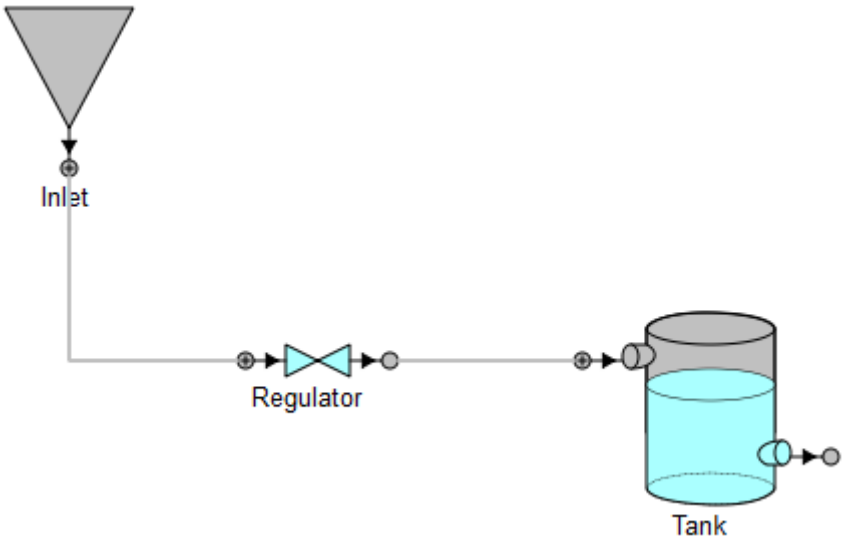
2.1 Regulator and tank design test

Model Name: tank_design_test.eds

Partition Name: default

2.1.1 Model description

This is the simplest example performed with the library in this guide. The model represents a pressurisation circuit of a pressure feed engine. The inlet simulates a gas tank. The regulator, in Type=Known_pt_out mode, imposes the outlet total pressure. Hence, the tank pressure is established by the regulator.



2.1.2 Results

First of all, a design calculation will be performed. First, create a default partition with the tank in Type=Design mode. Then, create a steady experiment and set the outlet tank mass flow to 100 kg/s. In a complete engine, the tank mass flow would be set by the nozzle or another exhaust (see *Appendix A*). This experiment will calculate the pressurisation gas input area of the tank, which will be a known value in the off-design partition.

Note that design tank pressure "p_d" must be lower than the outlet total pressure imposed by the regulator. Otherwise, the convergence will not be reached because the experiment has no physical meaning.

2.2 Regulator and tank off design test

Model Name: tank_offDesign_test.eds

Partition Name: default

2.2.1 Model description

The model is the same as the previous one.

2.2.2 Results

Change the tank mode to Type=Off_design and set "A_g" to the area calculated in the previous experiment. Again, create a default partition and a steady experiment and set the outlet tank mass flow to 100 kg/s..

As the library works in steady state, the inlet conditions can be manually changed in order to evaluate the behaviour of the regulator and the tank as though the gas tank were being emptied. To do this, perform several steady calculations changing the gas tank (Inlet) total pressure. For values above $p_t + \Delta p_{t,min} = 1201500\ Pa$, the tank pressure will be the same as the imposed in the design calculation. $\Delta p_{t,min}$ is the lowest Δp_t value that the regulator requires to work. For lower values, the regulator is completely open. Therefore, the tank pressure is lower than the design pressure because the regulator is not able to impose the desired pressure. In this case, although it is open, the regulator has a pressure drop equal to $\Delta p_{t,min}$.

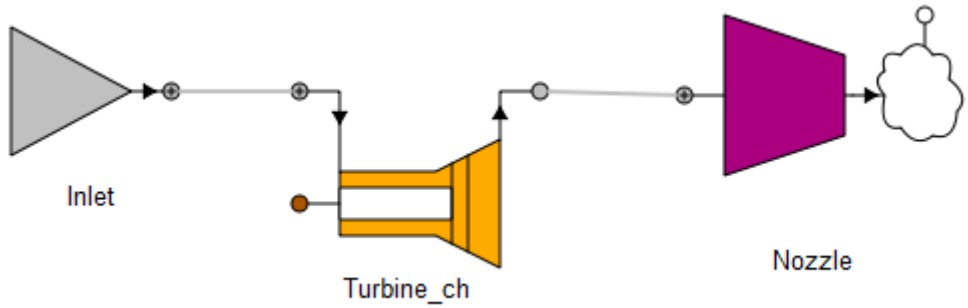
2.3 Choked turbine design test

Model Name: turbine_ch_design_test.eds

Partition Name: default

2.3.1 Model description

The model represents a choked turbine discharging to the atmosphere. The inlet sets the gas inlet conditions.



2.3.2 Results

A design calculation will be performed. To do this, create a default partition with the turbine in Type=Known_pi mode (which is a design mode together with Type=Known_W) and the nozzle in Type=Design mode. Then, create a steady experiment and set the desired power for the turbine. A value of $1940000\ W$ is used in this example. This experiment will calculate the turbine input area, the discharge area and the mass flow to obtain the required power.

The calculation has been performed for two cases of turbine expansion ratio “pi”: 10 and 1.43. An expansion ratio of 10 has been chosen because it is a typical value of a choked turbine. This expansion ratio will not be found in a turbine with a non-choked inlet. Therefore, a lower expansion ratio of 1.43 has been used in order to compare the results of a “Turbine_ch” component and a “Turbine” component.

In design mode, if a pump were connected, it would establish the power by knowing the desired pressure increase in it.

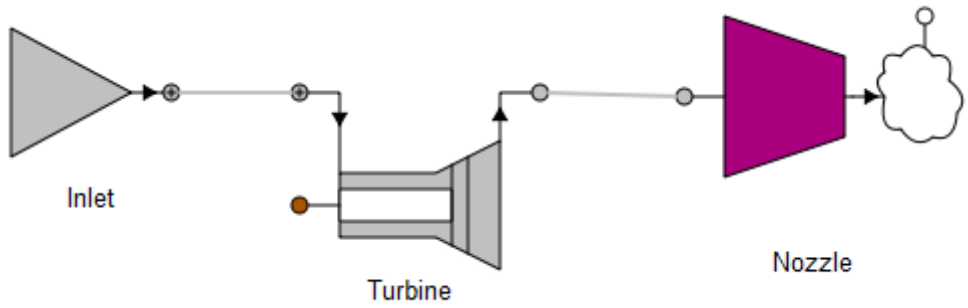
2.4 Turbine design test

Model Name: turbine_design_test.eds

Partition Name: default

2.4.1 Model description

The model represents a turbine discharging to the atmosphere. The inlet sets the gas inlet conditions.



2.4.2 Results

This example is the same as the previous one but the turbine may be choked or not. It will be performed a design calculation. To do this, create a default partition with the turbine in Type=Known_pi mode (which is a design mode together with Type=Known_W) and the nozzle in Type=Design mode. Then create a steady experiment and set the power of the turbine to 1940000 W. This experiment will calculate the turbine input area, the average radius, the discharge area and the mass flow to obtain the power required.

The calculations have been performed for a turbine expansion ratio “pi” of 1.43. For the turbine design Mach number “M”, 0.8, 1 and 1.5 have been used.

Note that the result obtained with Mach 1 is the same as the result obtained in the choked turbine design test.

For Mach 1.5, the component takes the Mach number as if it were 1 instead of 1.5 in the mass flow equation. Hence, the result is also the same as for Mach 1 and the calculation of the average radius is not representative because the performance equation is not being accomplished in that case.

In design mode, if a pump were connected, it would establish the power by knowing the desired pressure increase in it.

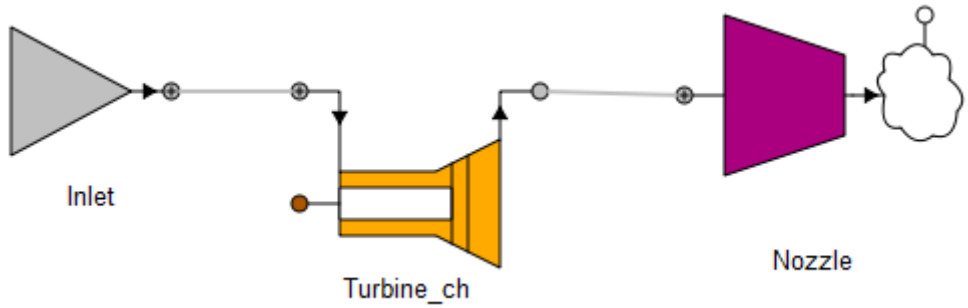
2.5 Choked turbine performance test

Model Name: turbine_ch_performance_test.eds

Partition Name: partition1

2.5.1 Model description

The model represents a choked turbine discharging to the atmosphere. The inlet sets the gas inlet conditions.



