Abstract

Object oriented modeling of physical systems is an interesting paradigm, which has the potential to offer reusable models and model components. The aim of this study is to investigate how to build mean value models for automotive engines. MathModelica, a modeling tool for the object oriented modeling language Modelica, is used in this study.

Several sub models have been developed for the different parts of the engine. The models cover the air filter, intercooler, throttle, base engine, exhaust system, compressor, turbine, turbine shaft, and volumes. It is shown how the components can be connected to form both turbo charged engines as well as a naturally aspirated engines, which shows that the paradigm is applicable for the modeling and confirms the modeling principle. One problem that has popped up at several occasions is the selection of initial conditions for the simulation. Especially when restrictions with low pressure drops are connected between two volumes, the simulation engine has problems finding initial conditions.

The models have been compared to measured engine data collected at a test bench in Vehicular Systems laboratory at Linköping University. The agreement with measurement data is good and the models work as expected.

Nomenclature

Symbol	**Quantity**	**Unit**
A	Area	m^2
(A / F)	air to fuel ratio	-
c_p	specific heat (constant volume)	J / (kg K)
c_v	specific heat (constant pressure)	J / (kg K)
d	diameter	m
g	gravitational force	m / s^2
h	specific enthalpy	-
L	length	m
m	mass	kg
$\dot{m}$	mass flow	kg / s
M	molar mass	kg / mol
p	pressure	Pa
q	specific heat	J / m^2
Q	heat	J
$\dot{Q}$	heat flow	W = J / s
$\dot{q}$	heat flux	W / m^2
r	radius	m
R	molar gas constant	J / (kg K)
S	circumferance	m
t	time	s
T	temperature	K
u	internal energy per unit mass	J / kg
U	internal energy	J
v	specific volume velocity	m^3 / kg m / s
V	volume	m^3
w	specific work	J / kg
W	work	J
x, y, z	position coordinates	m

subscript	**explanation**
0	stagnation or total
1	inlet
2	outlet
c	cold fluid
F	fluid
h	hot fluid
i	inner
m	mean / average
o	outer
V	volume
W	wall

Contents

Introduction

This Paper presents a Mean Value Engine Model (MVEM) of a turbo charged spark ignited engine that has been developed at the department of Vehicular Systems, ISY, University of Linköping in cooperation with MathCore and Mecel. The model consists of several sub models for the different parts of the engine. These sub models are mostly the same as those described in [Bergström & Brugård, 1999], [Nielsen & Eriksson, 2000], [Eriksson et.al., 2001], and [Pettersson, 2000].

When modeling a vehicle engine it is desirable to be able to decide the model in sub models corresponding to the components of a real engine, such as intercooler, compressor, and cylinders. By connecting this together in different ways, different engines may be studied. To make this possible the model is implemented in Modelica, a language especially well suited for physical modeling.

To understand the underlying physics Chapter 1-3 present the theory on which the models rely; thermodynamics, heat transfer and fluid mechanics. The actual model and it's components are presented in Chapter 5 and 6. Finally Chapter 7 presents conclusions and future possibilities.

1. Thermodynamics

Thermodynamics constitutes the basic tool for analyzing systems that involve fluid flow in tubes as well as heat and work transfer of compression and expansion processes. In this chapter some of the most important relationships are presented together with assumptions and restrictions that have to be made for the system in question. There will also be some examples where the basic expressions are used to calculate general model structures.

1.1 Reversible Process

Definition *A process is reversible if, after it has occurred, both the system and the surroundings can by any means whatsoever be returned to their original states. Any other process is irreversible.*

Due to for example friction, mixing, and free expansion all real processes are irreversible, but even though reversible processes do not occur, they are extremely useful and serve as standards of comparison since they often are the limiting cases of actual processes. For example, the reversible process provide the maximum work from work-producing devices and the minimum work input to devices that consume work, e.g. turbines and compressors.

1.2 Ideal Gas, Specific Heat and Enthalpy

An ideal gas is defined as a substance for which the following equation is valid:

$$\mathrm{p}\, v = \mathrm{R}\,\mathrm{T} \tag{1.1}$$

where *p* is the pressure, *v* the specific volume, i.e., volume per unit mass, *T* the temperature and *R* the gas constant. *R* has to be calculated for the substance considered and is given by

$$R = \frac{\overline{R}}{M}$$

where the universal gas constant $\overline{R}$ has the value $\overline{R}$ = 8.31441 [J / (mol K)]. M denotes the molar mass of the substance and for dry air M = 28.97 [kg / mol], so the gas constant R for air has the value R = 287 [J/(kg K)]. The value of R varies with the relative humidity of the air, but the deviation is quite small and can be neglected in most practical cases.

The specific volume is defined as $v = \frac{V}{m}$, where V is the volume of the system and m the mass of the gas within the system, the equation can be put in the following alternative form

$$p\,V = m\,R\,T \tag{1.2}$$

With the assumption of ideal gas, the specific internal energy u and a property called specific enthalpy, defined as $h = u + pv$, becomes functions of the temperature only. That is

$$u = u(T) \qquad \text{and} \qquad h = h(T)$$

The specific heat at constant volume, c_v, and constant pressure, c_p, of an ideal gas are defined in terms of internal energy and enthalpy

$$c_v = \left(\frac{\partial u}{\partial T}\right)_v = [u = u(T)] = \frac{\mathrm{du}}{\mathrm{dT}} \Longrightarrow \mathrm{du} = c_v\,\mathrm{dT} \tag{1.3}$$

$$c_p = \left(\frac{\partial h}{\partial T}\right)_p = [h = h(T)] = \frac{\mathrm{dh}}{\mathrm{dT}} \Longrightarrow \mathrm{dh} = c_p\,\mathrm{dT} \tag{1.4}$$

Note that c_p and c_v are functions of temperature but for limited temperature intervals, such as in the intake system, they can be regarded as constant. By using the rightmost expressions in the equations above together with the definition $h = u + pv = u + RT$, some useful relations can be derived. Differentiation of the previous relation with respect to temperature gives the following, since R is a constant

$$\frac{\mathrm{dh}}{\mathrm{dT}} = \frac{\mathrm{du}}{\mathrm{dT}} + R\,\frac{\mathrm{dT}}{\mathrm{dT}} \Longrightarrow c_p = c_v + R \tag{1.5}$$

The ratio of the specific heats, γ, is defined as

$$\gamma = \frac{c_p}{c_v} \tag{1.6}$$

Combining equations (1.5) and (1.6) results in the following expressions for c_p and c_v

$$c_v = \frac{R}{\gamma-1} \qquad \text{and} \qquad c_p = \frac{R\gamma}{\gamma-1}$$

For monatomic gases, that is a gas with molecules that consist of one atom only, e.g. the inert gases *He* and *Ne*, and for other gases in limited temperature ranges, the specific heats can be considered constant. This assumption simplifies the calculations of internal energy difference, Δu, and enthalpy difference, Δh. Integrating the expressions for *du* and *dh* in equations (1.3) and (1.4) yields

$$\Delta \mathrm{u} = \int c_v \, \mathrm{dT} \approx c_v \, \Delta \mathrm{T} \tag{1.7}$$

$$\Delta \mathrm{h} = \int c_p \, \mathrm{dT} \approx c_p \, \Delta \mathrm{T} \tag{1.8}$$

1.3 One Dimensional Steady Flow

A common assumption when analyzing thermodynamic systems is to say that the fluid flow is steady and one dimensional. One dimensional flow exists if the velocity, temperature and other properties are uniform at each cross section of the flow. From one cross section to another the properties may change, but for each value of the coordinate, defined as distance in the direction of the flow, there is a single value of the velocity, a single value of the density and so on. This is illustrated in Figure 1.1.

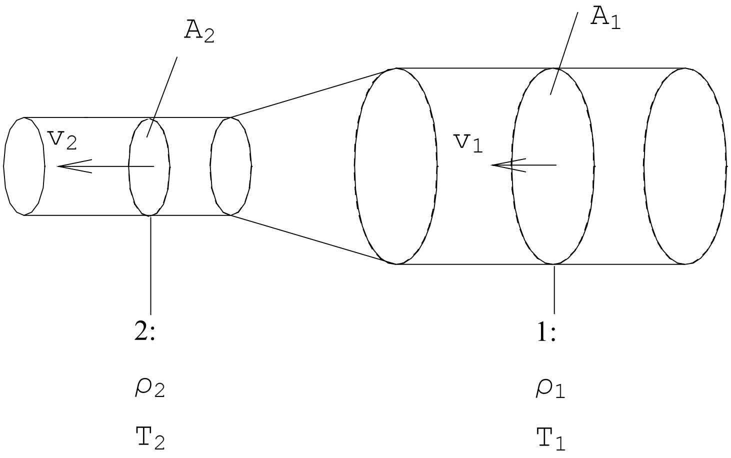


Figure 1.1. One dimensional flow. Throughout each section 1 and 2 with cross sectional area A_1 and A_2 respectively, the velocity, density and temperature are the same.

The assumption of one dimensional flow simplifies the calculation of certain properties of the flow. The velocity, *v*, normal to the cross section of area, *A*, is

$$v = \frac{\dot{m}}{\rho A} \tag{1.9}$$

where ρ is the density of the fluid. The continuity equation expresses the rate of change of mass within the system at any instant

$$\frac{\mathrm{d}}{\mathrm{dt}} m = \rho_1 v_1 A_1 - \rho_2 v_2 A_2 \tag{1.10}$$

This equation may be used to calculate an expression for the pressure in a system in terms of the mass rate of flow in and out of it. This is used to develop a volume model in Section 6.8.

If the properties at each cross section are also independent of time, that is $\dot{m}_1 = \dot{m}_2$, the fluid is referred to as *one dimensional steady flow*. In this case the mass within the system is constant, and thus the continuity equation is reduced to

$$\rho_1 v_1 A_1 = \rho_2 v_2 A_2$$

By combining equation (1.10) with the expression for the velocity, the following useful relationship can be stated for a one dimensional steady flow system with one inlet and outlet

$$\dot{m} = \rho_1 v_1 A_1 = \rho_2 v_2 A_2$$

Later on in this chapter we also introduce the restriction that the fluid is incompressible. That is $\rho_1 = \rho_2 = \rho$ and the above equation reduces to

$$\dot{m} = \rho v_1 A_1 = \rho v_2 A_2 \qquad (1.11)$$

1.4 The First Law of Thermodynamics

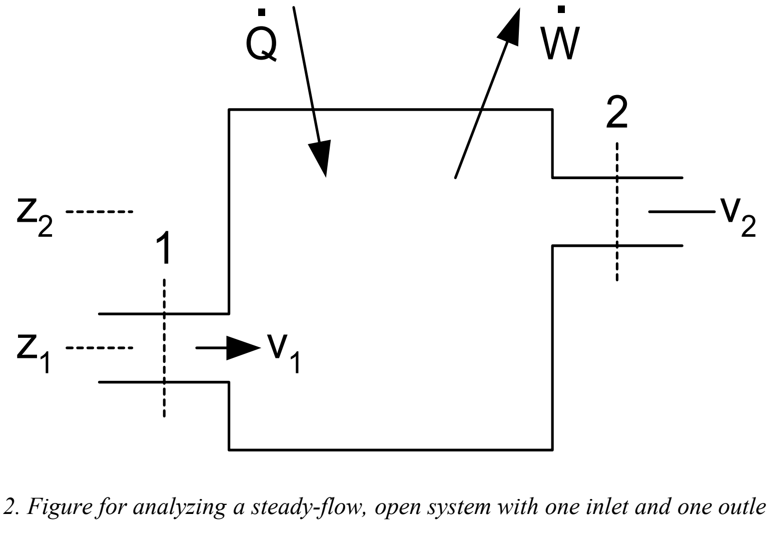


Figure 1.2. Figure for analyzing a steady-flow, open system with one inlet and one outlet.

The first law of thermodynamics, expresses the conservation of energy. A general statement of this law for a steady flow system, i.e. a system in which no mass is stored, is

$$\begin{pmatrix}\texttt{Net amount of}\\ \texttt{energy added to}\\ \texttt{the system as}\\ \texttt{heat and all}\\ \texttt{forms of work}\end{pmatrix} = \begin{pmatrix}\texttt{Stored}\\ \texttt{energy}\\ \texttt{of matter}\\ \texttt{leaving}\\ \texttt{the system}\end{pmatrix} - \begin{pmatrix}\texttt{Stored}\\ \texttt{energy}\\ \texttt{of matter}\\ \texttt{entering}\\ \texttt{the system}\end{pmatrix} \tag{1.12}$$

or

$$\oint \delta \mathrm{Q} = \oint \delta \mathrm{W} \tag{1.13}$$

Work and heat are denoted *W* and *Q* respectively. Work is defined positive when done by the system while heat defined positive when added to the system. This is illustrated in Figure 1.2 where the arrows point in the positive direction for *W* and *Q*. Denote the specific enthalpy *h*, the speed of the fluid *v* and the height from a reference point *z*, see Figure 1.2. Using this, and the assuming that the effects of electricity, magnetism, and surface tension are negligible , the first law can be written in the following form:

$$\dot{Q} - \dot{W} = \dot{m}\left(h_2 - h_1 + \frac{v_2{}^2 - v_1{}^2}{2} + g(z_2 - z_1)\right)$$

Dividing this equation by the mass rate of flow, neglecting the heat transfer with the surroundings and the change in potential energy, the first law can be expressed in terms of the specific work, *w*, i.e. the amount of work per unit mass in the system

$$-w = h_2 - h_1 + \frac{v_2{}^2 - v_1{}^2}{2} = h_{02} - h_{01} \tag{1.14}$$

In the rightmost equality, the terms for the static and the dynamic enthalpy are put together to form the so called total or *stagnation* enthalpy,

$$h_0 = h + \frac{v^2}{2} \tag{1.15}$$

Together with total enthalpy, both total pressure and total temperature may be defined, and for an ideal gas with constant specific heats they become

$$p_0 = p + \frac{\rho v^2}{2} \tag{1.16}$$

and

$$T_0 = T + \frac{v^2}{2 c_p} \tag{1.17}$$

1.5 The Second Law of Thermodynamics

Definition, Clausius Statement *It is impossible for any device to operate in such a manner that it produces no effect other than the transfer of heat from one body to another body at a higher temperature.*

Clausius statement is one of many different forms in which the second law of thermodynamics has been expressed. However, if any one of the statements is accepted as a postulate, all the other statements can be proved from this starting point. Note that the statement does not say that it is impossible to transfer heat from a lower-temperature body to a higher temperature-body, but for this to happen an energy input is needed, usually in the form of work.

2. Heat Transfer

A thorough understanding of heat transfer is important, especially when modeling heat exchangers, such as an intercooler, but is also helpful when understanding the losses due to heat transfer that occur in other processes.

Heat, Q, is a form of energy, consequently it is measured in joule, [J]. The heat flow, or heat per unit time is given by $\dot{Q} = \frac{\partial Q}{\partial t}$, with the unit watt, W = [J/s]. Finally the heat flux, $\dot{q} = \dot{Q}/A$ is the heat flow per unit area W/m^2. The transfer of heat consists of three distinct phenomena:

1. Heat transfer by conduction.
2. Heat transfer by convection.
3. Heat transfer by radiation.

At low and moderate temperatures convection and conduction play a major role in heat transfer, while loss of heat due to radiation may often be neglected. In the following section heat transfer due to conduction and convection will be treated and heat transfer efficiency will be defined.

2.1 Heat Transfer by Conduction

Heat conduction is the transfer of energy between molecules in a substance due to a temperature gradient. In metals also the free electron transfer of energy. The heat flows from one part of a medium at a higher temperature to another part at a lower temperature, according to the second law of thermodynamics.

The transfer of energy in a conductive material is described by the field of heat flux

$$\bar{\dot{q}} = \bar{\dot{q}}(\bar{x}, t)$$

where $\bar{x}$ is the position vector and t time. The mechanism of heat conduction is theoretically hard to understand. Nevertheless, if we consider the temperature gradient, ∇T, as the cause of heat flow in a conductive material, it suggests that a simple proportionality between cause and effect may be assumed, allowing the heat flux to be written as

$$\dot{\bar{q}} = -\lambda \cdot \nabla T \tag{2.1}$$

This is the *Fourier law* for the conduction of heat. The proportionality constant λ, in some literature referred to as *k*, is a property of the material, called *thermal conductivity*, and may depend on both temperature *T* and pressure *p*. The thermal conductivity is a scalar as long as the material is *isotropic*, i.e. the ability of the material to conduct heat depends on position within the material, but for a given position not on the direction.
At steady state conditions the conductive heat transfer in a fluid can be calculated using an analogy to electronic circuit theory. This is illustrated in the following example.