2.5.2 Results

A performance calculation will be performed. To do this, create a custom partition with the turbine and the nozzle in Type=Off_design mode.

To calculate the performance map of the turbine, several stationary operating points of the turbine must be calculated. To change the operating point, the discharge area of the nozzle must be changed. Hence, in the partition, transform the discharge area of nozzle "A" to an unknown variable and select it as a boudary. Then create a transient experiment. The transient experiment is used to change the discharge area of the nozzle using a temporary law. It will not really be a transient experiment because the model used for the turbine does not take into account non-stationary effects.

The temporary law is:

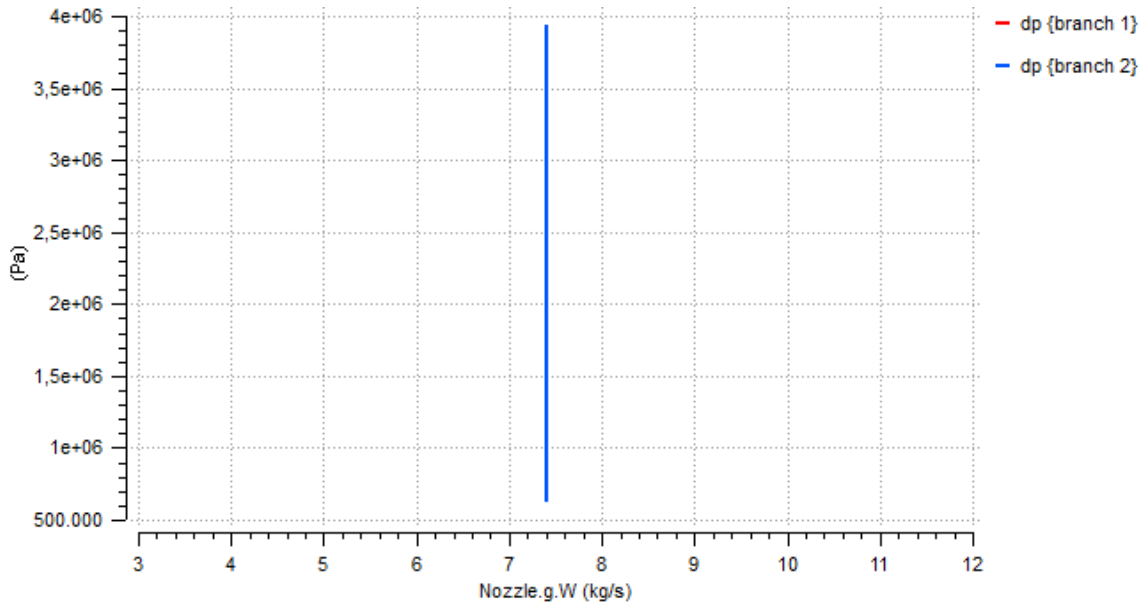
$$A = 0.0018 \cdot (0.8 + \frac{t}{40})$$

The simulation is done from $t = 0$ to $t = 20$.

Note that a discharge area of $0.0018\ m^2$ is the rounded-off discharge area obtained in the turbine design test for pi=1.43. The same happens to the turbine input area, whose rounded-off value is $0.0019\ m^2$. This value has been used for the expansion ratio because the performance map of obtained for component "Turbine_ch" will be

compared to the one obtained for component "Turbine". Otherwise, with higher expansion ratios, it would not be possible to have a non-choked turbine.

Finally, the performance map is:



As can be noted from the above chart, the mass flow does not change because the model states that the turbine is always choked by using the following equation:

$$\dot{m} = A_{in}\cos(\alpha_2)\cdot\frac{p_{t,in}}{\sqrt{R_g T_{t,in}}}\cdot\sqrt{\gamma}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

Component "Turbine_ch" is not affected by the rotational speed. In design mode, it is set as data because the pumps or compressors connected to it will need it. In off-design mode, the pumps or compressors connected to it will calculate the rotational speed. Hence, it is imposed as a boundary in the experiment because the turbine does not have a pump or a compressor connected to it. Anyway, the value given is not important as can be seen in the above chart, where two calculations have been performed with a rotational speed of 6000 rad/s and 0 rad/s and the results have been identical.

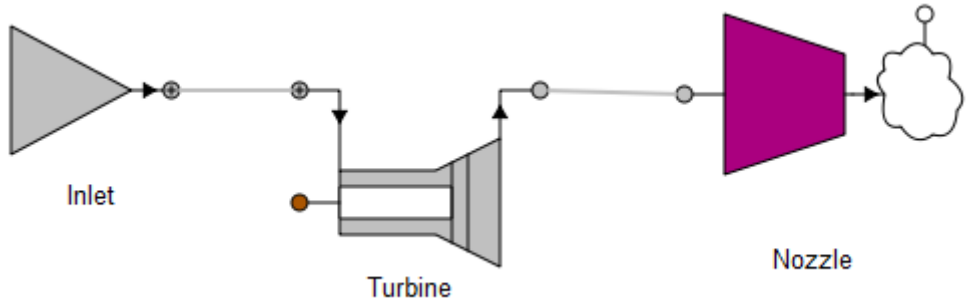
2.6 Turbine performance test

Model Name: turbine_performance_test.eds

Partition Name: partition1

2.6.1 Model description

The model represents a turbine discharging to the atmosphere. The inlet sets the gas inlet conditions.



2.6.2 Results

This example is the same as the previous one but the turbine may be choked or not. A performance calculation will be performed. To do this, create custom partition with the turbine and the nozzle in Type=Off_design mode.

To calculate the performance map of the turbine, several stationary operating points of the turbine must be calculated. To change the operating point, the discharge area of the nozzle must be changed. Hence, in the partition, transform the discharge area of nozzle "A" to an unknown variable and select it as a boundary. Then, create a transient experiment. The transient experiment is used to change the discharge area of the nozzle using a temporary law. It will not really be a transient experiment because the model used for the turbine does not take into account non-stationary effects.

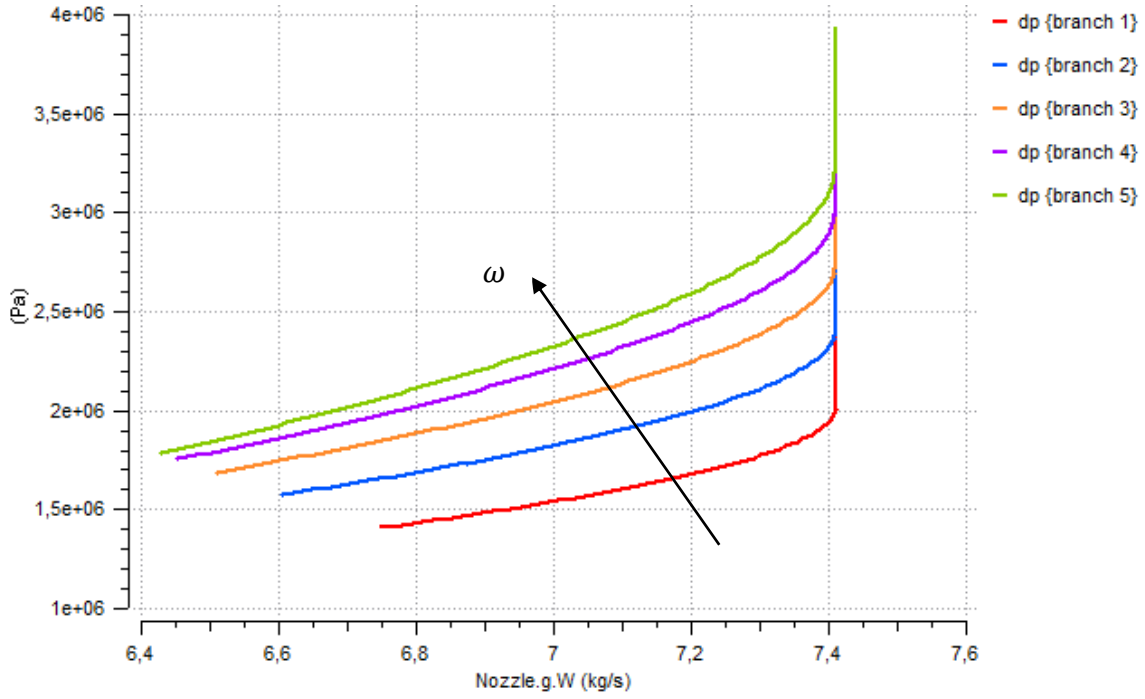
The temporary law is:

$$A = 0.0018 \cdot (0.8 + \frac{t}{40})$$

The simulation is done from $t = 0$ to $t = 20$.