Example, Electrical Analogy

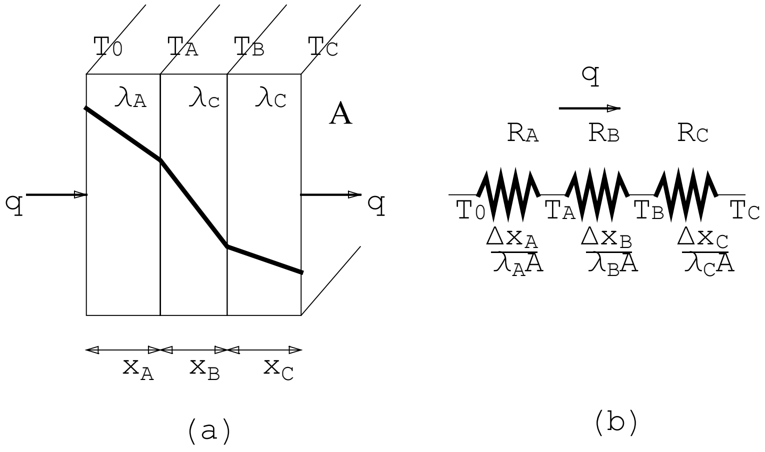


Figure 2.1. Heat flow through a composite wall.

Suppose we want to calculate the heat transfer for the composite wall in Figure 2.1 (a) First consider the first part of the wall, with area A, wall thickness Δx_A, wall face temperatures T_0 and T_A and constant thermal conductivity λ_A. Integration over area of the heat flux given in equation (2.1) yields the heat transfer rate

$$\dot{Q} = -\lambda \cdot A\ \frac{T_A - T_0}{\Delta x_A}$$

Note that the heat flow must be the same through all sections, so

$$\dot{Q} = -\lambda_A \cdot A\ T_A - \frac{T_0}{\Delta x_A} = -\lambda_B \cdot A\ T_B - \frac{T_0}{\Delta x_B} = -\lambda_C \cdot A\ T_C - \frac{T_0}{\Delta x_C}$$

Solving these three equations simultaneously, the heat flow can be written

$$\dot{Q} = \frac{T_0 - T_C}{\Delta x_A/\lambda_A\, A + \Delta x_B/\lambda_B\, A + \Delta x_C/\lambda_C\, A}$$

Considering the heat transfer rate as a flow and the combination of thermal conductivity, thickness of the wall and area as a resistance of this flow, yields

$$\text{heat flow} = \frac{\text{thermal potential difference}}{\text{thermal resistance}}$$

where thermal resistance is defined as

$$R_{\text{th}} = \frac{\Delta \text{x}}{\lambda} A$$

We have a relation similar to Ohms law in electric circuit theory (I=U/R). The one-dimensional heat flow equation for this type of problem may be written

$$\dot{Q} = \frac{\Delta \text{T}_{\text{overall}}}{\Sigma \text{R}_{\text{th}}} \tag{2.2}$$

This electrical analogy to the heat transfer is illustrated in Figure 2.1 (b).

□

2.2 Heat Transfer by Convection

In a flowing fluid, the heat is transferred not only by conduction, but also by the macroscopic movement of the fluid. The latter is known as convective heat transfer and according to [Lakshminarayana, 1996] it is the most common form of heat transport in turbomachinery flows.

Heat transfer between a solid wall and a fluid, e.g. in a cooled tube with a warm gas flowing inside it, as in the intercooler, is of special technical interest. It is known from the *boundary layer theory* founded by Ludwig Prandtl, that the fluid layer close to the wall has the greatest effect on the amount of heat transfer. In the boundary layer the velocity component parallel to the wall changes, over a small distance, from zero at the wall to almost the maximum some distance from the wall. The temperature in the boundary layer also changes in a similar way. Heat will flow from the warm fluid to the cold wall as a result of the temperature difference. Heat will flow in the opposite direction if we have a cool fluid inside a warm tube. The heat flux from the wall $\dot{q}_W$ depends on the temperature and velocity fields in the fluid. As these fields are quite complex to evaluate, it is common to use a simplified model, named *Newtons law of cooling*:

$$\dot{q}_W = \alpha\,(T_W - T_F)$$

where T_W is the temperature at the wall and T_F the fluid temperature. From this relation the local *heat transfer coefficient* α, in some literature referred to as h, is defined:

$$\alpha \equiv \frac{\dot{q}_W}{T_W - T_F} \tag{2.3}$$

If the immediate neighborhood of the wall is considered, the fluid velocity is zero, hence energy can only be transported by heat conduction. That is the heat flux is given by Fouriers law (2.1). If the distance to the wall is denoted by y the heat flux is given by

$$\dot{q}_W = -\lambda \, T_W \left(\frac{\partial T}{\partial y}\right)_W$$

Putting this in to equation (2.3) gives

$$\alpha = -\lambda \, \frac{\left(\frac{\partial T}{\partial y}\right)_W}{T_W - T_F} \tag{2.4}$$

So far only the local heat transfer coefficient, which can be different at every point of the wall, has been considered. In practice only an *average heat transfer coefficient* α_m is required to evaluate the flow of heat A into the fluid:

$$\dot{Q} = \alpha_m \, A \, \Delta \mathrm{T}$$

or

$$\alpha_m = \frac{\dot{Q}}{A \, \Delta \mathrm{T}} \tag{2.5}$$

We conclude this section with a couple of examples. In the first the *overall heat transfer coefficient* is introduced and in the second it is used to describe the heat flow through a hollow cylinder. The latter example will be of great help when modeling the temperature loss over the intercooler.

Example, Overall Heat Transfer Coefficient

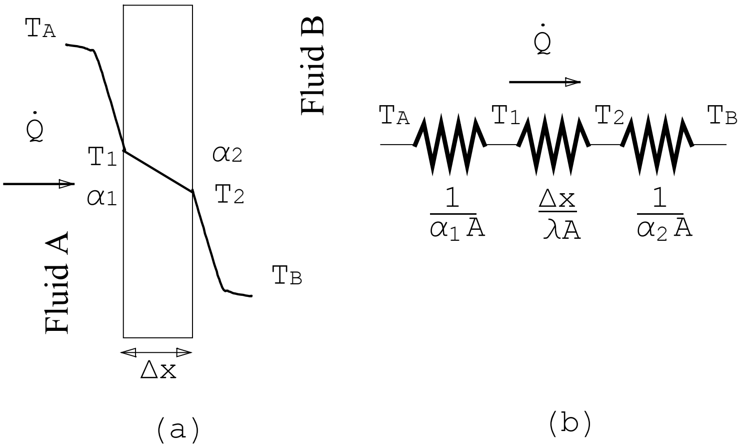


Figure 2.2. Heat flow through a plane wall.

Consider the plane wall shown in Figure 2.2 (a) exposed to a hot fluid A on one side and a cooler fluid B on the other side. The heat transfer is expressed by

$$\dot{Q} = \alpha_{m,1}\, A(T_A - T_1) = \frac{\lambda A}{\Delta x}\,(T_1 - T_2) = \alpha_{m,2}\, A(T_2 - T_B)$$

The heat transfer process may be represented by the resistance network presented in figure Figure 2.2 (b), and the overall heat transfer is given by

$$\dot{Q} = \frac{\Delta \mathrm{T}_{\mathrm{overall}}}{\Sigma \mathrm{R}_{\mathrm{th}}} = \frac{T_A - T_B}{\frac{1}{\alpha_{m,1}\, A} + \frac{\Delta x}{\lambda\, A} + \frac{1}{\alpha_{m,2}\, A}} = U\, A\, \Delta \mathrm{T}_{\mathrm{overall}}$$

where the overall heat transfer coefficient

$$\mathrm{U} = \frac{1}{\frac{1}{\alpha_{m,1}} + \frac{\Delta x}{\lambda} + \frac{1}{\alpha_{m,2}}} \tag{2.6}$$

has been introduced.

Example, Hollow Cylinder

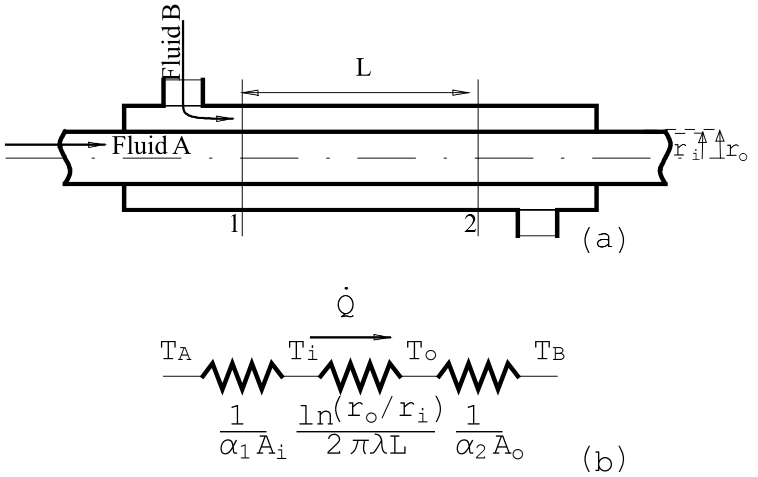


Figure 2.3. Heat flow through a hollow cylinder.

Consider a hollow cylinder as in Figure 2.3 (a), with a hot fluid A flowing inside and a cooler fluid B on the outside. Heat will flow from fluid A to the wall by convection, then by conduction through the wall, and finally by convection from the wall to fluid B. Suppose that the thickness of the wall separating the both fluids, $r_o - r_i$, is very small compared to the length L, then it might be assumed that heat flows only in the radial direction. The area for heat flow in a cylindrical system is

$$A_r = 2\pi r L$$

so Fouriers law (2.1) is written

$$\frac{\dot{Q}}{2\pi r L} = \frac{\dot{Q}}{A_r} = \dot{q} = -\lambda \frac{\mathrm{dT}}{\mathrm{dr}} \tag{2.7}$$

Let T_A and T_B denote the temperature of fluid A and B respectively. Using the method of separation of variables and integrating (2.7) yields

$$\int_{r_i}^{r_o} \frac{\dot{Q}}{A_r}\, dr = -2\pi\lambda L \int_{T_i}^{T_o} dT$$

$$\Longrightarrow$$

$$\dot{Q} = \frac{2\pi\lambda L\,(T_i - T_o)}{\ln\frac{r_o}{r_i}}$$

Hence the thermal resistance for a circular tube becomes

$$R_{\mathrm{th}} = \frac{\ln\frac{r_o}{r_i}}{2\pi\lambda L}$$

Note that the area for convection is not the same for both fluids in this case, these areas depend on the inside tube diameter and wall thickness. This gives the overall heat transfer:

$$\dot{Q} = \frac{T_A - T_B}{\frac{1}{\alpha_{m,i}\,A_i} + \frac{\ln(r_o/r_i)}{2\,\pi\,\lambda\,L} + \frac{1}{\alpha_{m,o}\,A_o}}$$

in accordance with the thermal network shown in Figure 2.3 (b). The terms A_i and A_o represent the inside and the outside surface areas of the inner tube respectively. The overall heat transfer coefficient may be based on either the inside, $U_i = \dot{Q}/(A_i\,\Delta T)$, or the outside area of the tube. Accordingly, if the inside is chosen

$$U_i = \frac{1}{\frac{1}{\alpha_{m,i}} + \frac{A_i\ln(r_o/r_i)}{2\,\pi\,\lambda\,L} + \frac{A_i}{A_o}\,\frac{1}{\alpha_{m,i}}} \tag{2.8}$$

□

2.3 Heat Transfer Efficiency

The ability of a heat exchanger to lower the temperature is characterized by a measure of efficiency, ϵ, which may be defined as:

$$\epsilon = \frac{\text{actual heat transfer}}{\text{maximum possible heat transfer}}$$

The maximum value of heat transfer could be attained if one of the fluids were to undergo a temperature change equal to the maximum temperature difference in the heat exchanger, which is the difference in entering temperatures for the hot and the cool fluid. If both the fluids present in the exchanger are air, conservation of energy requires that the fluid that undergoes this maximum temperature change is the one with the minimum value of the air mass flow rate.

$$\dot{Q}_{\text{max}} = (\dot{m}\,c_p)_{\text{min}}\,(T_{h_{\text{in}}} - T_{c_{\text{in}}})$$

where the subscripts h and c denote the hot and the cool fluid respectively. The actual heat transfer may be computed by calculating either the energy lost by the hot fluid or the energy gained by the cold fluid. If the former version is chosen and the warm fluid is the minimum fluid, the expression for ϵ becomes:

$$\epsilon = \frac{T_{h_{\text{in}}} - T_{h_{\text{out}}}}{T_{h_{\text{in}}} - T_{c_{\text{in}}}} \tag{2.9}$$

3. Fluid Mechanics

In addition to what was said about the thermodynamic aspects of fluid flow in Chapter 1, we are here going to deal with some further properties of the fluid flow in pipes. Of great importance for energy losses in pipe flow is to decide whether the fluid is flowing in parallel layers, i.e. laminar, or with superimposed eddies, i.e. turbulent. Which case is present is determined by the *Reynolds number* that is a dimensionless quantity defined as

$$\mathrm{Re} = \frac{\rho\, v\, d}{\mu}$$

where ρ, v, and μ is the density, the velocity, and the dynamic viscosity of the flowing fluid and d is the pipe diameter. When the Reynolds number exceeds a certain value, the flow turns from laminar to turbulent. The specific value of Re when this happens, called the critical Reynolds number, Re_{crit}, has to be determined experimentally for the geometry of the fluid flow. For flow in pipes though, turbulent flow can be assumed for Re>2300. The dynamic viscosity, μ, depends on both the temperature and the pressure of the fluid, however only the temperature dependence is considered here. For the temperature interval ranging from 220-400 *K*, which should cover the interesting temperatures, μ of dry air can be described by the relation

$$\mu = 2.3937\ 10^{-7}\ T^{0.7617} \tag{3.1}$$

Figure 3.1 shows the agreement with tabulated values.

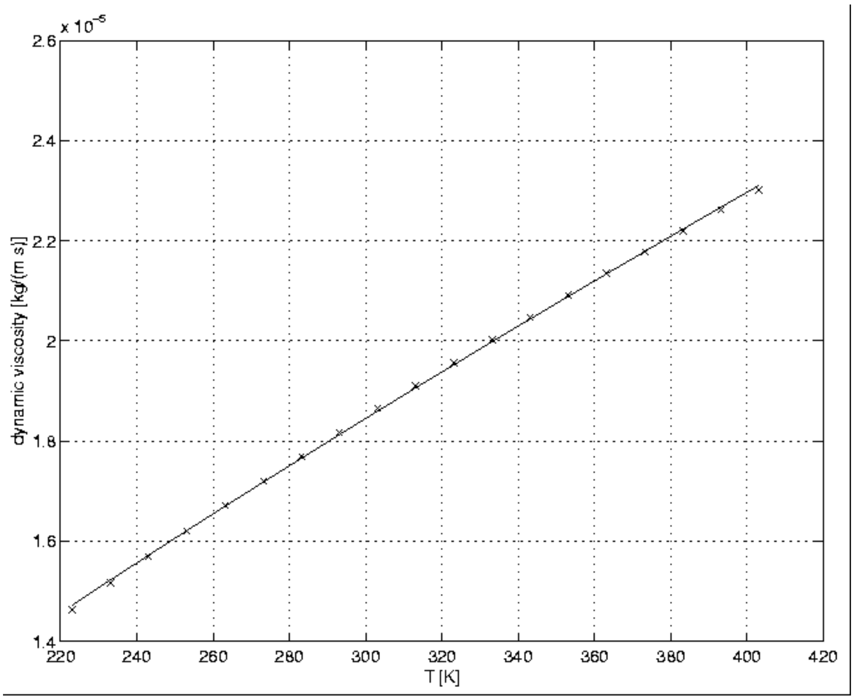


Figure 3.1. x: Tabulated values for the dynamic viscosity of dry air at atmospheric pressure plotted versus temperature. The solid line is the fitted model.