Note that a discharge area of $0.0018\ m^2$ is the rounded-off discharge area obtained in the choked turbine design test for M=1. The same happens to the turbine input area, whose rounded-off value is $0.0019\ m^2$, and to the average radius, whose rounded-off value is $0.089\ m$. This value has been used for the Mach number to have a choked turbine if the discharge area is increased, and a non-choked turbine if it is decreased.

Finally, the performance map is:



As can be noted from the above chart, the mass flow does not change when the turbine is choked, but it varies when the turbine is not choked. Moreover, with the "Turbine" component, the results vary depending on the rotational speed.

Although this chart and the one obtained in the choked turbine performance test are different, both are correct, because in the choked turbine performance test, the average radius of the turbine was not specified (although the other parameters of the turbine are the same, it is not the same turbine if the average radius is changed). Hence, it is possible to have it always choked.

However, it is important to take care using the "Turbine_ch" component because when analysing a real turbine, the average radius will be known, and for example, if the average radius is the same as in this test, the turbine would not be choked for big discharge areas and therefore, the calculations done in the choked turbine performance test would not be correct because the performance map would be like in this experiment (the analysed turbine would be the same).

In this example, Type_AC=Angles in the turbine. It could have been changed to Type_AC=Coefficients and the result would be the same if phi_d= 1.14177006 and psi_d= 0.800971309, which are the values of the flow coefficient and the loading coefficient obtained in the previous design calculation.

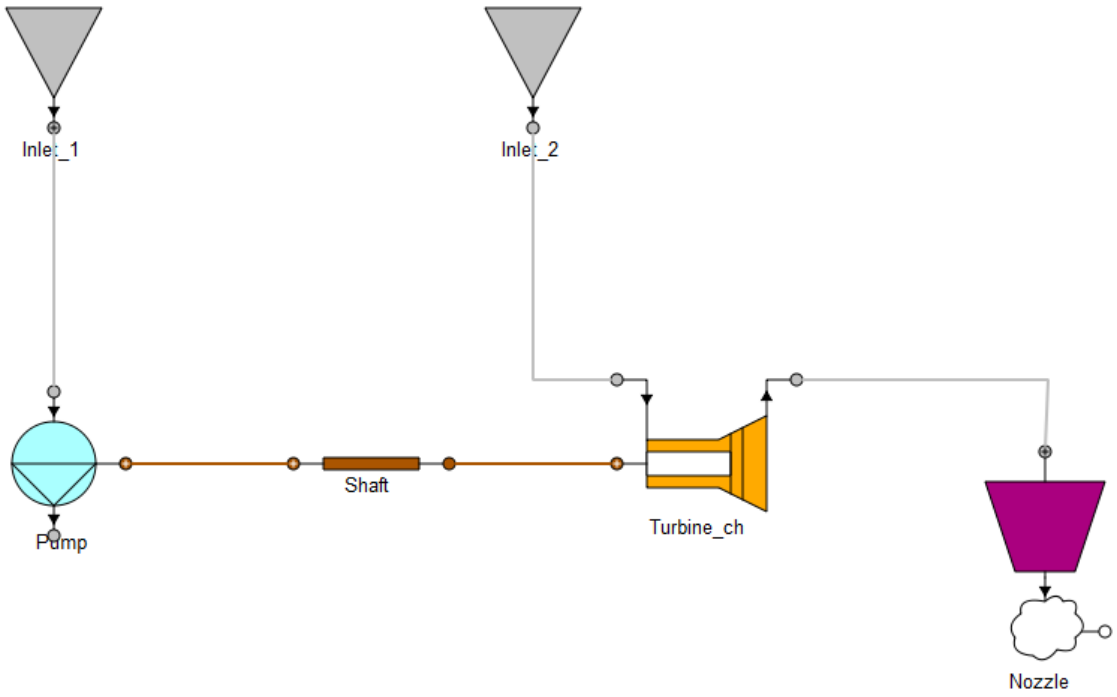
2.7 Turbopump design test

Model Name: turbopump_design_test.eds

Partition Name: default

2.7.1 Model description

The model represents a turbine which is moving a pump. The turbine is discharging to the atmosphere. The inlets set the inlet conditions of the pump and the turbine.



2.7.2 Results

A design calculation will be performed. To do this, create a default partition with the pump and the nozzle in Type=Design mode and the turbine in Type=Known_pi mode (which is a design mode together with Type=Known_W). Then, create a steady experiment. This experiment will calculate the pump input area, the pump average radius, the turbine input area, the discharge area and the mass flow through the turbine.

The mass flow through the pump is set in its inlet as data. In a complete engine, the pump mass flow would be set by the nozzle or another exhaust (see *Appendix A*).

The calculation has been performed for two cases of rotational speed (same for the turbine, shaft and pump): 30000 rpm and 10000 rpm. Note that the results are the same in both cases except for the pump average radius, which is smaller in the first case than in the second case.

2.8 Turbopump off-design test

Model Name: turbopump_offDesign_test.eds

Partition Name: default

2.8.1 Model description

The model is the same as the previous one.

2.8.2 Results

An off-design calculation will be performed. To do this, create a default partition with the pump, the turbine and the nozzle in Type=Off_design mode. Then, create a steady experiment.

The results obtained in the turbopump design test for a rotational speed of 30000 rpm have been used as data in this test. It can be checked that the results obtained in this test for the pump pressure increase are the same as the data used in the turbopump design test.

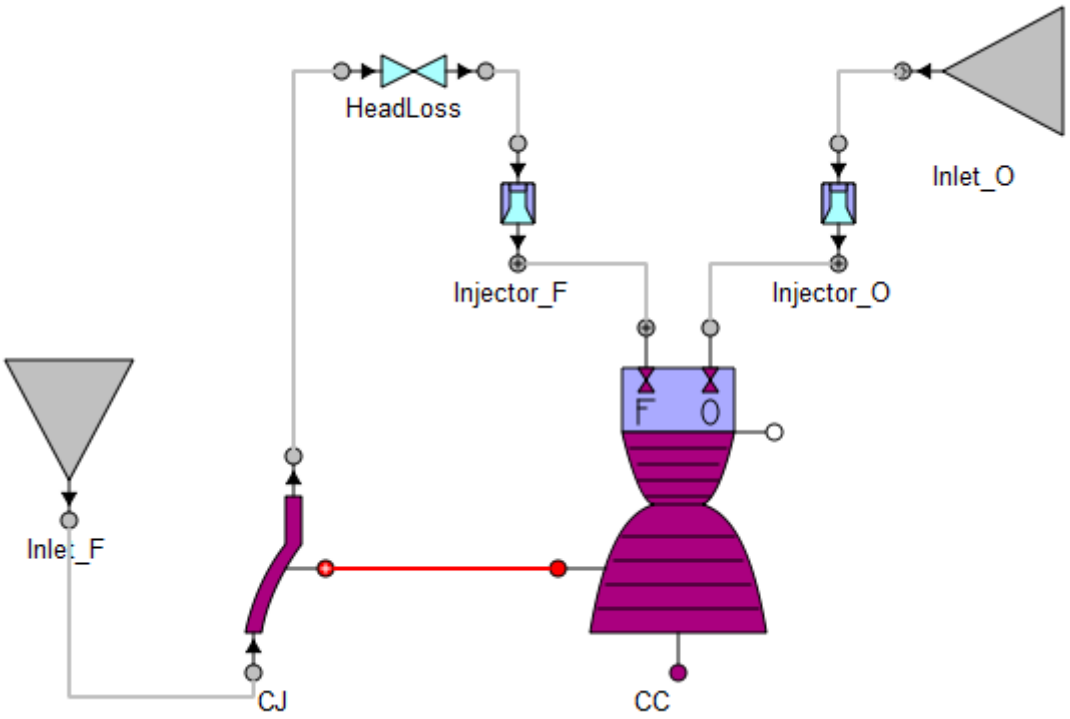
2.9 Cooling jacket design test

Model Name: coolingJacket_design_test.eds

Partition Name: default

2.9.1 Model description

The model represents the cooling circuit of gas generator engine. It is composed of a cooling jacket whose outlet is connected to the fuel injector of the combustion chamber. The inlets set the inlet conditions of the cooling jacket and the oxidant injector.