For a given pipe dimension, one can use this expression for μ together with the definition of *Re* to calculate the mass flow for which the flow becomes turbulent. The first equality of equation (1.10) relates the mass flow to the gas velocity

$$\dot{m} = \rho_1 \, v_1 \, A_1 \tag{3.2}$$

The conclusion is that the flow is turbulent if the following condition is fulfilled

$$\dot{m} > \mathrm{Re}_{\mathrm{crit}} \, \pi \, d \, \frac{\mu}{4} \tag{3.3}$$

For the case of pipes with a non-circular cross section area, the diameter, d, is replaced by the so called *hydraulic diameter, d_h*

$$d_h = \frac{4A}{S}$$

where A denotes the cross section area and S is the circumference.

3.1 Measuring of Properties in Fluid Flow

In this section methods of measuring pressure and temperature is briefly described. We will not give the thermodynamical aspects of why a property is measured in a certain way, but merely show the experimental setup required to measure the property of interest. For a more thorough explanation, see e.g. [Jones & Dugan]. The static pressure can be measured with a probe placed in a small hole in the wall as shown in Figure 3.2.

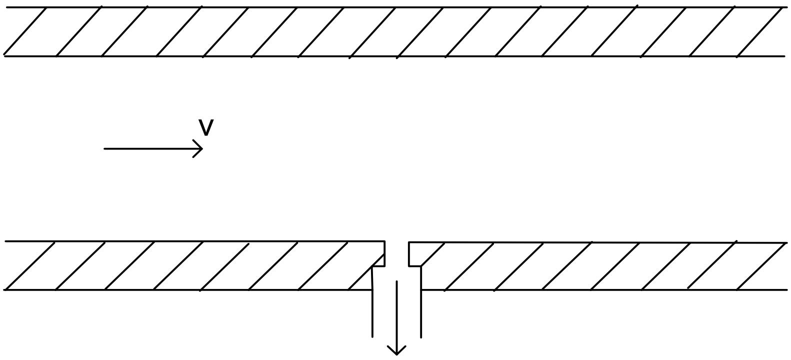


Figure 3.2. Measurement of static pressure.

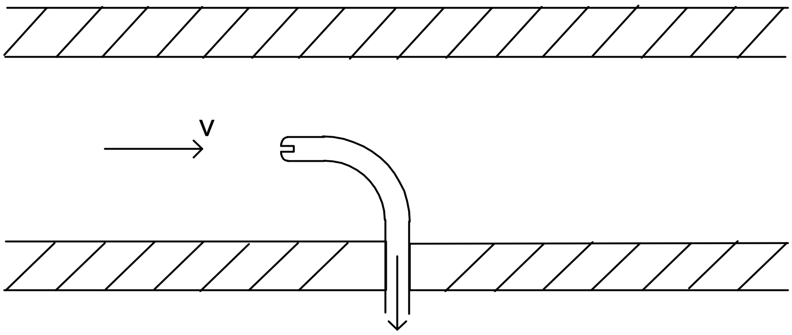


Figure 3.3. Measurement of total pressure.

The total pressure as defined by equation (1.16)

$$p_0 = p + \frac{\rho v^2}{2}$$

can be directly measured with a probe placed as in Figure 3.3. Measuring of static temperature is not discussed here, but the relation between the static and the total temperature is given by equation (1.17)

$$T_0 = T + \frac{v^2}{2c_p}$$

and the total temperature can be measured with a probe placed like in Figure 3.4. In this case it is essential that the probe is isolated from the wall of the tube.

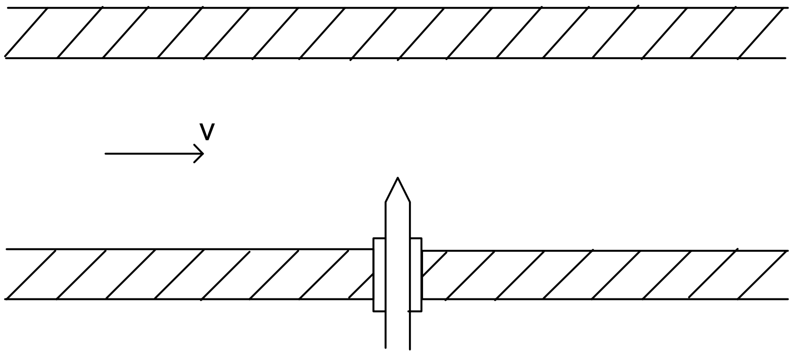


Figure 3.4. Measuring of total temperature.

From the definition of the total pressure it is seen that if only one of the pressures p and p_0 can be measured directly, the other can be calculated if the velocity and density of the flowing fluid is known. The same is valid for the temperatures T and T_0 with the additional request of a known value of c_p. The following example illustrates the difference between the static and total properties.

Example, Static and Total Properties

Assume that air, with the constant specific heat $c_p = 1.01$ [J/(kg K)], flows in a tube with cross sectional area $A = \pi \times 0.05^2$ [m^2]. The experimental setup allows for measurements of the static pressure p, the total temperature T_0 and the mass rate of flow $\dot{m}$. In this example the total pressure, p_0, and the static temperature T are calculated using experimental data. The expressions for the total pressure and temperature (1.16) and (1.17) can be rewritten in terms of measurable properties using equations (1.9) and (1.2):

$$p_0 = p + \frac{\rho v^2}{2} = p + \frac{\dot{m}^2}{2\rho A^2} = p + \frac{R\,T\,\dot{m}^2}{2A^2} \tag{3.4}$$

and

$$T_0 = T + \frac{v^2}{2c_{p,a}} = T + \frac{1}{2c_{p,a}}\left(\frac{R\dot{m}}{A\,p}\right)^2 T^2 \tag{3.5}$$

Solving equation (3.5) for T gives

$$T = \frac{p^2 A^2 c_p}{\dot{m}^2 R^2} \pm \sqrt{\left(\frac{p^2 A^2 c_p}{\dot{m}^2 R^2}\right)^2 + \frac{2\, p^2 A^2 c_p}{\dot{m}^2 R^2}\, T_0} \tag{3.6}$$

where the negative square root may be omitted. Using Equations (3.4) - (3.6) p_0 and T can be calculated, and the results are shown in Table 1 below. Note that the relative errors are very small for both pressure and temperature. E.g. for the first measurement we have

$$p_{rel,error} = \frac{p_0 - p}{p} = \frac{2.873}{117.374\, 10^3} \approx 2.44\, 10^{-5}$$

i.e. the relative error is less then 0.0025% for this case.

Comparing Static and Total properties.

Table 1

measured			calculated		deviation	
ṁ [kg/h]	p [kPa]	T_0 [K]	p_0 [kPa]	T [K]	p_0-p [Pa]	T_0-T [K]
79.3642	117.374	298.265	117.377	298.263	2.873	0.0021
116.758	132.929	299.625	132.935	299.622	5.5156	0.0035
150.338	137.65	301.378	137.659	301.372	8.8824	0.0055
173.981	138.027	302.977	138.039	302.97	11.9263	0.0074
203.267	137.245	304.438	137.262	304.428	16.4508	0.0104
221.238	137.62	305.613	137.639	305.601	19.5101	0.0123
242.361	137.745	307.073	137.768	307.058	23.5037	0.0149
272.363	137.787	308.532	137.817	308.513	29.8144	0.019
296.615	137.075	310.445	137.111	310.422	35.764	0.023

The deviations between the static and total properties are quite small and the error introduced when using static pressure instead of the total pressure may in most cases be neglected.

□

4. Modelica

This chapter gives an introduction to Modelica. Please refer to [Fritzon] for a more complete coverage.

Modelica is a modeling language that allows you to specify mathematical models of complex physical systems. It is based on equations instead of assignment statements, has multi-domain capability, it is an object-oriented language, and has constructs for creating and connecting components.

4.1 Creating Models

There are two different syntax for defining models in *MathModelica*, a plain Modelica and a *MathModelica* syntax. The former, called ModelicaInputForm, corresponds to the original syntax of the Modelica language, while the latter, MathModelicaInputForm, is closer to the syntax normally used in *Mathematica*. This syntax gives you additional benefits such as 2D-input and specialized symbols. To choose syntax is mainly a question about taste. This chapter uses the *MathModelica* syntax, however it is easy to convert between the two input forms, using the MathModelica Palette.

4.1.1 Model Structure

A *MathModelica* model contains three parts, name, declarations, and equations, with the following structure

```
Model[name, "comment"
  declarations;
 Equation[
  equations
 ]
]
```

The next example shows how a population of two species develops over time, when one species is dependent on the other. This model was first presented by V. Volterra in his work *Variations and Fluctuations of the Number of Individuals in Animal Species Living Together*. The model described by the following equations

```
x' = -a*x+b*x*y,  x(0) = 2
y' = c*x-d*x*y,   y(0) = 2
a = b = c = d = 1
```

is represented like this in *MathModelica*:

```
Model[Volterra,
 "Number of Individuals in Animal Species",
 Real x[{Start == 2}] "Number of predators";
 Real y[{Start == 2}] "Number of preys";
 Parameter Real {a == 1, b == 1};
 Parameter Real {c == 1, d == 1};
 Equation[
  x' == - a x + b x y;
  y' == c y - d x y;
 ]
]
```

Thus the name of the model is Volterra and in the declaration part x and y are declared to be of the type Real with the initial value 2. Furthermore a, b, c, and d are declared to be of type real, but with the restriction that they are parameters, i.e., they are not time dependent. A parameter is constant during a simulation, but may be changed between simulations. To learn more about available types please refer to Section 4.3.2. Comments can also be included within citation marks in the model. Finally the model equations are stated. Note that these are *true* equations and not assignments. The equations do not specify which variables are inputs and which are outputs, whereas in assignment statements variables on the left-hand side are always outputs (results) and variables on the right-hand side are always inputs. Thus, the causality of equation-based models is unspecified and becomes fixed only when the corresponding equation systems are solved. This is called non-causal modeling. The term physical modeling reflects the fact that acausal modeling is very well suited for representing the physical structure of modeled systems.

4.1.2 Types

Modelica has built-in “primitive” data types to support floating-point, integer, boolean, and string values. These primitive types contain data that *Modelica* understands directly. The type of every variable must be typed explicitly. The primitive data types of *Modelica* are listed in Table **2**.

Type	**Description**
`Boolean`	either true or false
`Integer`	corresponding to the `C int` data type, usually 32 – bit two' s complement
`Real`	corresponding to the `C double` data type, usually 64 – bit floating – point
`String`	string of 8 – bit characters

Predefined data types in Modelica
Table 2

Defining a Type

It is possible to define new types

```
Type[name, annotation, type]
```

An example is to define a temperature measured in Kelvin, K, which is of type Real with the minimum value zero;

```
Type[Temperature, "temperature measured in Kelvin",
  Real[{Unit == "K", Min == 0}]];
```

The attributes that can be set are listed in Modelica syntax in Table 3.

Type	**Attribute**	**Default**	**Description**
Real	quantity		
	unit		Unit used in equations
	displayUnit		Default display unit
	min	-Inf	Inf denotes a large value
	max	+ Inf	
	start	0	Initial value
	fixed	true (for parameter / constant)	
		false (for other varaibles)	
	nominal		
Integer	quantity		
	min	-Inf	Inf denotes a large value
	max	+ Inf	
	start	0	Initial value
	fixed	true (for parameter / constant)	
		false (for other variables)	
Boolean	quantity		
	start	false	Initial value
	fixed	true (for parameter / constant)	
		false (for other varaibles)	
String	quantity		
	start		Initial value

Attributes of predefined types.
Table 3

4.1.3 Constants

Named constants in *Modelica* are created by using the prefixes Constant and Parameter in declarations, and providing a declaration equation as part of the declaration, e.g.

```
Constant Real c0 == 2.99792458 10^8;
Constant String redcolor == "red";
Constant Integer population == 1234;
Parameter Real speed == 25;
```

Opposite to constants parameters can be declared as input to the model, thus Parameter can be declared without a declaration equation. For example

```
Parameter Real {mass, velocity};
```

4.1.4 Connectors

Modelica connectors defines the variables that are part of the communication interface. Thus, connectors specify external interfaces for interaction.
Variables in a connector can be of two types: non-flow (default), or flow declared using the prefix Flow. Below is an example of a Flange connector used to connect one dimensional rotational bodies:

```
Connector[Flange, "1D rotational flange",
  SIunits.Angle phi
   "Absolute rotation angle of flange";
  Flow SIunits.Torque tau "Cut torque in the flange"
 ];
```

4.1.5 Connections

Connections between components can be established between connectors of equivalent type. *MathModelica* supports equation-based acausal connections, which means that connections are realized as equations. For acausal connections, the direction of data flow in the connection need not be known. Additionally, causal connections can be established by connecting a connector with an input attribute to a connector declared as output. This means that the flow should go from the output to the input connector.

Two types of coupling can be established by connections depending on whether the variables in the connected connectors are non-flow (default), or declared using the prefix Flow:

- Equality coupling, for non-flow variables.

- Sum-to-zero coupling, for flow variables, analogous to Kirchhoff's current law.

For example, define an electrical pin

```
Connector[Pin, "Pin of an electrical component",
 Modelica.SIunits.Voltage v "Potential at the pin";
 Flow Modelica.SIunits.Current i
  "Current flowing into the pin";
]
```

the keyword `flow` for the variable `i` of type `Current` in the `Pin` connector class indicates that all currents in connected pins are summed to zero, according to Kirchhoff's current law.

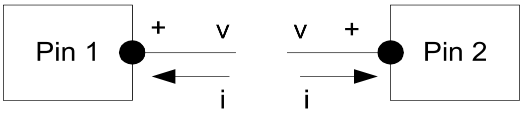


Figure 4.1. Connecting two components that have electrical pins.

Connection statements are used to connect instances of connection classes. A connection statement

```
Connect[Pin1, Pin2]
```

with `Pin1` and `Pin2` of connector class `Pin`, connects the two pins so that they form one node. This produces two equations, namely:

```
Pin1.v == Pin2.v;
Pin1.i + Pin2.i == 0;
```

The first equation says that the voltages of the connected wire ends are the same. The second equation corresponds to Kirchhoff's current law saying that the currents sum to zero at a node (assuming positive value while flowing into the component). The sum-to-zero equations are generated when the prefix `Flow` is used. Similar laws apply to flows in piping networks and to forces and torques in mechanical systems.

4.1.6 Inheritance

It is possible to inherit from a class to extend the behavior and properties of that class. The original class, known as the superclass, is extended to create a more specialized version, known as the subclass. In this process, the behavior and properties of the original class in the form of field declarations, equations, and other contents is reused, or inherited, by the subclass.

A superclass can be declared as Partial, this suggests that the class is not complete. For example consider the following model for an electrical oneport

```
Partial
 Model[OnePort,
  "Component with two electrical pins p and n
    and current i from p to n",
  Modelica.SIunits.Voltage v
   "Voltage drop between the two pins (= p.v - n.v)";
  Modelica.SIunits.Current i
   "Current flowing from pin p to pin n";
  Modelica.Electrical.Analog.Interfaces.PositivePin p;
  Modelica.Electrical.Analog.Interfaces.NegativePin n;
  Equation[
   v == p.v - n.v;
   0 == p.i + n.i;
   i == p.i
  ]
 ]
```

This model might be used as a superclass for electrical oneports, such as resistors, inductors, and capacitors. However it needs to be extended with an equation that states the relation between the current and voltage. For a resistor this would be the well-known ohms law, v = R i, so this may be implemented as follows

```
Model[Resistor, "Ideal linear electrical resistor",
 Extends[Modelica.Electrical.Analog.Interfaces.
   OnePort];
 Parameter Modelica.SIunits.Resistance R == 1
  "Resistance";
 Equation[
  R i == v
 ]
]
```

4.1.7 Using Predefined Components

It is possible to use predefined components either by referring to them by their full path, e.g.

```
Model[Foo,
 Modelica.Mechanics.Translational.Spring spring1;
 Modelica.Mechanics.Translational.Damper damper1;
   ...
  Equations[
   Connect[spring1.flange␣b, damper1.flange␣a];
    ...
  ]
]
```

or by loading a package first with ModelicaImportAll to have direct access to desired components:

```
ModelicaImportAll[Modelica.Mechanics.Translational];

Model[Foo,
 Spring spring1;
 Damper damper1;
   ...
  Equations[
   Connect[spring1.flange␣b, damper1.flange␣a];
    ...
  ]
]
```

5. Base Classes

In this chapter the connectors and partial models used for the engine library are presented. The model structure is shown in Figure 5.1, where the connectors have been denoted with circles and partial models are dashed.

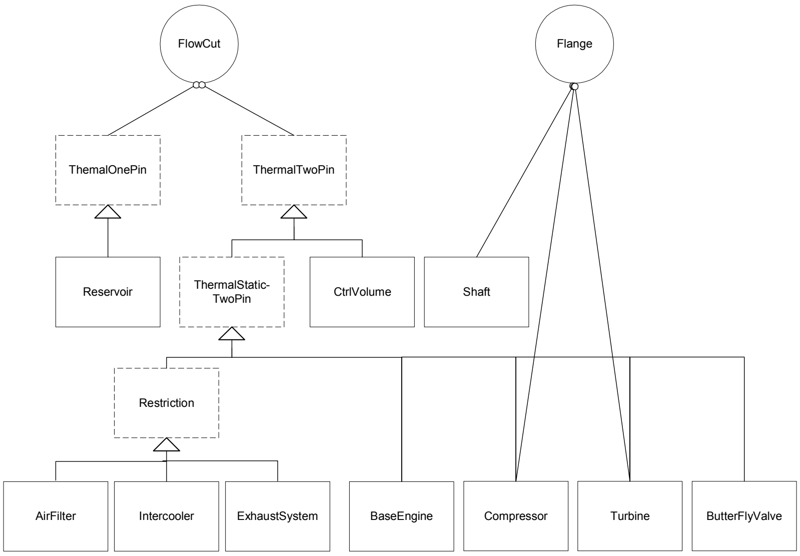


Figure 5.1. Diagram of the model structure. Partial classes are dashed, and connectors are shown in circles.

5.1 LinearRoot

Before connectors and base classes are presented a root function is introduced. To avoid numerical problems with root equations a function, called LinearRoot, is introduced. This function takes the input, x, power, p, and linearization limit, dx, as arguments. Whenever $x >$ dx the function returns x^p, but if $x < -$dx it will return $-(-x)^p$. For the interval abs(x) < dx a third order polynomial that makes LinearRoot a C^1-function, see Figure 52, is used to calculate the return value.

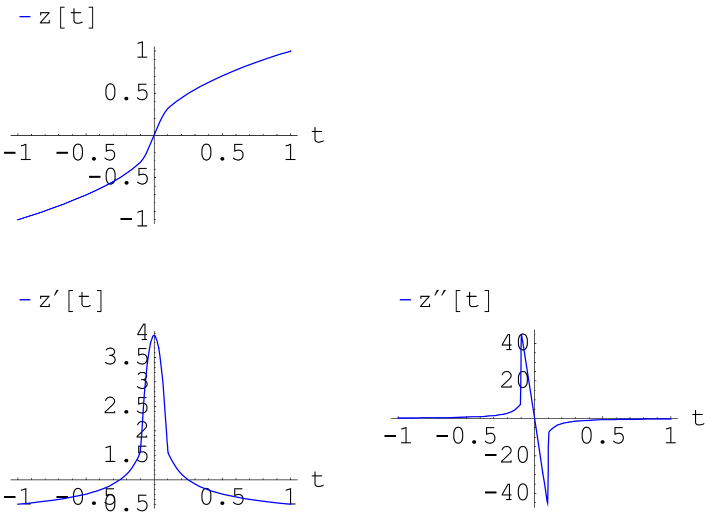


Figure 5.2. Upper left; z=$\sqrt{t}$, linearized for |dt| < 0.1. Lower left; derivative, z'. Lower right; second order derivative z''.

MathModelica Model

```
function LinearRoot
   input Real x "input x";
   input Real p "power";
   input Real dx "linearization limit";
   output Real y "output";

protected
   Real Gamma;
   Real C3;
   Real C1;
   Real adx;
   Real padx;

algorithm
   adx := abs(dx);
   if x > adx then
      y := x^p;
   else
      if x < -adx then
         y := -(-x)^p;
      else
         padx := adx^(p-1);
         C1 := (1.5-0.5*p)*padx;
         C3 := 0.5*(p-1)*padx/(adx*adx);
         y := (C1+C3*x*x)*x;
      end if;
   end if;
end LinearRoot;
```

5.2 Types

The following types are defined to be used in the models. Note that the units are not always SI units, this is done to avoid that the model becomes ill conditioned. For example, if SI units would be used, pressures would be of the magnitude 10^6 [Pa], while air mass flow would be of the magnitude 10^{-2} [kg/s].

```
Type[Angle = Real[{Unit == "rad"}]];
Type[AngularVelocity = Real[{Unit == "rad/s"}]];
Type[EnergyFlow = Real[{Unit == "J/s"}]];
Type[Inertia = Real[{Unit == "kg m2"}]];
Type[Length = Real[{Unit == "m"}]];
Type[Mass = Real[{Unit == "g", Min == 0}]];
Type[
  MassFlow = Real[{Unit == "g/s", Min == -300, Max == 300}]];
Type[Pressure = Real[{Unit == "kPa", Min == 10}]];
Type[RotationalSpeed = Real[{Unit == "rps"}]];
Type[Speed = Real[{Unit == "m/s"}]];
Type[Temperature = Real[{Unit == "K", Min == 220}]];
Type[Volume = Real[{Unit == "l", Min == 0}]];
Type[Torque = Real[{Unit == "Nm"}]];
```

5.3 Thermofluid Connector, FlowCut

There are three basic properties that have to be communicated between two components for fluid flow; pressure, mass flow, and energy or temperature. Mass flow and energy are flow variables, while pressure and temperature are non flow variables.

The energy transmitted at a connector is then given by multiplying (1.14) with the air mass flow

$$-\dot{W} = \dot{m}_{\text{air}}\,\Delta\text{h}$$

Furthermore the change of enthalpy is a function of the temperature only for monatomic gases (1.7) and (1.8). This is also a good approximation for air, i.e. we can consider the transmitted energy to be a function of temperature only. Thus it is possible to use either temperature or energy in the connector.

However we must know from which component the temperature or energy should be taken. If the flow is from left to right the left component should give the temperature (or energy), otherwise it should be given by the right component. To solve this problem a connector that uses pressure, *p*, air mass flow, m_{dot}, energy flow, Q_{dot}, and temperature, *T*, is used for thermofluid. In Section 5.5 it is shown how this redundancy can be used to solve the problem with selection of temperature.

MathModelica Models

There are two thermofluid connectors, FlowCut_aa and FlowCut_bb, which are completely identical. There is only a difference in icons, in order to tell the difference between them in a connection diagram.

```
Connector[FlowCut_aa,
 ■;
 ■;
 Temperature T;
 Pressure p;
 Flow MassFlow m_dot;
 Flow EnergyFlow Q_dot
]
```

```
Connector[FlowCut_bb,
 ■;
 ■;
 Temperature T;
 Pressure p;
 Flow MassFlow m_dot;
 Flow EnergyFlow Q_dot
]
```

5.3.1 Alternative Approach

Another idea is to use a pWH-connector instead of a pWHT-connector, but this approach needs an If clause without an else branch to work, which is not included in the Modelica language (a solution to this, using MathModelicas pre processing capability, is presented by [Silverlind] in his forthcoming Master Thesis). This approach seems to bee the most appealing. With the current pWHT-connector there might be problems when connecting different components. E.g. if two restrictions are to be connected parallel to each other, they must have a volume between them both at inlet and outlet to handle the difference in temperature that might result. For a pWH connector this would not be a problem as the temperature does not appear in the connector. In this case the temperature would be an internal signal calculated from mass and energy flow for each component. The redundancy in the connector may also result in difficulties for the simulation engine to find a proper solution to simulations.

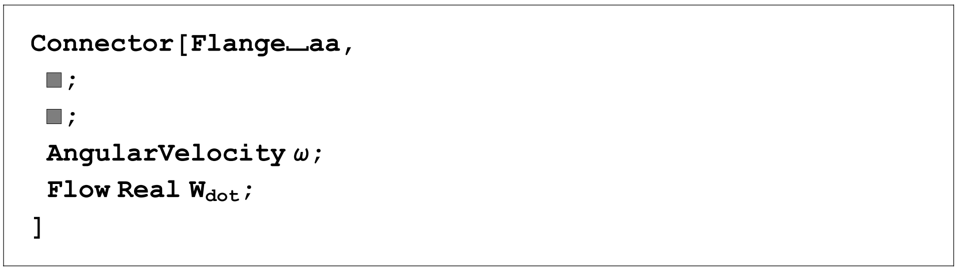
Another interesting area to look into is multi component flows. For this it could be a good idea to use the Thermofluid library that is under development by Hubertus Tummescheit and Jonas Eborn. The interested reader should refer to [Tummescheit & Eborn].

5.4 Rotational Mechanics Connector, Flange

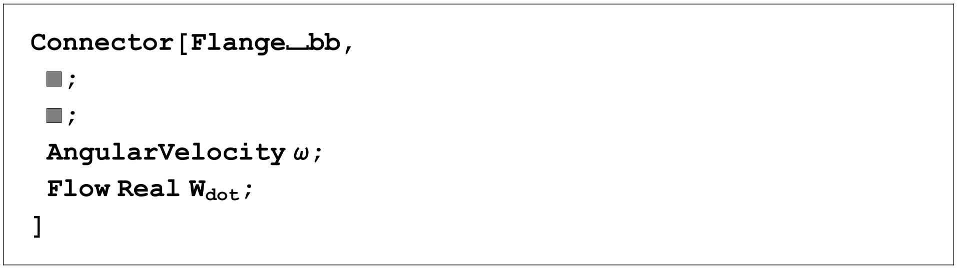
Modelicas rotational mechanics library uses a connector, called flange, with the flow variable tau (τ), cut torque in the flange, and the non flow variable phi (φ), absolute rotation angle of flange. However if this connector is used in the MVEM the simulation engine has problems to find the initial values. Therefore a new connector which uses the energy flow, W_{dot}, and the angular velocity ω, is defined. This gives a better selection of state variables and thus making it easier for the simulation engine to find a solution.

MathModelica Models

There are two thermofluid connectors, Flange_aa and Flange_bb, which are completely identical. There is only a difference in icons, in order to tell the difference between them in a connection diagram.



```
Connector[Flange_aa,
 ■;
 ■;
 AngularVelocity ω;
 Flow Real Wdot;
]
```



```
Connector[Flange_bb,
 ■;
 ■;
 AngularVelocity ω;
 Flow Real Wdot;
]
```

5.5 Thermofluid One Pin and Two Pins

All models presented, except the shaft model, work with a fluid flow (see Figure 5.1). This means that some of the properties of these models are the same. Therefore two base classes are developed, one for thermofluid with one inlet, and one for two inlets, from which the models will inherit their common properties. Note that the gas constant *R*, and the specific heat capacity, c_p, are assumed to be constant. This is a good assumption at the inlet, where the temperature interval is limited, but for the base engine and outlet side this is not the case. Nevertheless the assumption is used there also for simplicity.

Suppose we have two components connected together. The temperature at the connector is given by the restriction from which the air is flowing outwards. In other words the energy at the connector is a function of this temperature

$$q_1 = \begin{cases} w_{res}\, c_p\, T_{res} & \text{if } w_{res} < 0 \\ w_{res}\, c_p\, T_1 & \text{else} \end{cases} \tag{5.1}$$

Figure 5.3 illustrates the energy flow.

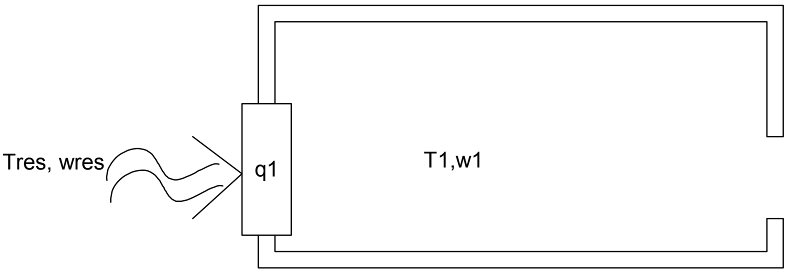
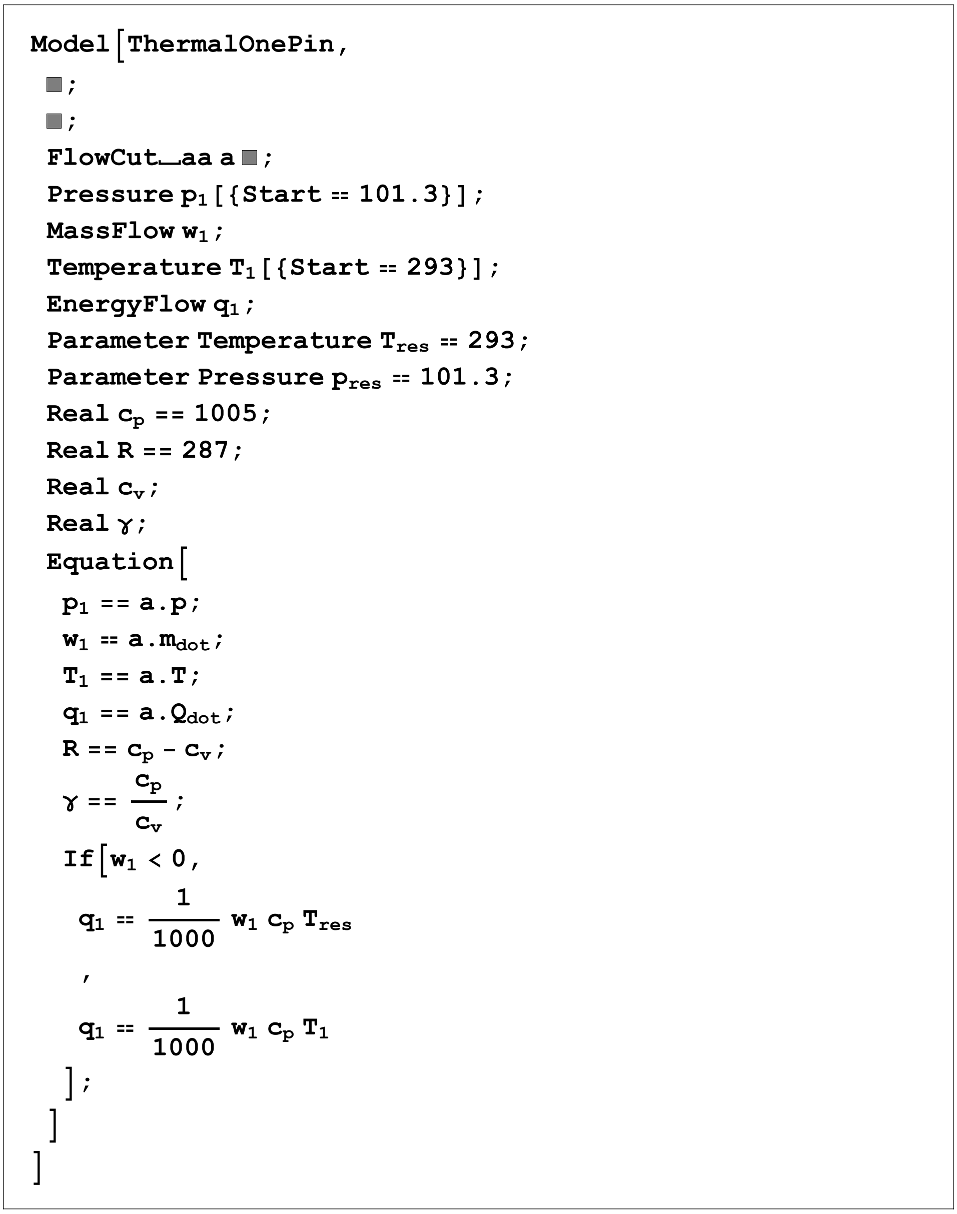


Figure 5.3. *Temperature at a connector. Consider an open window, wind blows from the outside, $w_{res} < 0$, with temperature T_{res} into a room with temperature T_2. The temperature at the window, or connector, will be the same as the outside temperature.*