This test takes its parameters from the GGEngine example included in the ROCKET_EXAMPLES Library (version 3.1.0) from ESPSS.

2.9.2 Results

First of all, a design calculation will be performed. To do this, create a default partition with the injectors in Type=Design mode. Since the combustion chamber is being refrigerated, Cooled=Yes. Then, create a steady-state experiment.

A comparison will be made between the GGEngine example and this test. The GGEngine example is a non-stationary example, but, for the purpose of the comparison, the values taken belong to a point (TIME=3.95) in which the engine has reached the steady-state phase. Some representative values are shown in the following table:

	LPRES	ESPSS
Cooling jacket outlet total temperature	$141\ K$	$141\ K$
Cooling jacket outlet total pressure	$106.5 \cdot 10^5\ Pa$	$96.8 \cdot 10^5\ Pa$
Combustion temperature	$3586\ K$	$3554\ K$
Combustion pressure	$73.7 \cdot 10^5\ Pa$	$73.7 \cdot 10^5\ Pa$
Fuel injector area	$0.00834\ m^2$	$0.00793\ m^2$
Oxidant injector area	$0.00225\ m^2$	$0.002\ m^2$

The results are quite similar between both libraries, with an error of approximately 5%. The results are more accurate in the ESPSS models because, for example, the cooling jacket and combustion chamber are discretised.

The head loss is a "Regulator" component that has been included in order to perform a good calculation of the fuel injection area because it is a swirl injector. The head loss simulates the additional pressure drop generated by the swirl. It is important to bear in mind that C_D is not used for gases. Hence, its value in the fuel injector is unimportant.

The injection areas were imposed in the GGEngine and the mass flows were calculated. In this partition the mass flows are imposed and the injection areas are calculated. It is done in this way because in design mode it is easier to reach convergence than in off-design mode. Hence, it is recommendable to start always in design mode and then perform the off-design calculation. The off-design calculation performed in this test is done in the same way as in the GGEngine example.

As seen in the GGEngine data, the combustion chamber is longer than the nozzle. Hence, the characteristic section of heat exchange will be in the combustion chamber (and convergent zone of the nozzle) instead of in the divergent zone of the nozzle. It has been selected by means of the "Zone" variable. The value of data "AR_r" has been adjusted to improve the results.

Data "A_wet" of the "CombCha" component, which is the nozzle wet area of the cooled zone, has been taken from the GGEngine example. It can be calculated from the geometry of the combustion chamber and the nozzle.

2.10 Cooling jacket off-design test

Model Name: coolingJacket_offDesign_test.eds

Partition Name: default

2.10.1 Model description

The model is the same as the previous one.

2.10.2 Results

Change the injectors mode to Type=Off_design and set "A" in both injectors to the areas calculated in the previous experiment. Again, create a default partition and a steady experiment. To reach convergence, the value of data "W_F0" of the combustion

chamber must be changed to a value which is near the solution. The value given has been 20.

It can be checked that the results obtained in off-design mode are the same as the results obtained in design mode.

In some cases of this manual, the off-design mode results are made in order to show how to make off-design calculations, but the operating point of the engine has not been changed. The objective of an off-design calculation is to change the operating point of the engine and see its performance. Now the user can change some of the parameters in order to study the performance.

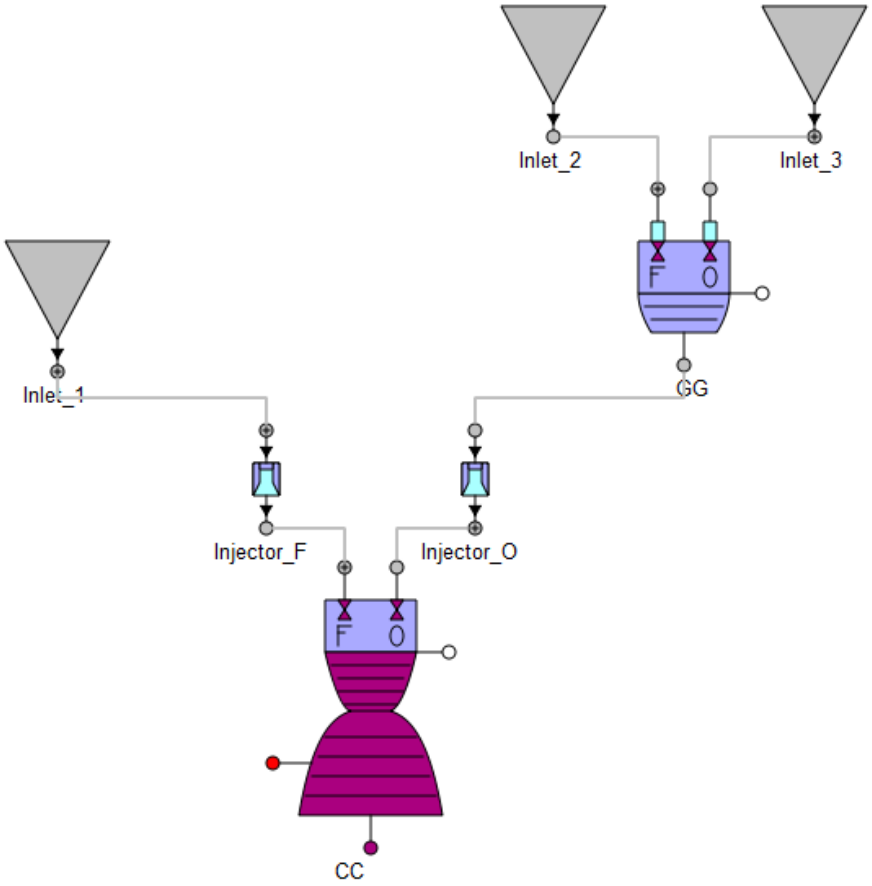
2.11 Staged combustion test

Model Name: staged_combustion_test.eds

Partition Name: default

2.11.1 Model description

The model constitutes an example in which the fluid of one of the combustion chamber inlets comes from a gas generator. It is done in order to enlighten the combustion model.



2.11.2 Results

A design calculation will be performed. To do this, create a default partition with the injectors and the combustion chamber in Type=Design mode. There is no refrigeration in this case. Hence, Cooled=No. Then, create a steady-state experiment.

Two calculations have been performed. In the first calculation, variable "OF" of the gas generator has been set to 16. Consequently, as the stoichiometric mixture ratio is equal to 8, there is oxidant excess in the gas generator. Hence, fuel will enter through the combustion chamber fuel inlet and combustion gases with excess of oxidant through the oxidant inlet. Therefore, the combustion will be produced in the combustion chamber.

Note that Combustion=TRUE and Q_comb_effective=Q_comb in the combustion chamber and the gas generator because combustion is produced in both.

In the second calculation, variable "OF" of the gas generator has been set to 6. Consequently, as the stoichiometric mixture ratio is equal to 8, there is fuel excess in the gas generator. Hence, fuel will enter through the combustion chamber fuel inlet and combustion gases with excess of fuel through the oxidant inlet. Therefore, the combustion will not be produced in the combustion chamber because there is no oxidant.

Note that in this case variable “Combustion” of the combustion chamber is equal to FALSE and “Q_comb_effective” to 0. In the gas generator Combustion=TRUE and Q_comb_effective=Q_comb because the combustion is produced in it. The calculation has not reached the convergence due to this fact.

The mass fraction of the used chemicals is printed in the experiment for both calculations. The mass fraction is included in port variable “fluid” and inside variables “fluid_O”, “fluid_F” and “fluid_P” of the combustion chamber and the gas generator. The following have been printed:

- Variable “fluid” at the outlet of the gas generator, which is the same as variable “fluid_P” of the gas generator because there are no unburned gases in it. It is also the same as variable “fluid” of the oxidant inlet port of the combustion chamber.

- Variable “fluid_O” of the combustion chamber, which is not the same as variable “fluid” of the oxidant inlet port of the combustion chamber because in variable “fluid_O” the combustion gases, which will not react, are not included.

- Variable “fluid_F” of the combustion chamber, which is the same as variable “fluid” of the fuel inlet port of the combustion chamber because there are no combustion gases entering through this port.

- Variable “fluid_P” of the combustion chamber.

- Variable “fluid” at the outlet of the gas generator, which is not the same as variable “fluid_P” of the combustion chamber because there are unburned gases in it (the unburned gases are the combustion gases that come from the gas generator).

Note that the sum of the three printed values is always equal to 1.

Besides, the combustion temperature and the outlet total temperature of the gas generator and combustion chamber have been printed for the first calculation. The combustion temperature has been calculated without taking into account the unburned gases. Hence, an adiabatic mix between the unburned gases and the burned gases in the component is done to calculate the total outlet temperature. Therefore, the combustion temperature and the outlet total temperature are equal in the gas generator because there are no unburned gases. In the combustion chamber they are not equal because there are unburned gases.

For the second experiment it makes no sense to print the combustion temperature and the outlet total temperature of the combustion chamber because there is no combustion in it.

It is important to be aware that the user does not have to change the values of “Q_comb”, “cp_P” and “M_P” although the mixture ratio is changed. Moreover, “Q_comb”, “cp_P” and “M_P” are the same in the gas generator and in the combustion

chamber because they work with the same fluids and these values are a property of the stoichiometric mixture between the oxidant and the fuel.

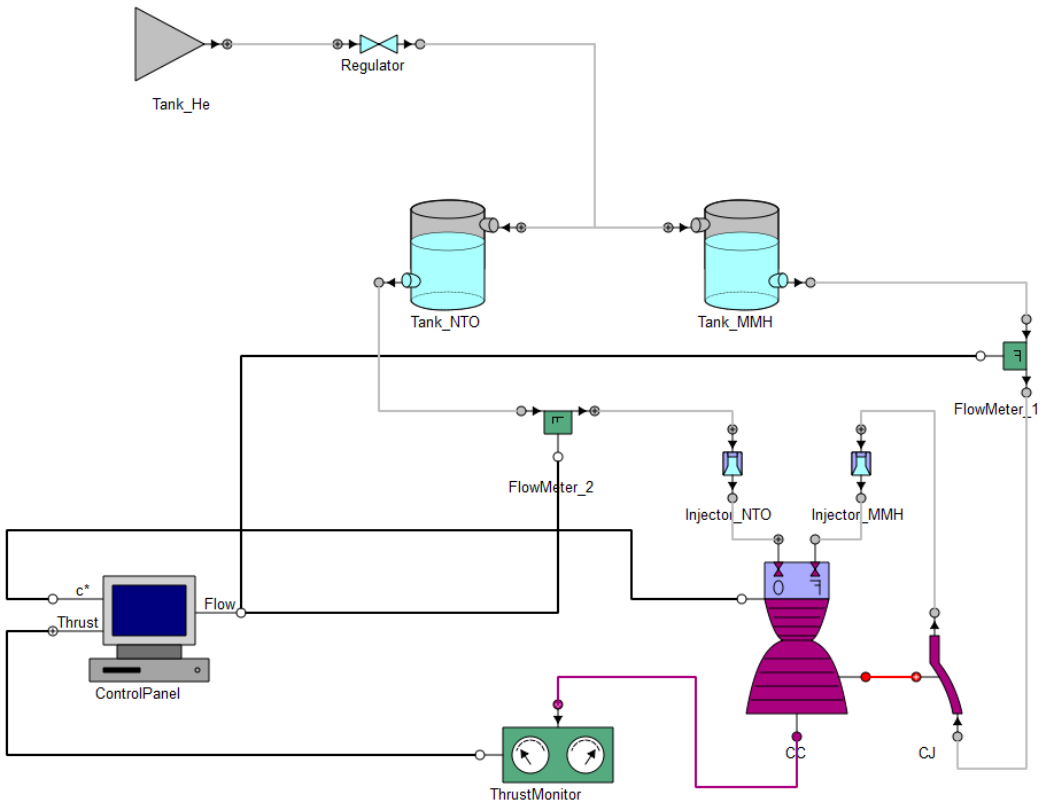
3 Engine Examples

3.1 Pressure fed engine design

Model Name: PressureFedEngine_design.eds

Partition Name: partition1

3.1.1 Model description



The model represents a pressure fed engine. The inlet simulates a gas tank. The regulator, in Type=Known_pt_out mode, imposes the outlet total pressure. Hence, the pressure of the tanks is fixed by the regulator. The tanks contain the oxidant and the fuel that will react in the combustion chamber. Besides, the fuel is used to cool the nozzle.

The flow meters, the control panel and the thrust monitor are included to calculate the global characteristics of the engine, such as the thrust, the specific impulse, the thrust coefficient or the characteristic exhaust velocity.

For this example, it will be assumed that the total pressure loss of the branching after component "Regulator" is null. In this case, it is not necessary to include a "SplitFrac" component after it, but it might be included. This is not possible for a junction, where the "Junction" component is always necessary.

This example takes its parameters from the PressureFedEngine example included in the STEADY_EXAMPLES Library (version 3.1.0) from ESPSS.

3.1.2 Results

First of all, a design calculation will be performed. To do this, set the injectors and the tanks to Type=Design mode. As the combustion chamber is being refrigerated, Cooled=Yes. Then, create a custom partition changing the suggested boundary, which is the altitude, to the ambient pressure "p_amb", which is a variable of the "ThrustMonitor" component. Finally, create a steady-state experiment and set the ambient pressure to 300. This is the ambient pressure given in the example done with ESPSS.

Generally, the ambient pressure is calculated by means of the altitude, but in this case, the altitude will be calculated by means of the ambient pressure. It is performed in this way because the ambient pressure was given instead of the altitude in the example done with ESPSS.

A comparison is going to be made between the example done with LPRES and with ESPSS. The pressurisation circuit of the example done with ESPSS is simulated as transient, but, for the purpose of the comparison, the values taken belong to TIME=15. Some representative values are shown in the following table:

	LPRES	ESPSS
Cooling jacket outlet total temperature	$386\ K$	$394\ K$
Cooling jacket outlet total pressure	$22.2 \cdot 10^5\ Pa$	$17.8 \cdot 10^5\ Pa$
Combustion temperature	$3023\ K$	$3032\ K$
Combustion pressure	$15.6 \cdot 10^5\ Pa$	$15.6 \cdot 10^5\ Pa$

The results are quite similar between both libraries, with an error of approximately 5%. The results are more accurate in the ESPSS models because, for example, the cooling jacket and combustion chamber are discretised.

Remember that C_D is not used for gases. Its value in the fuel injector is therefore unimportant.

In the example done with ESPSS the injection areas were imposed and the mass flows were calculated. In this partition the mass flows are imposed and the injection areas are calculated. It is done in this way because it is easier to reach convergence in design mode than in off-design mode. Thus, it is recommendable to start always in design mode and then perform the off-design calculation. The off-design calculation done with LPRES is performed in the same as in the example done with ESPSS.

As seen in the data of the example done with ESPSS, the combustion chamber is longer than the nozzle. Therefore, the characteristic section of heat exchange will be in the combustion chamber (and convergent zone of the nozzle) instead of in the divergent zone of the nozzle. It has been selected by means of variable "Zone". The value of data "AR_r" has been adjusted to improve the results.

Data "A_wet" of the "CombCha" component, which is the nozzle wet area of the cooled zone, has been taken from the example done with ESPSS. It can be calculated from the geometry of the combustion chamber and the nozzle.

3.2 Pressure fed engine design

Model Name: PressureFedEngine_offDesign.eds

Partition Name: partition1

3.2.1 Model description

The model is the same as the previous one.

3.2.2 Results

Change the injectors mode to Type=Off_design and set "A" to $0.0006\ m^2$ in the NTO injector and to $0.0015\ m^2$ in the MMH injector. Then, change the tanks mode to Type=Off_design and set "A_g" to $2 \cdot 10^{-5}\ m^2$ in the NTO tank and to $2.5 \cdot 10^{-5}\ m^2$ in the MMH tank. Note that these values are similar to the values calculated in the previous experiment. Again, create a custom partition changing the suggested boundary,

which is the altitude, to the ambient pressure “p_amb”, which is a variable of the “ThrustMonitor” component. Finally, create a steady experiment and set the ambient pressure to 300.