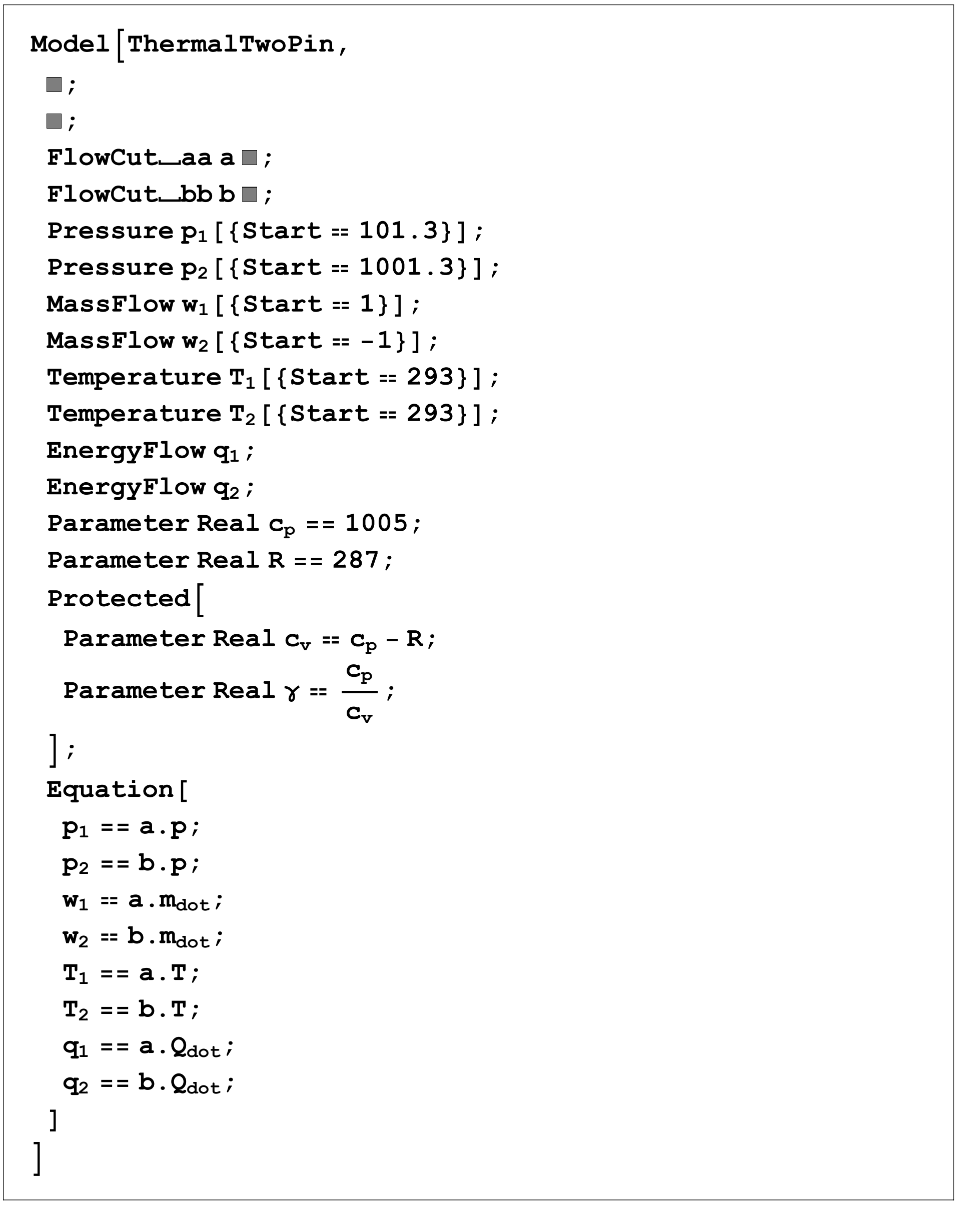
MathModelica Model

A ThermalOnePin is used as a base class for thermofluid components with just one connector. The energy at the connector is given according to equation (5.1).



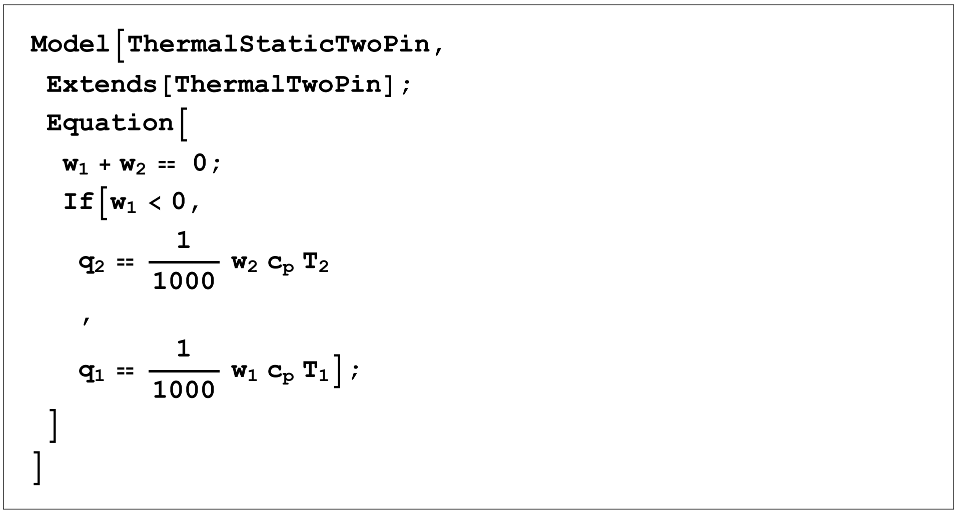
```
Model[ThermalOnePin,
 ■;
 ■;
 FlowCut␣aa a ■;
 Pressure p1[{Start == 101.3}];
 MassFlow w1;
 Temperature T1[{Start == 293}];
 EnergyFlow q1;
 Parameter Temperature Tres == 293;
 Parameter Pressure pres == 101.3;
 Real cp == 1005;
 Real R == 287;
 Real cv;
 Real γ;
 Equation[
  p1 == a.p;
  w1 == a.mdot;
  T1 == a.T;
  q1 == a.Qdot;
  R == cp - cv;
        cp
  γ == ---- ;
        cv
  If[w1 < 0,
           1
   q1 == ------ w1 cp Tres
          1000
   ,
           1
   q1 == ------ w1 cp T1
          1000
  ];
 ]
]
```

ThermalTwoPin is used for all thermofluid components with two connectors. The equations for energy flow depend on if the component is static ($w_1 + w_2 = 0$) or dynamic ($w_1 + w_2 \neq 0$), therefore this equations are added later on.



```
Model[ThermalTwoPin,
  ■;
  ■;
  FlowCut⎵aa a ■;
  FlowCut⎵bb b ■;
  Pressure p1[{Start == 101.3}];
  Pressure p2[{Start == 1001.3}];
  MassFlow w1[{Start == 1}];
  MassFlow w2[{Start == -1}];
  Temperature T1[{Start == 293}];
  Temperature T2[{Start == 293}];
  EnergyFlow q1;
  EnergyFlow q2;
  Parameter Real cp == 1005;
  Parameter Real R == 287;
  Protected[
   Parameter Real cv == cp - R;
                         cp
   Parameter Real γ == ── ;
                         cv
  ];
  Equation[
   p1 == a.p;
   p2 == b.p;
   w1 == a.mdot;
   w2 == b.mdot;
   T1 == a.T;
   T2 == b.T;
   q1 == a.Qdot;
   q2 == b.Qdot;
  ]
]
```

For static thermofluid two pins, i.e. $w_1 + w_2 = 0$, the ThermalStaticTwoPin is used. When the flow is going out at connector a, i.e. $w_1 > 0$, the energy flow at the connector will be given by $q_1 = w_1\, c_p\, T_1$, otherwise it will be given by the component connected to it. As the flow, w_2, at connector b will always be opposite to the flow, w_1, at connector a the energy flow at this connector will always be given from the component it self whenever $w_2 > 0$, i.e. $w_1 < 0$, otherwise it will be given by the component connected to it.

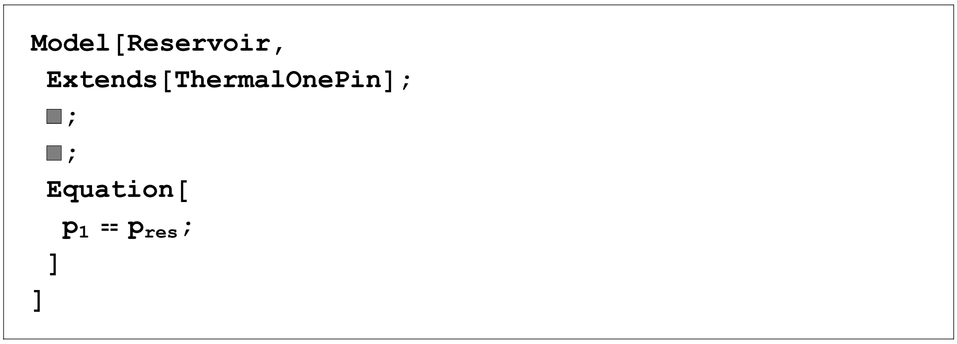


```
Model[ThermalStaticTwoPin,
 Extends[ThermalTwoPin];
 Equation[
  w1 + w2 == 0;
  If[w1 < 0,
            1
   q2 == ──────  w2 cp T2
          1000
   ,
            1
   q1 == ──────  w1 cp T1];
          1000
 ]
]
```

5.5.1 Reservoir

In a reservoir the pressure and temperature is held constant. Note however that locally at the connector the temperature may be given by the component that is connected to the reservoir (see discussion above).

MathModelica Model



```
Model[Reservoir,
 Extends[ThermalOnePin];
 ■;
 ■;
 Equation[
  p1 == pres;
 ]
]
```

5.6 Restriction

The air filter, exhaust system and intercooler work as restrictions for the air flow. Over a restriction the air mass flow is given by the pressure difference between outlet and inlet.

Consider steady and one dimensional fluid flow through a pipe section. Assume adiabatic flow, i.e. there is no heat transfer to or from the fluid, for simplicity assume there are no energy losses. Then the first law of thermodynamics (1.14) becomes

$$0 = h_{02} - h_{01}$$

Using equation (1.8) and (1.2) this expression can be rewritten as follows:

$$\begin{aligned} h_{02} - h_{01} &= \Delta \mathrm{h}_0 = [\Delta \mathrm{h} = c_p\, \Delta \mathrm{T}] = c_p\, \Delta\, T_0 = \\ &= \left[p = \rho\, R\, T \Longrightarrow T = \tfrac{p}{\rho\, R}\right] = c_p\, \Delta\left(\tfrac{p_0}{\rho\, R}\right) = \\ &= c_p\left(\tfrac{p_{02}}{\rho_2} - \tfrac{p_{01}}{\rho_1}\right) = 0 \qquad \Longrightarrow \tfrac{p_{02}}{\rho_2} = \tfrac{p_{01}}{\rho_1} \end{aligned}$$

By restricting the flow to be incompressible, so that $\rho_1 = \rho_2 = \rho$ and expanding the total pressure in its static and dynamic components the equation above takes the form

$$\frac{p_{01}}{\rho_1} = \frac{p_{02}}{\rho_2} \Longrightarrow p_{01} = p_{02} \Longrightarrow p_1 + \frac{\rho\, v_1{}^2}{2} = p_2 + \frac{\rho\, v_2{}^2}{2} \tag{5.2}$$

The above equation states that if there are no energy losses in the fluid flow the total pressure will be conserved. However, some of the energy is lost along the path of the flow due to friction, sudden area changes or bends. The energy lost is expressed as a loss in the total pressure and is denoted $\Delta \mathrm{p}_f$. Hence, equation (6.2) changes to

$$p_{01} = p_{02} + \Delta \mathrm{p}_f \Longrightarrow p_1 + \frac{\rho\, v_1{}^2}{2} = p_2 + \frac{\rho\, v_2{}^2}{2} + \Delta \mathrm{p}_f \tag{5.3}$$

Another useful relationship for fluid flow is the dynamic equation for steady one dimensional flow, which relates the resultant force on a mass element in the flow to the change in velocity.

$$F = \dot{m}\, \Delta \mathrm{v} = \dot{m}\, v_2 - \dot{m}\, v_1 \tag{5.4}$$

The example below shows how the expressions in this section together with the continuity equation can be used to calculate the loss in total pressure for a sudden change in area.

Example, Area Change

Consider Figure 6.4. The fluid is flowing with velocity v_1 in the narrow section of the pipe, with cross section area A_1, until it reaches the wide section with area A_2 where the velocity changes to v_2. Due to turbulence at the entrance of the wide section some of the energy of the fluid will be lost. In this example the energy loss will be calculated in terms of loss in total pressure. The fluid flow is considered to be steady, one dimensional and incompressible.

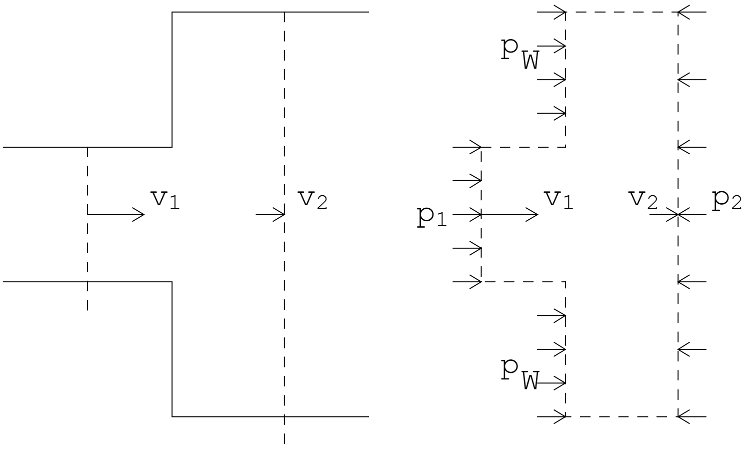


Figure 5.4. Sudden area increase.

Equation (6.3) gives an expression for the pressure loss

$$\Delta \mathrm{p}_f = p_1 - p_2 + \tfrac{\rho}{2}\,({v_1}^2 - {v_2}^2) \tag{5.5}$$

The resultant force acting on the mass segment indicated in Figure 6.4, is determined by studying the pressure in the different sections as illustrated in the right part of the figure. Equation (6.4) yields,

$$p_W(A_2 - A_1) + p_1\,A_1 - p_2\,A_2 = \dot{m}(v_2 - v_1) \tag{5.6}$$

Assume that $p_W = p_1$. This assumption occurs frequently in the literature however it is not entirely valid but it simplifies the derivation of the pressure loss. This simplifies the above expression to

$$(p_1 - p_2)\,A_2 = \dot{m}\ (V_2 - V_1) \tag{5.7}$$

Combine equations (6.5), (6.7) and (1.11)

$$\begin{aligned} \Delta \mathrm{p}_f &= \tfrac{\dot{m}}{A_2}\,(v_2 - v_1) + \tfrac{\rho}{2}\,({v_1}^2 - {v_2}^2) = \left[\tfrac{\dot{m}}{A_2} = \rho\,v_2\right] = \rho\,v_2(v_2 - v_1) + \tfrac{\rho}{2}\,({v_1}^2 - {v_2}^2) = \\ &= \tfrac{\rho}{2}\,({v_2}^2 - 2\,v_1\,v_2 + {v_1}^2) = \tfrac{\rho}{2}\,\left(v_1\left(1 - \tfrac{v_2}{v_1}\right)\right)^2 = \\ &= \left[\rho\,A_1\,v_1 = \rho\,A_2\,v_2 \Longrightarrow \tfrac{v_2}{v_1} = \tfrac{A_1}{A_2}\right] = \tfrac{\rho}{2}\,{v_1}^2\left(1 - \tfrac{A_1}{A_2}\right)^2 = K\,\rho\,{v_1}^2 \end{aligned}$$

where K is a constant. The above result can be expressed in terms of the mass flow. Again make use of equation (1.11) together with the ideal gas equation of state (1.2).

$$\begin{aligned} &K\,\rho\,{v_1}^2 = \left[v_1 = \tfrac{\dot{m}}{\rho_1\,A_1} = \tfrac{\dot{m}}{\rho_2\,A_2}\right] = \\ &\quad K\,\rho\left(\tfrac{\dot{m}}{\rho\,A_1}\right)^2 = \tfrac{K}{{A_1}^2}\,\tfrac{\dot{m}^2}{\rho} = \left[\rho = \rho_1 = \tfrac{p_1}{R\,T_1}\right] = \tfrac{K}{{A_1}^2}\,\dot{m}^2\,R\,\tfrac{T_1}{p_1} = \xi\,R\,T_1\,\tfrac{\dot{m}^2}{p_1} \end{aligned}$$

That is

$$\Delta \mathrm{p}_f = \xi R T_1 \frac{\dot{m}^2}{p_1} \tag{5.8}$$

where ξ is constant.

□

In the fluid mechanics literature, e.g. [Nakayama and Boucher, 1999], it is shown that similar expressions as the above apply for bends and valves. Usually the relationship is stated

$$\Delta \mathrm{p}_f = \xi\,(\mathrm{Re})\,\mathrm{RT}_1 \frac{\dot{m}^2}{p_1}$$

where *Re* is the Reynolds number and characterizes the influence of the frictional and internal forces on the flow field. As the ξ dependence on *Re* is normally quite small the same equation, (6.8), as for a sudden area change might be used.

MathModelica Model

```
Model[Restriction,
 Extends[ThermalStaticTwoPin];
 ■;
 ■;
 Parameter Real ξ == 8;
 Equation[
  If[w1 < 0,
   w2 LinearRoot[ξ R T2, 0.5, 0.01] ==
    LinearRoot[10^7 p2 (p2 - p1), 0.5, 0.01]
   ,
   w1 LinearRoot[ξ R T1, 0.5, 0.01] ==
    LinearRoot[10^7 p1 (p1 - p2), 0.5, 0.01]
  ]
 ]
]
```

6. Engine Models

In this chapter the engine components in the library are presented. The parameters in the simulation are only tuned so that they give a qualitative description of the actual behavior, but not a quantitative. The parameters and constants for those models that have been validated and quantitatively tested in [Bergström & Brugård, 2000], [Pettersson, 2000], and [Eriksson et.al., 2001], but they have all been changed after validation and furthermore all plots of validation data are biased, so the characteristics of the actual engine will not be reviled, as it is confidential.

A connection diagram for the complete engine is shown in Figure 6.1. The engine consists of an air filter, compressor, intercooler, throttle, base engine, turbine, exhaust system, and a shaft that connects the compressor and turbine to each other. At the inlet and outlet of the engine reservoirs, which corresponds to the ambient, are connected. The throttle is governed by an electric signal, and engine speed is given as an input to the base engine.

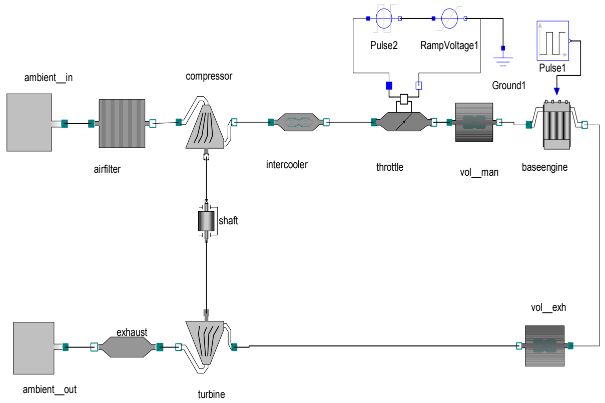


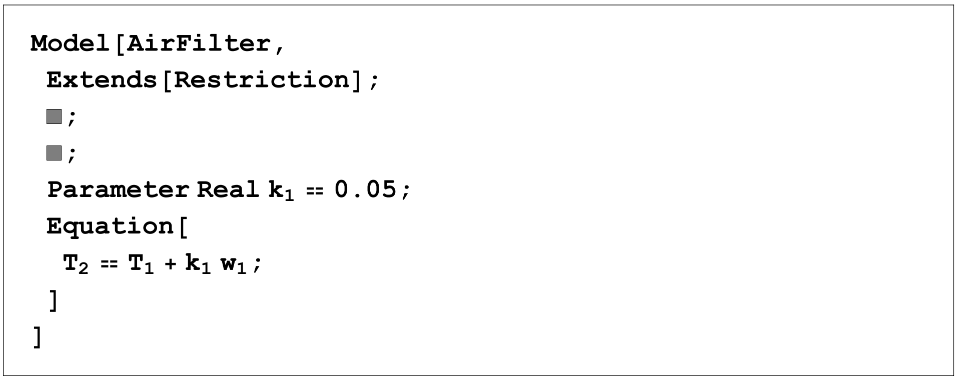
Figure 6.1. Engine diagram including a manifold volume and one exhaust volume.

When air enters the system it first passes an air filter that takes care of some pollutants, such as dust, to protect the engine. After this the air reaches the compressor where pressure is increased, resulting in increased density. A larger amount of air may thus enter the cylinders at each cycle, resulting in higher power output. However, the effect of the pressure increase is partly counteracted by an increase of temperature in the compressor. More importantly this temperature increase may also cause knock during combustion. To decrease the temperature, the air passes through a heat exchanger, the intercooler, before it passes the throttle. The throttle is used to govern the amount of air that should enter the cylinders, where the actual work is done in a four stroke cycle, described in Section 6.5. After the combustion the exhaust gases are used to propel the turbine, which supplies the compressor with power through the turbine shaft. Finally the exhaust gases pass through the exhaust system, consisting of exhaust pipes, silencer, and catalyst.

6.1 Air Filter

The air filter cleans the air from some pollutants, such as dust particles. It has the effect of decreasing the pressure of the fluid and also slightly decreases the temperature. The air filter consists of two sudden changes in area, the filter itself, and pipes in and out of it, i.e. the pressure model could be inherited from the Restriction class. A simple temperature model is added, which states that the temperature loss is proportional to the air mass flow.

MathModelica Model



```
Model[AirFilter,
 Extends[Restriction];
 ■;
 ■;
 Parameter Real k1 == 0.05;
 Equation[
  T2 == T1 + k1 w1;
 ]
]
```

6.1.1 Validation

A least squares fit to measured data, gives the constant ξ for equation (6.8). In Figure 6.2 the model is represented by a solid line, with validation data superimposed. As can be clearly seen in the figure this model describes the pressure loss very accurately; the average error is 0.05 kPa and the maximum error is less then 0.16 kPa. However it should be noted that we cannot manipulate ambient temperature, pressure nor humidity at our test bench, consequently there has been no possibility to validate the model for different ambient conditions, nor to make a proper temperature model.

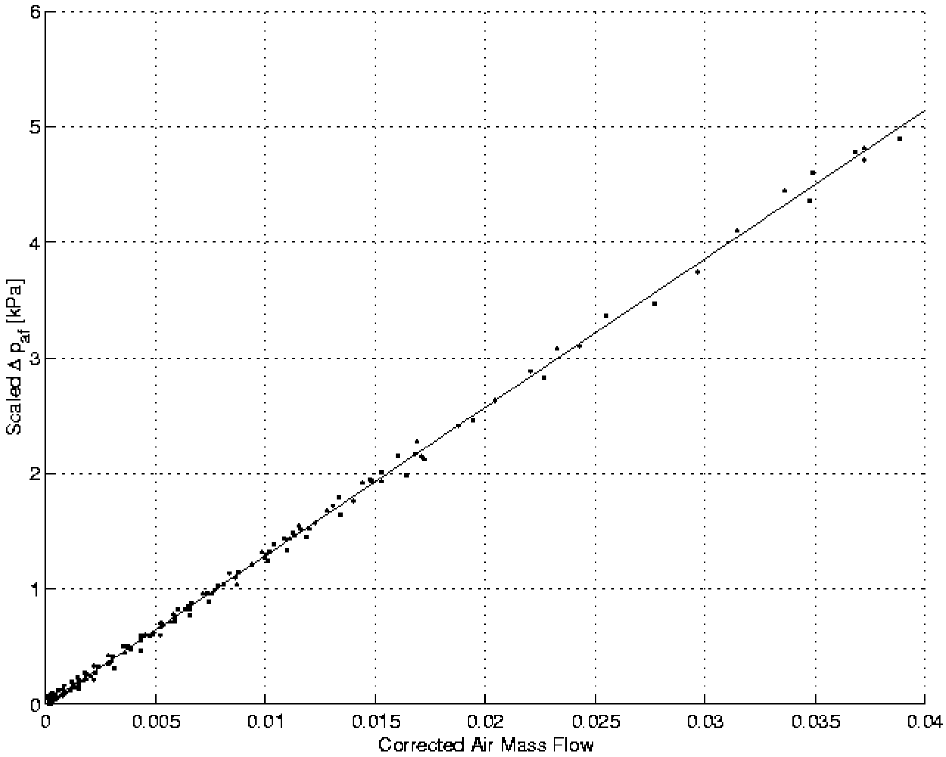


Figure 6.2. Validation of pressure loss over air filter, the corrected air mass flow is given by $T_{amb}\,\dot{m}_{air}^2$ / p_{amb}. Note that the pressure loss has been slightly scaled so that it will not reveal pressure loss in the particular engine.

Independence of Flow Direction

Simulations that shows that the model works independently of flow direction and chosen input signal where done and results of these tests can be seen in Figures 6.3 and 6.4. Note that the pressure is chosen as input signal for the first example, and air mass flow in the second. Independently of which is chosen the AirFilter works for both 'positive' and 'negative' flow direction.

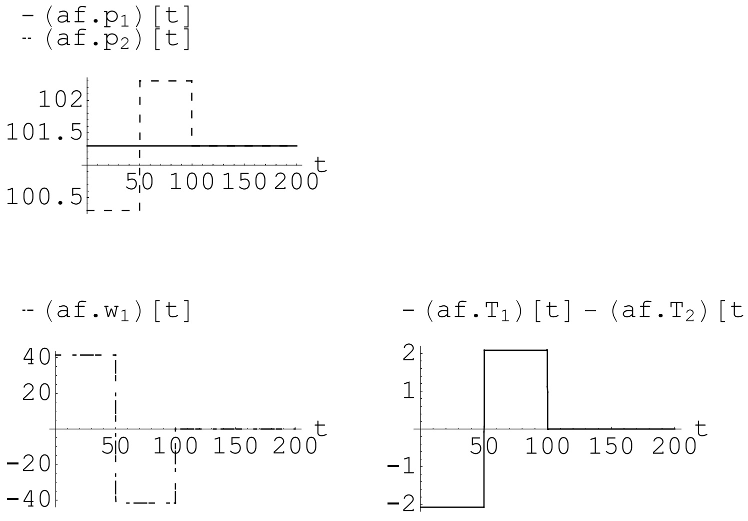


Figure 6.3. *Simulation of air filer. The pressure at inlet is set to constant at 101.3 kPa while outlet pressure is varied over time, upper left figure. Lower left hand figure shows the resulting air mass flow through the filter and the figure at lower right hand side shows the temperature change over time.*

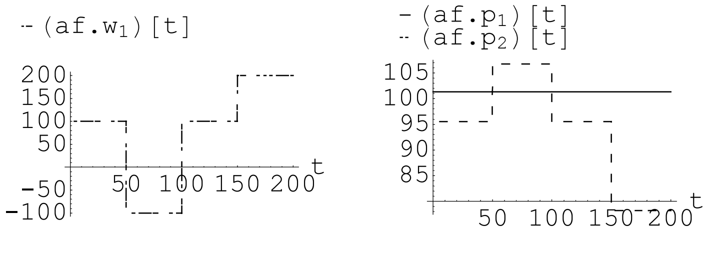


Figure 6.4. *Simulation of air filer. The inlet pressure is held constant at 101.3 kPa while the air mass flow is varied over time. Note that from 50 to 100 s the flow is set negative. The right hand picture shows the resulting pressure.*

6.2 Compressor

The compressor increases the fluid density to allow a greater amount of air to enter the cylinders in each stroke.

Compression and Expansion Process

In this section the most important relationships for analysis of compression and expansion processes are presented. One dimensional, steady flow and an adiabatic process is assumed. The assumptions are listed below

- The first law for one dimensional flow

$$w_{\text{in}} = h_{02} - h_{01} = h_2 - h_1 + \frac{V_2^2 - V_1^2}{2}$$

where w_{in} is the specific work supplied to the system. That is $w_{\text{in}} = -w$ in equation (1.14)

- The relationship for a reversible adiabatic process of an ideal gas with constant specific heats

$$p\, v^\gamma = \text{constant}$$

For a reversible adiabatic process between state 1 and 2 the following equality is given

$$p_1\, v_1{}^\gamma = p_2\, v_2{}^\gamma$$

$\gamma = c_p / c_v$ is the ratio of specific heats. Using the ideal gas equation, (1.1), the following equality may be stated

$$\frac{p_1\, v_1}{T_1} = \frac{p_2\, v_2}{T_2}$$

These two equations can be combined to relate the pressure ratio and the temperature ratio for the two states (polytropic temperature increase caused by adiabatic compression)

$$\frac{T_2}{T_1} = \left(\frac{p_2}{p_1}\right)^{\frac{\gamma-1}{\gamma}}$$

The above equation together with the first law give the following expression for the specific energy required for compression between a specified initial state and a state with pressure p_2

$$w_{\text{in}} = h_{02} - h_{01} = c_p\,(T_{02} - T_{01}) = c_p\, T_{01}\left(\left(\frac{p_{02}}{p_{01}}\right)^{\frac{\gamma-1}{\gamma}} - 1\right)$$

where subscripts 01 and 02 refer to the total or stagnation pressures and temperatures. This may now be used to calculate the consumed energy from the compressor

$$\dot{W}_{\text{cons,ideal}} =$$

$$\dot{m}\,(h_{02} - h_{01}) = \dot{m}\,c_p\,T_{01}\left(\left(\frac{p_{02}}{p_{01}}\right)^{\frac{\gamma-1}{\gamma}} - 1\right) = \dot{m}\,c_p\,T_{01}\left(\frac{T_{02}}{T_{01}} - 1\right) = \dot{m}\,c_p\,(T_{02} - T_{01})$$

As mentioned earlier on, with reference to the definition of a reversible process, these are useful to simplify the analysis of a system. However, the results from this analysis must then be adjusted when applied to real systems. This is often done by introducing a measure of efficiency. For a reversible adiabatic process the efficiency η_c is defined as

$$\eta_c = \frac{\text{work of reversible adiabatic compression from state 1 to 2}}{\text{work of actual adiabatic compression from state 1 to 2}}$$

The isentropic work, w_{in}, and the actual specific work, $w_{\text{in,act}}$, are related as

$$w_{\text{in}} = \eta_c\,w_{\text{in,act}}$$

Using this a final expression for the consumed energy may be expressed

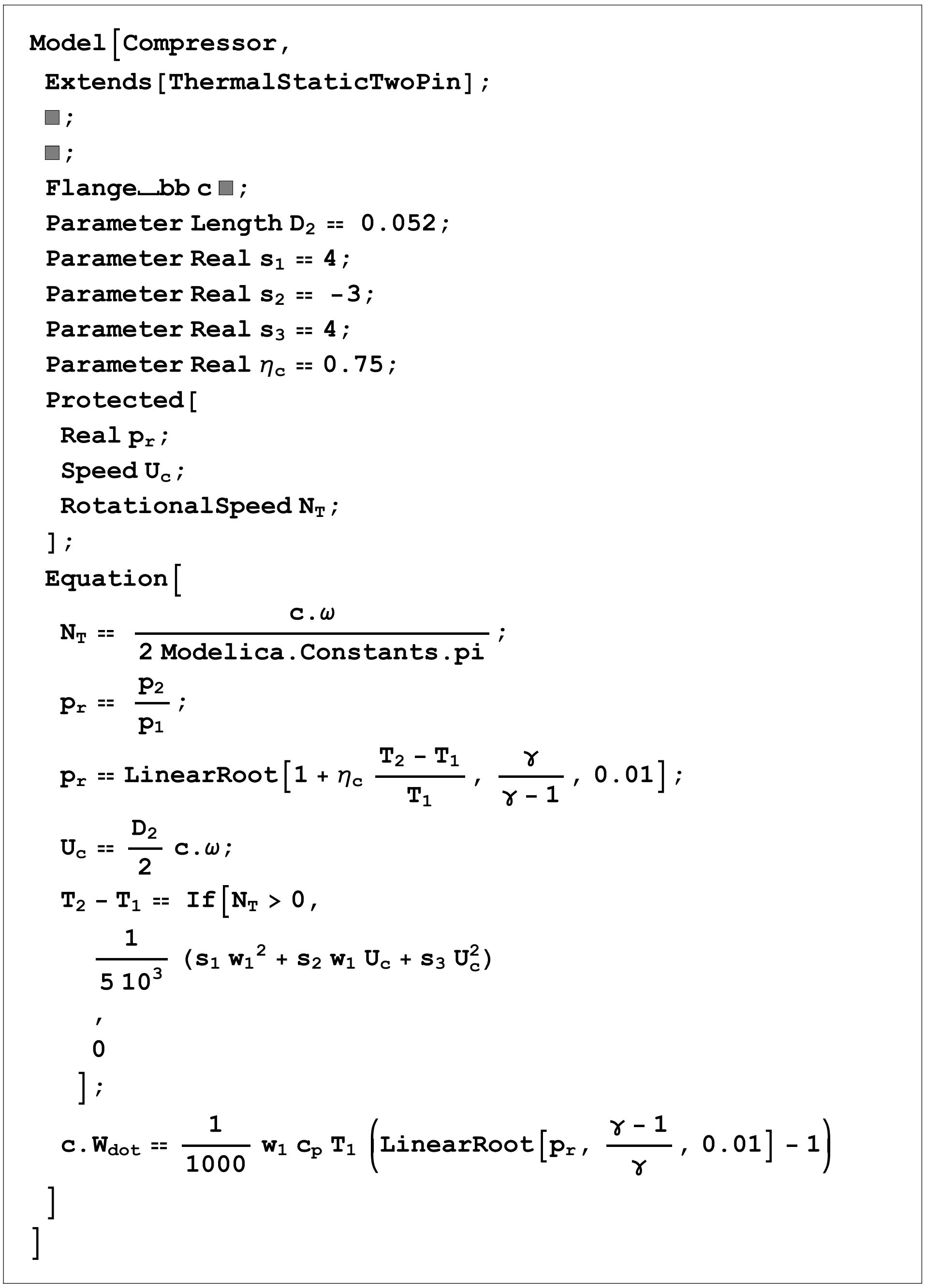
$$\dot{W}_{\text{cons}} = \dot{m}\,w_{\text{in,act}} = \frac{1}{\eta_c}\,\dot{m}\,(h_{02} - h_{01}) = \frac{1}{\eta_c}\,\dot{m}\,c_p\,(T_{02} - T_{01})$$