It can be checked that the results obtained in off-design mode are similar to the results obtained in design mode. They are not equal because the operating point of the engine has been slightly changed.

3.2.3 Aid to reach convergence

As explained before, in design mode it is easier to reach convergence than in off-design mode. Hence, it is recommendable to start always in design mode and then perform the off-design calculation.

Anyway, in some examples, it could be difficult for the user to set the initial values to reach convergence in an off-design experiment. As an example of how to solve the problem, at the end of the design experiment of this example, the sentence:

```
SAVE_STATE("PressureFedEngine_design_state.txt")
```

has been included. It has been done in order to use the results of the design experiment as the initial values for an off-design calculation. To use it in the off-design experiment, the user must write:

```
SET_INIT_ACTIVE(FALSE)

RESTORE_STATE("..\..\+pressure+fed+engine.design\exp1\Press
ureFedEngine_design_state.txt")
```

and later, set the new data and boundaries of the experiment. For this particular example, the user should write:

```
Tank_NTO.A_g = 2e-005

Tank_MMH.A_g = 2.5e-005

Injector_NTO.A = 0.0006

Injector_MMH.A = 0.0015
```

Everything must be written before the calculation sentence.

If the data and boundaries are not changed, the convergence will be reached without iterations because the initial condition is the own solution. The objective of an off-design calculation is to change the operating point of the engine and see its performance, but the design operating point can be a very good initial point to start the iterations.

In this example these sentences are commented because it is not necessary to use this aid to reach convergence.

This aid can also be found in other examples of this manual, but it is not necessary to use it in any of the examples.

3.3 Rocketdyne F-1 (gas generator cycle) design

Model Name: Rocketdyne_F1_design.eds

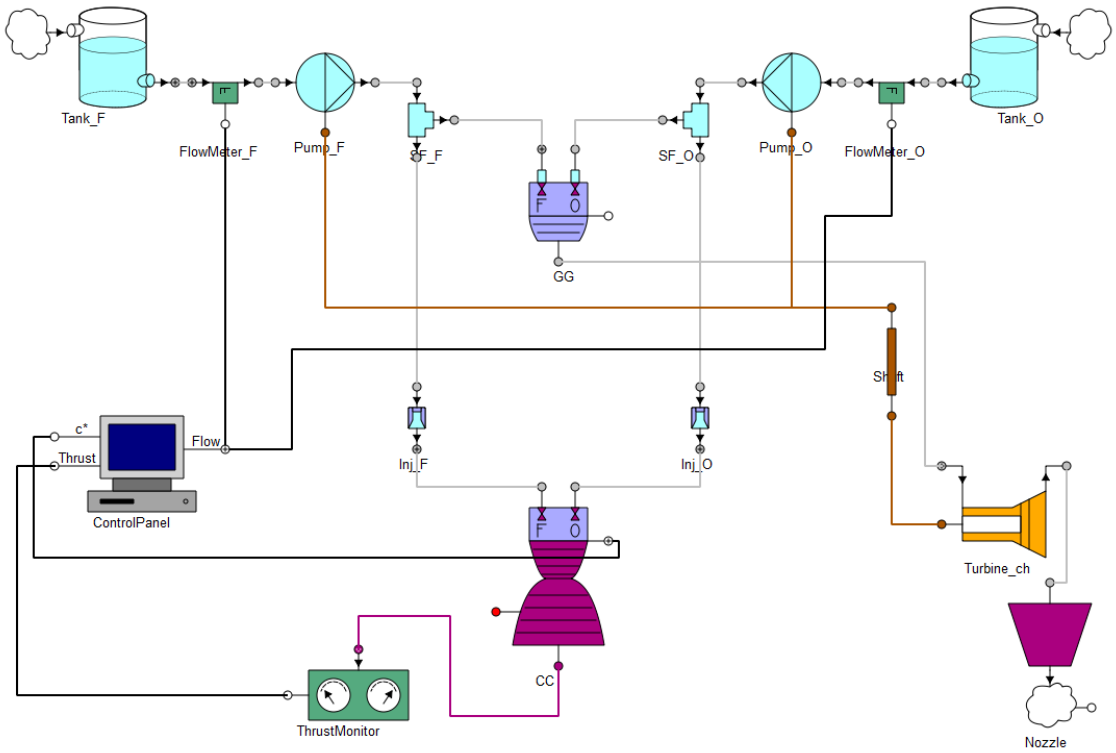
Partition Name: partition1

3.3.1 Model description

The model represents the Rocketdyne F-1 engine. It has a gas generator cycle. The tanks contain the oxidant and the fuel that will react in the combustion chamber. The pressure of the liquid of the tanks is increased in two pumps. A turbine is used to move the pumps. The turbine works with the combustion gases of a gas generator. The gas generator is fed by a part of the fuel and part of the oxidant, which are branched after the pumps.

The flow meters, the control panel and the thrust monitor are included to calculate the global characteristics of the engine, such as the thrust, the specific impulse, the thrust coefficient or the characteristic exhaust velocity.

Looking at the position of the flow meters, they are included before the branch. If they were included after the branch, the calculation of the specific impulse and the thrust coefficient would not be correct because the total mass flow of the engine would not be used. If the engine did not have any branch, the position of the flow meters in the flow line would not be important.



This example takes the great majority of its parameters from (NASA).

3.3.2 Results

A design calculation will be performed. To do this, set the pumps, the nozzle situated downstream from the turbine, the injectors and the combustion chamber to Type=Design mode. Set the turbine to Type=Known_W mode. There is no refrigeration in this case, so Cooled=No in the combustion chamber. Then, in the partition, transform the mass flows through the injectors to unknown variables and select the mass flow through the flow meters as boundaries. This is done in this way because in the documentation found about the Rocketdyne F-1 engine, the total oxidant and fuel mass flow used, not only the mass flows after the branches, were given. These are the mass flows of the injectors. Finally, create a steady experiment and set the altitude to 30000 m, which is approximately the average operating altitude of the engine. Therefore, it will be used as the design point.

From the experiment the pumps areas and average radiuses, the turbine input and discharge areas, the injectors areas, and the throat area of the combustion chamber among others are obtained.

3.4 Rocketdyne F-1 (gas generator cycle) off-design

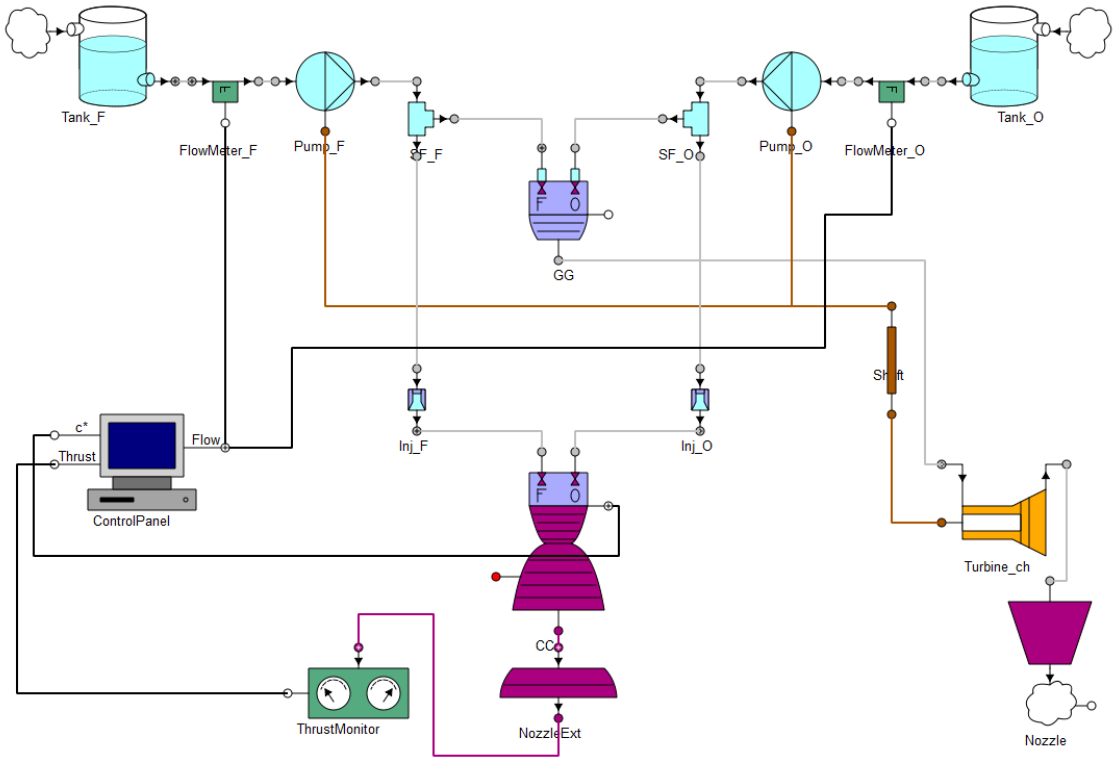
Model Name: Rocketdyne_F1_offDesign.eds

Partition Name: default

3.4.1 Model description

The model is the same as the previous one but, after the combustion chamber, a nozzle extension has been included to increase the area ratio.