The temperature loss is modeled according to [Bergström & Brugård], which is a simplification of the procedure in [Müller, 1998]

$$T_2 - T_1 = {U_c}^2\left(s_1\left(\frac{w_1}{U_c}\right)^2 + s_2\,\frac{w_1}{U_c} + s_3\right) \tag{6.1}$$

Where U_c is the impeller tip speed and s_i constants.

MathModelica Model



```
Model[Compressor,
 Extends[ThermalStaticTwoPin];
 ■;
 ■;
 Flange_bb c ■;
 Parameter Length D2 == 0.052;
 Parameter Real s1 == 4;
 Parameter Real s2 == -3;
 Parameter Real s3 == 4;
 Parameter Real ηc == 0.75;
 Protected[
  Real pr;
  Speed Uc;
  RotationalSpeed NT;
 ];
 Equation[
                   c.ω
  NT == ───────────────────────── ;
        2 Modelica.Constants.pi
        p2
  pr == ── ;
        p1
                          T2 - T1     γ
  pr == LinearRoot[1 + ηc ───────, ─────, 0.01];
                            T1      γ - 1
        D2
  Uc == ── c.ω;
        2
  T2 - T1 == If[NT > 0,
      1
    ────── (s1 w1^2 + s2 w1 Uc + s3 Uc^2)
    5 10^3
    ,
    0
   ];
              1                               γ - 1
  c.Wdot == ──── w1 cp T1 (LinearRoot[pr, ─────, 0.01] - 1)
            1000                                γ
 ]
]
```

6.2.1 Validation

Pressure Ratio

According to the plots in Figures 6.5 and 6.6 the model for the pressure ratio is very accurate at stationary conditions (Mean relative error < 1/2%, maximum relative error ≤2.6). It also captures the dynamic behavior very good, especially at higher engine speeds.

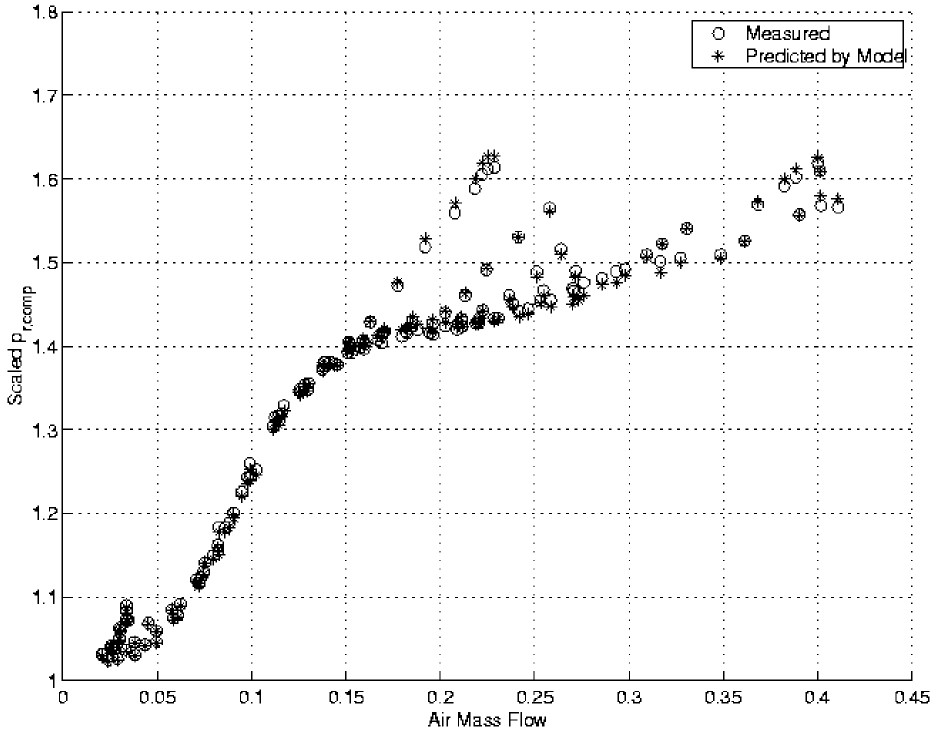


Figure 6.5. Validation of pressure model for the compressor at stationary conditions.

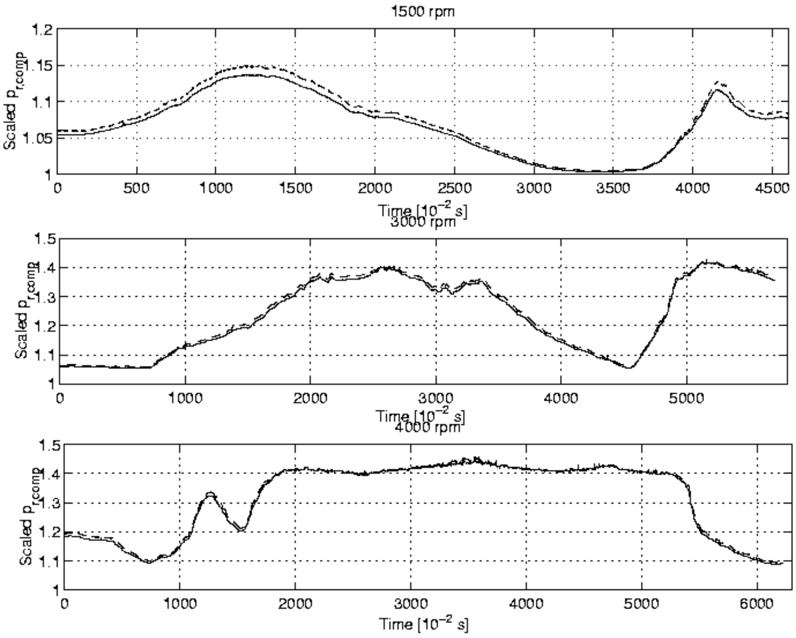


Figure 6.6. *Validation of pressure model for the compressor at dynamic conditions. The engine speed is held constant, while the throttle angle is varied. Measured pressure ratio is represented by the solid line, while, modeled pressure ratio is represented by the dashed line.*

Independence of Flow Direction

The compressor model is not developed to work correctly for negative air mass flows, as there is no data available to test the functionality for this case. However it should still work for negative flows even though the output will not be correct according to physics. That this is the case is shown in Figure 6.7 below.

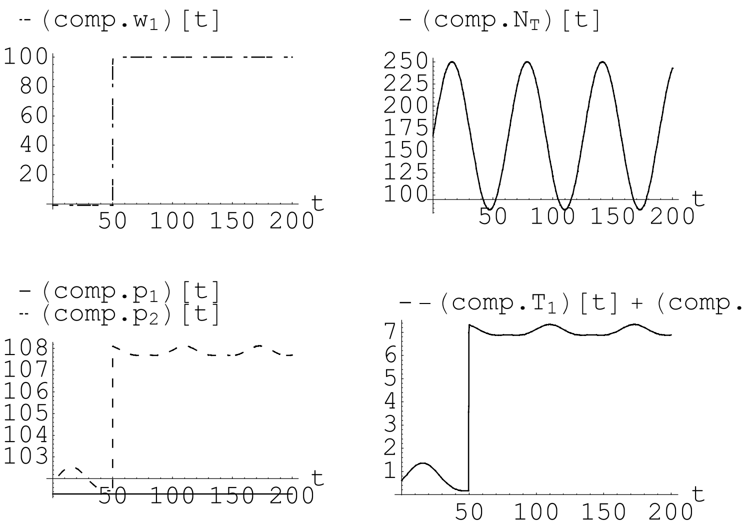


Figure 6.7. Air mas flow and turbine shaft speed are set as input to the compressor. Pressure and temperature increases according to the change in the input signals. Note that for t < 50 the air mass flow is set negative.

6.3 Intercooler

The intercooler is a heat exchanger, which decreases the fluid temperature before it enters the cylinders. This is done for two reasons; to increase the fluid density and consequently the amount of air that may enter in each cylinder stroke, but more important, to decrease knock tendency.

Pressure

Just as for the air filter and exhaust system, equation (6.8) is used to model the pressure loss. However this equation requires turbulent flow. The example below shows that this is a reasonable assumption for the intercooler.

Example, Turbulent Flow

In this example a qualitative calculation of the flow conditions for the tubings in the intake system and the intercooler is made. The assumptions are the following:

■ *$Re_{crit} = 2300$*
■ *The tubings in the intake system have circular cross-section width d=50 mm.*
■ *The intercooler consists of 33 pipes with a 40×1 mm rectangular cross-section area and thus for d_h:*

$$d_h = \frac{4\,A}{S} = \frac{4\times 40}{82}\ mm.$$

Using equation (3.3) and (3.1) together with the data above, the critical mass flow is plotted versus temperature. Here the mass flow through the intercooler refer to the sum of the mass flow of the 33 pipes. The result, shown in Figure 3.8 illustrates that the flow in both cases turns turbulent for quite moderate values of the mass flow.

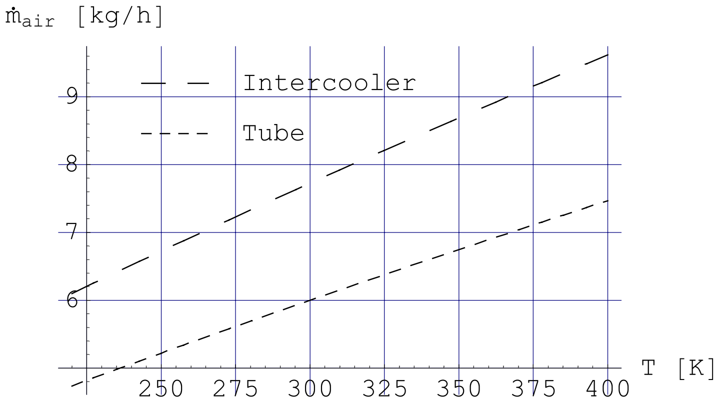


Figure 6.8. *The curves show the limit between laminar and turbulent flow in a circular tube with inner diameter 50 mm and in the intercooler respectively. The flow through the intercooler is normally well above 10 kg / h.*

Temperature

For all practical purposes of intercoolers the flow rate of the cooling fluid, $\dot{m}_{\text{cool}}$, is greater then the air mass flow, $\dot{m}_{\text{air}}$, through it. This means that equation (2.9) could be used

$$T_2 = T_1 - \epsilon\,(T_1 - T_{\text{cool}})$$

where T_1 and T_2 are inlet and outlet temperature respectively, and T_{cool} is the temperature of the cooling air. From this expression it is seen that in order to predict T_2, one need a model for the intercooler efficiency ϵ. The intercooler is model according to the NTU method described in [Bergström & Brugård, 1999]

$$\epsilon = 1 - e^{\frac{e^{[-N^{0.78}\,C]}-1}{N^{-0.22}\,C}}$$

where C is the given by

$$C = \frac{c_{p,\mathrm{air}}\,\dot{m}_{\mathrm{air}}}{c_{p,\mathrm{cool}}\,\dot{m}_{\mathrm{cool}}} = [c_{p,\mathrm{air}} = c_{p,\mathrm{cool}}] = \frac{\dot{m}_{\mathrm{air}}}{\dot{m}_{\mathrm{cool}}}$$

and N by

$$N = \frac{U\,A}{c_p\,\dot{m}_{\mathrm{air}}} = \frac{K}{c_p\,\dot{m}_{\mathrm{air}}^{0.2}\,\mu_i^{0.5}}$$

and A is constant and denotes some effective surface for the heat transfer. U is the overall heat transfer coefficient, describing the combined heat transfer by convection on the inner tube surface, conduction inside the tube material, and again convection on the outer tube surface, as described in Chapter 2. Finally the dynamic viscosity is given by

$$\mu_i = 2.3937\,\frac{\left(\frac{1}{2}\,(T_1+T_2)\right)^{0.7617}}{10^7}$$

The unknown constant K is determined from a least squares fit to measured data and as can be seen, the model is expressed in variables that can easily be measured or estimated ($\dot{m}_{\mathrm{air}}$, $\dot{m}_{\mathrm{cool}}$, T_1 and T_{cool}). The grouping $U\,A\,/\,c_p$ is called the *number of heat transfer units* (NTU) and as the above model is based on this grouping it is referred to as the *NTU-model*.

Example, Intercooler Efficiency

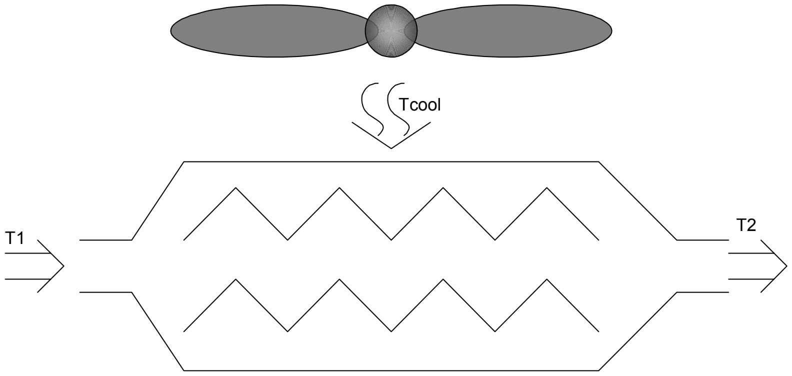


Figure 6.9. Sketch of intercooler.

Suppose we have air flowing through an intercooler with mass flow rate, $\dot{m}_{air}$ and that the intercooler is cooled by air with mass flow rate $\dot{m}_{cool}$ as illustrated in Figure 6.9. If $\dot{m}_{cool} > \dot{m}_{air}$ the efficiency is given from equation (2.9)

$$\epsilon = \frac{T_1 - T_2}{T_1 - T_{\text{cool}}}$$

A plot of the ideal intercooler efficiency is presented in Figure 6.10. In Figure 6.11 the efficiency is plotted for the NTU-model with $\dot{m}_{cool} = 100\,g/s$ *and* $T_{cool} = 296\,K$.