Note that there are two pumps connected to the same port of the "Shaft" component. Likewise, two turbines could be connected to the same "Mechanical" port. Besides, a single turbine (or pump/compressor) could be connected to two different shafts to simulate, for example, a different efficiency in each one.



3.4.2 Results

The Rocketdyne F-1 engine can be found with and without nozzle extension. Without the nozzle extension, the area ratio is 10. The previous example was performed with this area ratio. In this example, the calculation will be performed with the nozzle extension, with which the area ratio is increased to 16. Hence, the area ratio of the nozzle extension is 1.6.

It can be checked that the results of this experiment for the design altitude are the same as if the area ratio of the previous example is changed to 16.

An off-design calculation is being performed. Set the pumps, the nozzle situated downstream from the turbine, the turbine, the injectors and the combustion chamber to Type=Off_design mode. Then, create a default partition and a steady experiment.

Set the area and the average radius of the pumps, the turbine input and discharge areas, the area of the injectors and the throat area of the combustion chamber to the values obtained in the design calculation of the Rocketdyne F-1 without nozzle extension example.

To reach convergence, the value of data "W_F0" of the combustion chamber and the same data of the gas generator must be changed to values which are near the solution. The values given have been 150 in the gas generator and 2000 in the combustion chamber.

Three calculations will be performed. The altitude was changed: design altitude, sea level and vacuum (space altitudes, out of the atmosphere).

A comparison between the results obtained from the experiment and the real values is presented in the following table:

	LPRES	Measured values
Thrust (sea level)	6.65 MN	6.77 MN
Specific impulse (sea level)	267 s	263 s
Thrust (vacuum)	7.79 MN	7.77 MN
Specific impulse (vacuum)	308 s	304 s

This example is the same as the previous one. To facilitate its completion, the previous schematic in off-design mode can be copied and then, the nozzle extension can be added. Besides, the experiment can also be copied.

3.5 RL10 (expander cycle) design

Model Name: RL10_design.eds

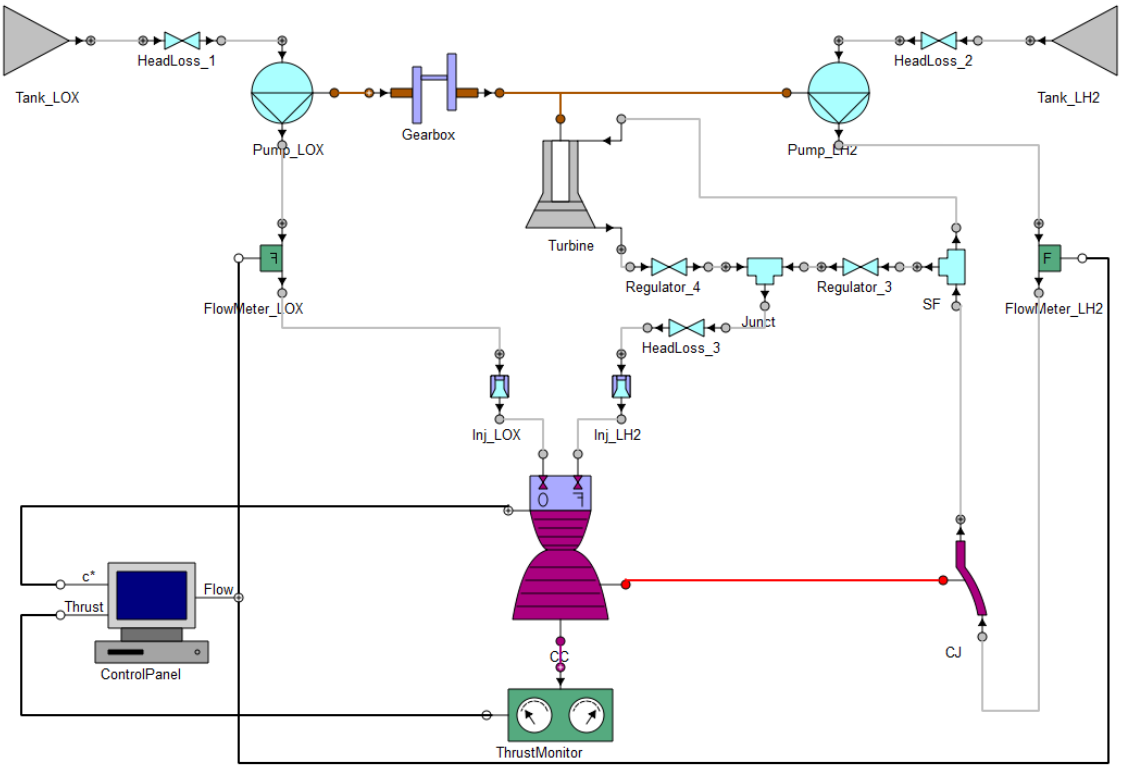
Partition Name: default

3.5.1 Model description

The model represents the RL10 engine. It has an expander cycle. The inlets simulate the liquid tanks. The tanks contain the oxidant and the fuel that will react in the combustion chamber. The pressure of the liquid in the tanks is increased in two pumps. A turbine is used to move the pumps. The turbine works with the fuel, whose enthalpy has been increased since passing through the cooling jacket, which is used to cool the nozzle.

The gearbox is used because the LOX pump works with a lower rotational speed.

The flow meters, the control panel and the thrust monitor are included to calculate the global characteristics of the engine, such as the thrust, the specific impulse, the thrust coefficient or the characteristic exhaust velocity.



This example takes its parameters from the RL10_cycle_ter example included in the STEADY_EXAMPLES Library (version 3.1.0) from ESPSS.

3.5.2 Results

A design calculation will be performed. To do this, create a default partition. Since the combustion chamber is being refrigerated, Cooled=Yes. Then, create a steady experiment and set the altitude to 9000 m. It is the altitude at which the ambient pressure is approximately 30000 Pa, which is the ambient pressure used in the example

done with ESPSS. The altitude in which the ambient pressure is 30000 Pa could also have been calculated, as done in the PressureFedEngine example.

A comparision is going to be made between the example done with LPRES and with ESPSS. Some representative values are shown in the following table (turbine design Mach number M=0.5):

	LPRES	ESPSS
Cooling jacket outlet total temperature	164 K	174 K
Cooling jacket outlet total pressure	$69.6 \cdot 10^5\ Pa$	$76.1 \cdot 10^5\ Pa$
Turbine mass flow	$2.1\ \frac{kg}{s}$	$2.2\ \frac{kg}{s}$
Turbine input area	$0.00069\ m^2$	$0.00042\ m^2$
Fuel injector area	$0.00178\ m^2$	$0.00177\ m^2$
Oxidant injector area	$0.00058\ m^2$	$0.00061\ m^2$
Combustion temperature	3181 K	3242 K
Throat area	$0.0134\ m^2$	$0.0116\ m^2$

The results are quite similar between both libraries, with an error of approximately 5%. The results are more accurate in the ESPSS models because, for example, the cooling jacket and combustion chamber are discretised.

The calculation of the turbine input area done with LPRES has not been accurate because the calculation of the turbine inlet pressure, which is the same as the cooling jacket outlet total pressure, has not been accurate.

The third head loss is component “Regulator” that has been included in order to perform a good calculation of the fuel injection area because it is a swirl injector. The head loss simulates the additional pressure drop generated by the swirl. Remember that C_D is not used for gases. Hence, its value in the fuel injector is unimportant.

As seen in the data of the example done with ESPSS, the divergent zone of the nozzle is longer than the combustion chamber plus the convergent zone of the nozzle. Hence, the characteristic section of heat exchange will be in the divergent zone of the nozzle. It has been selected by means of variable “Zone”. The value of data “AR_r” has been adjusted to improve the results.

Data “A_wet” of the “CombCha” component, which is the nozzle wet area of the cooled zone, has been taken from the example done with ESPSS. It can be calculated from the geometry of the combustion chamber and the nozzle.

It could be an interesting practice to evaluate the engine performance when changing the turbine mass flow.

Note that in this example, in the staged combustion test and in the cooling jacket test, the values of "Q_comb", "cp_P" and "M_P" are the same because they work with the same fluids and these values are a property of the stoichiometric mixture between the oxidant and the fuel.

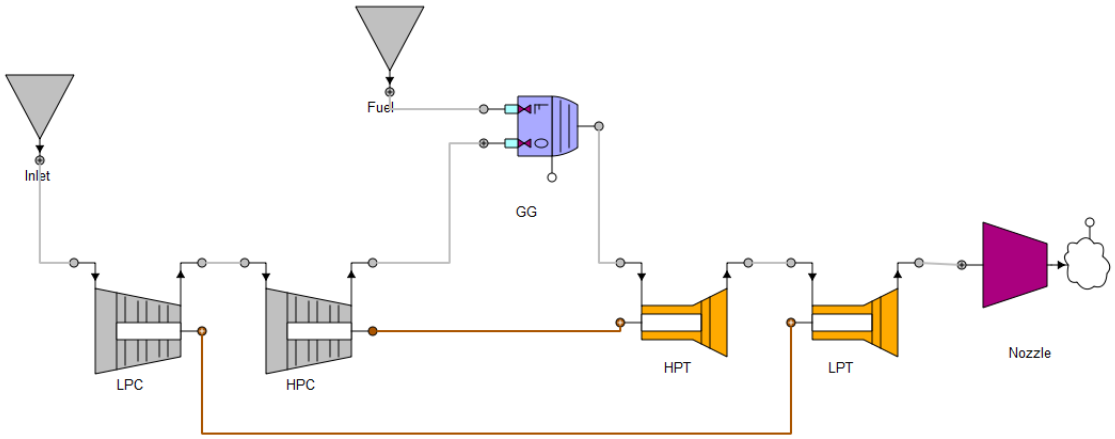
3.6 Turbojet Engine

Model Name: TurbojetEngine.eds

Partition Name: default

3.6.1 Model description

This example is made to show the possibility of analysing jet engines with the LPRES Library. The model represents a turbojet engine. The first inlet simulates the air intake. As total conditions are given, it does not need a diffuser (considering the diffuser as ideal). The total pressure is increased in the compressor. Then, the combustion is produced in the gas generator. The second inlet simulates the fuel tank. The power of the compressors is given by the turbines. The turbines are moved by the hot combustion gases. Finally, the combustion gases are ejected by the nozzle.



Note that if the efficiency of the shaft between the compressors and the turbine is 1, as assumed for this example, it is not necessary to use a "Shaft" component to connect them.

3.6.2 Results

A design calculation will be performed. To do this, create a default partition. Then, create a steady experiment. Set the altitude to 11000 m.

The total pressure of the second inlet must be higher than the high pressure compressor outlet because the injectors of the gas generator are modelled as:

$$p_{t,out} = p_c = TPL \cdot min(p_{t,O}, p_{t,F})$$

Hence, if the total pressure of the second inlet is set to its real value, which is typically the ambient pressure, the combustion pressure would be the ambient pressure, which is not real. This is because the model of the gas generator was thought for rocket engines in which the oxidant mass flow is similar to the fuel mass flow. In a jet engine, the oxidant mass flow is much higher than the fuel mass flow.

The value given to "Q_comb" has been $44 \cdot 10^6/15.6$. In jet engines, it is fairly common to work with the heat of combustion per fuel mass flow unit, usually denoted as L. For a mixture between kerosene and air, $L = 44 \cdot 10^6\ J/kg$. To transform the heat of combustion per fuel mass flow unit to the heat of combustion per oxidant mass flow unit, L must be divided by OF_{st}.

Besides, in jet engines, instead of working with mixture ratio O/F, it is more common to work with $f = F/O$. A typical value for f is 0.02. Therefore, $O/F = 1/0.02$.

In a turbojet engine the cross sectional area of the compressor decreases in the flux direction in order to help the compression. Looking at the results, note that the area of the first compressor is bigger than the area of the second compressor.

4 References

Empresarios Agrupados. (2015). *ESPSS Software Transfer Document.*

Empresarios Agrupados. (2015). *ESPSS: European Space Propulsion System Simulation - EcosimPro Libraries User Manual.*

Empresarios Agrupados. (2015). *Software Verification and Validation Plan.*

NASA. *F-1 Engine.* Saturn V News Reference.

Vilá Vacas, J. (2012). *Diseño y Optimización de Motores Cohete de Propulstante líquido con EcosimPro/ESPSS (Master's Thesis).* Universidad Politécnica de Madrid.

Zarchan, P., Yang, V., Habiballah, M., Hulka, J., & Popp, M. (2004). *Progress In Astronautics and Aeronautics: Liquid Rocket Thrust Chambers.* AIAA.

Appendix A: On-off design mode rules

Turbomachinery rules

If component “Pump” is in design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq” components connected to it must be in design mode too. If component “Pump” is in off-design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq”components connected to it must be in off-design mode too.

If component “Compressor” is in design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq” components connected to it must be in design mode too. If component “Compressor” is in off-design mode, the “Turbine_ch”, “Turbine” or “Turbine_liq” components connected to it must be in off-design mode too.

In design mode, the pump/compressor sets the power. The rotational speed is set by the turbine.

In off-design mode, the pump/compressor sets the rotational speed. The power is set by the turbine.

The mass flow in a pump/compressor is always set by another component. The mass flow in a turbine is set by another component in off-design mode and by itself in design mode.

If a turbine in design mode does not have a pump/compressor connected to it, it would be possible to set the mass flow to it as a boundary or by another component only if the power is not given as a boundary.

Exhaust rules

The “Injector” components don’t have to be in the same mode as the “CombCha” component connected to it.

A “CombCha” component plus its connected “Injector” components set the oxidant and the fuel mass flow in any mode.

If component “Nozzle” or “Ambient” are in off-design mode, the exhaust mass flow is set. If component “Nozzle” or “Ambient” are in design mode, the exhaust mass flow must be set by another component.

If component “Nozzle” is connected to a “Turbine_ch”, “Turbine” or “Turbine_liq” component, they must be in the same mode. This is because in design mode the turbine imposes the mass flow (with a pump or compressor connected to it or by giving the power of the turbine as a boundary) but in off-design mode the turbine does not impose it. Note that more data is given in the turbine design mode, than in the off-design mode.

If there is an “Injector” component downstream from a turbine, both must be in the same mode. If they are in off-design mode, there is no problem. If they are in design mode, it is not possible to choose Type=Known_pi. Instead, Type=Known_W should be chosen for the design mode of the turbine and data “W” set to the same value as the “W” data of the injector. Otherwise, only one of the data will be taken into account and the rest will not.

Issues to Consider

Note that inside a component with on-off design mode, the variables/data will be repeated. As an example, component “Ambient” will be analysed. It has data “A” which is the area, and variable “A_d”, which is the design area. In design mode, “A_d” is calculated, and data “A” is not taken into account. In off-design mode, A=A_d. The same thing happens in the other components.

In the **On-off design mode** section of each component, it is always considered that the inlet total pressure, the inlet total temperature (or temperature in the event of liquids) and the inlet working fluid (“fluid” variable) are known to say what will be calculated by the component in on-design and in off-design mode. This is what will happen if the user does not change anything.

Component “Inlet” when Type=All can set the mass flow.

Note that when a component is in off-design mode, the analysis is done as if it were manufactured because the given values are areas, lengths, etc.

Whenever this appendix states that a component is downstream from another, they do not have to be directly connected, they can have other components in between, but they must belong to the same flow line, which means that the mass flow is constant through the line (there are no “Junction” or “SplitFrac” components).