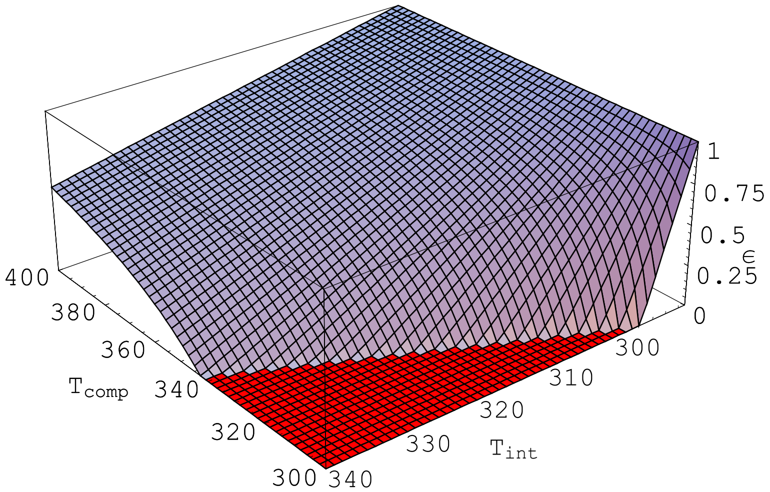


Figure 6.10. Calculated intercooler efficiency when the cooling temperature is held constant at 20° C. Note that the efficiency is undefined (or negative) when $T_{int} < T_{cool}$.

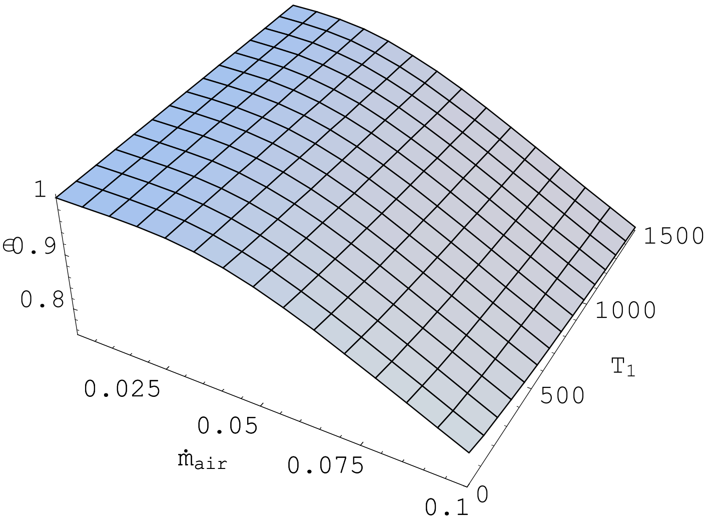


Figure 6.11. Intercooler efficiency calculated from NTU-model.

□

MathModelica Model

The intercooler is symmetric, so it should work in the same way independent of flow direction. For small values flows *N* becomes infinite, however we know that the efficiency, ϵ, is close to one for small flows, therefore to avoid numerical problems the ϵ and *N* are set constant for small flows.

```
Model[Intercooler,
 Extends[Restriction[{ξ == 8}]];
 ■;
 ■;
 Parameter Temperature T_cool == 293;
 Parameter MassFlow w_cool == 200;
 Parameter Real K == 22;
 Protected[
  Real { ϵ, C, N, μ_i};
  Temperature {T_in, T_out};
  Pressure {p_in, p_out};
  MassFlow w;
 ];
 Equation[
  If[w_1 < 0,
   T_in == T_2;
   T_out == T_1;
   p_in == p_2;
   p_out == p_1;
   w == w_2;
   ,
   T_out == T_2;
   T_in == T_1;
   p_out == p_2;
   p_in == p_1;
   w == w_1;
  ];
  T_out == T_in - ϵ (T_in - T_cool);
  μ_i ==
      1
     ───
     10^7
     (2.3937 LinearRoot[1/2 (Tin + Tcool), 0.7617, 0.01]);
   C == w/wcool;
   If[w > 0.1,
    ϵ == 1 - exp[(exp[-LinearRoot[N, 0.78, 0.01] C] - 1)/(C LinearRoot[N, 0.22, 0.01])];
    N ==
     K/(cp LinearRoot[1/1000 w, 0.2, 10^-4] 10^-3
        LinearRoot[10^6 μi, 0.5, 0.01]);
    ,
    N == 20;
    ϵ == 1
   ];
  ]
 ]
```

6.3.1 Validation

Temperature

The unknown constant K in the NTU-model is determined from a least square fit to measured data. Note however that the intercooler is supplied with cooling air by a fan at the test bench. Different fan speeds can be set, but there is no way to measure the actual cooling air flow. This means that we have to make an estimation of the air mass flow and therefore the model is not as good as we could hope.

Figures 6.12 and 6.13 show validations for the efficiency and temperature respectively. It can be noted that for low air mass flows the efficiency-model deviates a lot from the measured. A reason for this could be that the assumptions for estimating $\dot{m}_{\text{cool}}$ might be erroneous. Fortunately since the error is largest for low values of $\dot{m}_{\text{air}}$, where the inlet temperature, T_1, is also rather low, the resulting error for the outflow temperature, T_2, is rather low. Over all their should be possible to improve this model a lot.

Note that the fit should be done so the errors for the output temperature are minimized, not so that the errors for the efficiency are minimized, this should improve the temperature model in Figure 6.13.

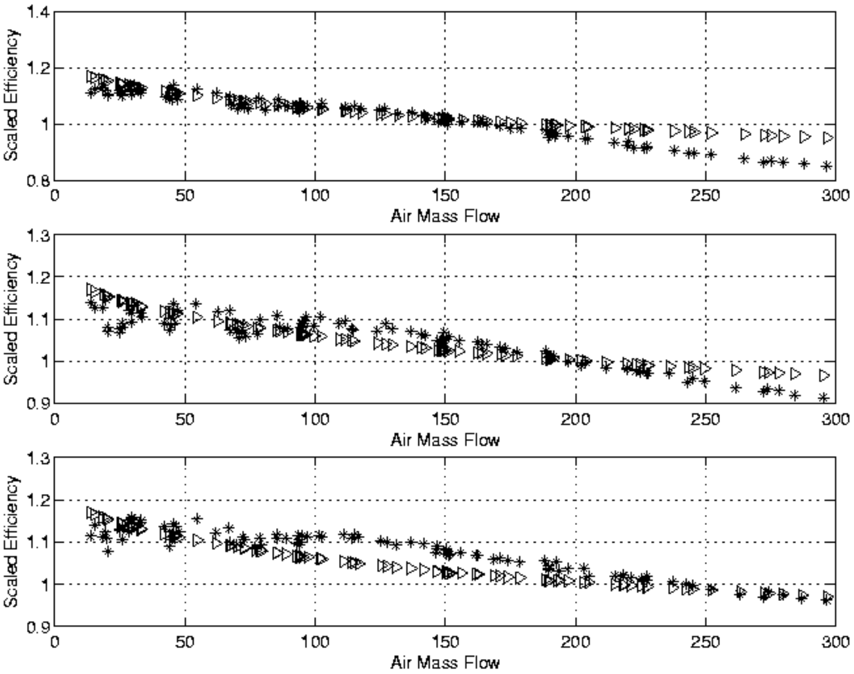


Figure 6.12. *Validation of the NTU-model for three different fan speeds. The upper figure corresponds to the lowest fan speed.*

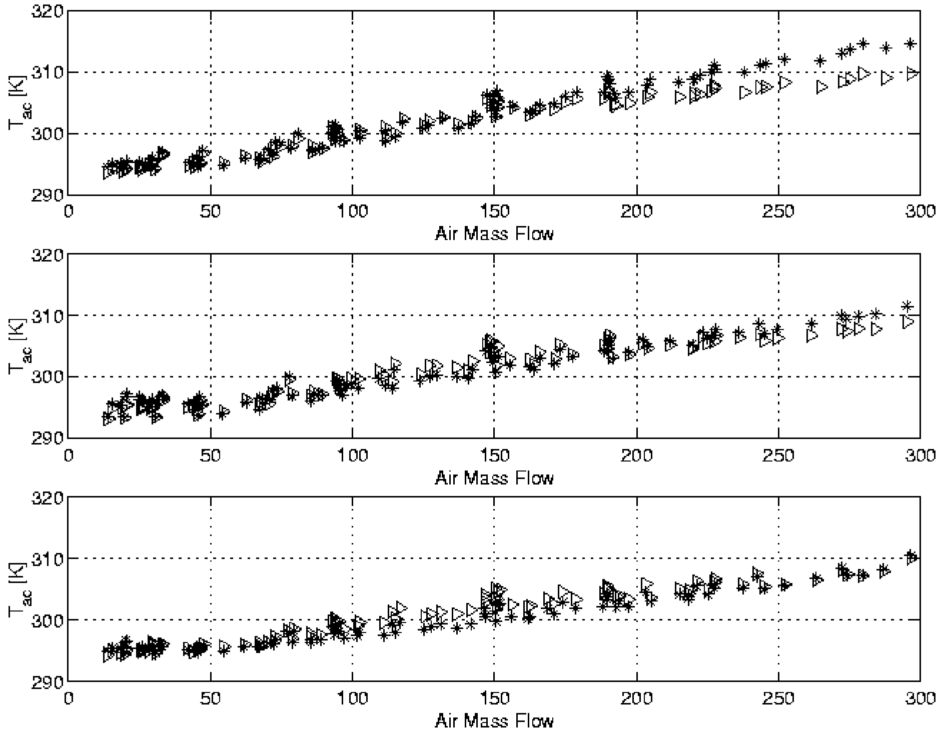


Figure 6.13. *Validation for T_{int} based on the NTU-model for three different fan speeds. The upper figure corresponds to the lowest fan speed.*

Pressure

A least square fit to measured data, gives the constant ξ for equation (6.8). In Figure 6.14 the model is represented by a solid line, with validation data superimposed. As can be seen clearly in the figure this model describes the pressure loss very accurately and the average error is around 100 Pa. For the intercooler both the inlet pressure and temperature varies more than for the air filter, consequently this model is validated for a broader range of data. However there is a systematic error. Studying the relative error in the lower plot of Figure 6.14, suggests that a linear term of $\dot{m}_{\text{air}}$ might improve the model accuracy.

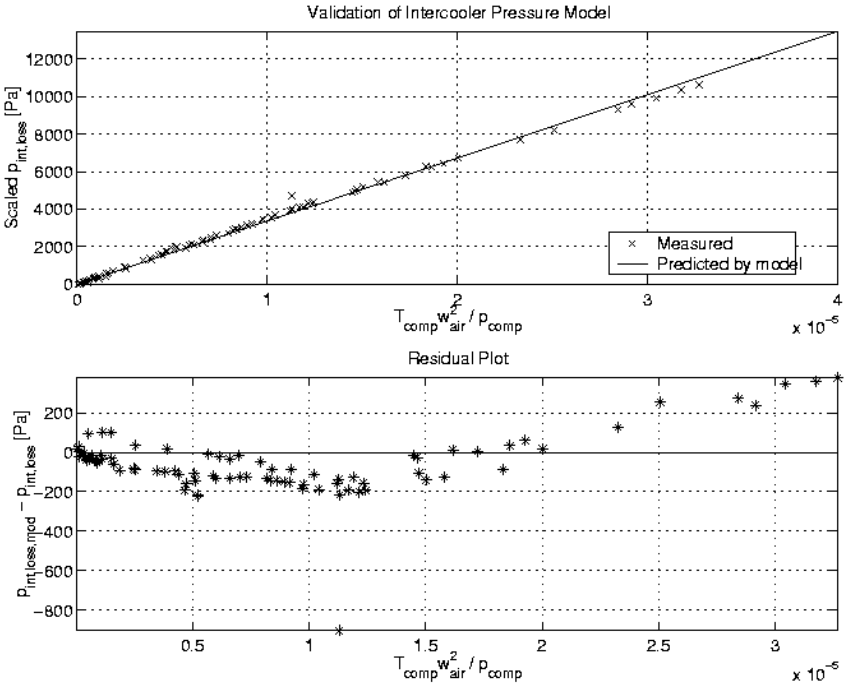


Figure 6.14. *Validation of pressure loss over intercooler,the corrected air mass flow is given by $T_{comp}\,\dot{m}_{air}^{2}$ / p_{comp}.Except for one outlier all errors are within an acceptable range. Note that the pressure loss has been scaled.*

Independence of Flow Direction

Simulations that shows that the model works independently of flow direction and chosen input signal where also done and results of these tests can be seen in Figure 6.15.

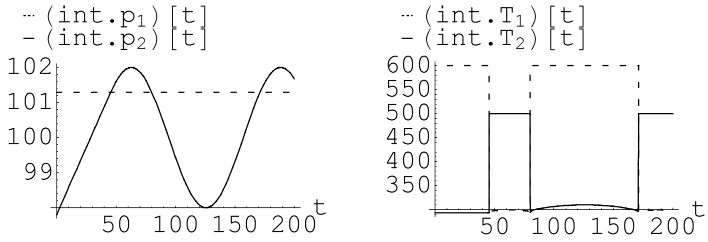


Figure 6.15. *Simulation of exhaust system. The pressure at inlet is held constant at 101.3 kPa while inlet pressure is varied over time. The right hand figure shows the resulting temperature.*

6.4 Butter Fly Valve

The throttle is a butter fly valve used to control the air mass flow into the engine.

A standard approach when modeling the throttle plate air mass flow, $\dot{m}_{\text{air}}$, is to consider the air mass flow through a ventury, which is given by

$$\dot{m}_{\text{air}} = \frac{p_{\text{in}}}{\sqrt{R\,T_{\text{in}}}}\,Q_{\text{th}}\,\Psi \tag{6.2}$$

where Q, called effective area, depends on the opening area, A_{th}, and a discharge coefficient, C_d, that in their turn depend on the shape of the flow area (refer to [Krysander, 2000] for further details)

$$Q_{\text{th}} = A_{\text{th}}\,C_d$$

and Ψ is a nonlinear function of the pressure ratio, $p_r = \frac{p_{\text{out}}}{p_{\text{in}}}$, given by

$$\Psi = \begin{cases} \sqrt{\frac{2\gamma}{\gamma-1}\left(p_r^{\frac{2}{\gamma}} - p_r^{\frac{\gamma+1}{\gamma}}\right)} & \text{for } p_r > \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \\ \sqrt{\frac{2\gamma}{\gamma-1}\left(\left(\frac{2}{\gamma+1}\right)^{\frac{2}{\gamma-1}} - \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}\right)} & \text{else} \end{cases} \tag{6.3}$$

where γ is the ratio of the two specific heats. The limit for Ψ is set by the velocity of sound. Figure 6.16 shows how Ψ depends on inlet and outlet pressure.

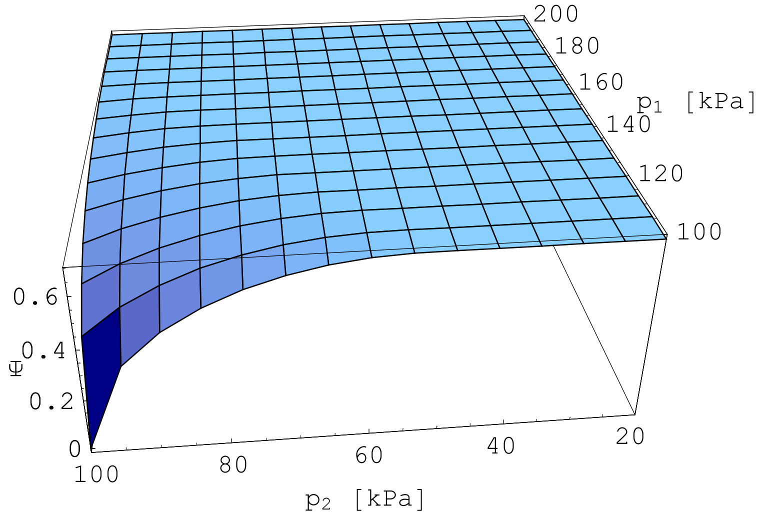


Figure 6.16. Ψ as a function of inlet and outlet pressure.

[Nyberg, 1999] suggests the following simple model for the opening area:

$$A_{\text{th}} = A_1 (1 - \text{Cos}[a_0\,\alpha + a_1]) + A_0$$

where $A_1$is the area that is covered by the throttle plate when the throttle is closed and A_0 is the *leak area* present even though the throttle is closed. The parameters a_o and a_1 are compensations for that the actual measured throttle angle may be scaled or biased because of production tolerances. The discharge coefficient, C_d, is assumed to be constant, thus

$$Q_{\text{th}} = Q_1 (1 - \text{Cos}[a_0\,\alpha + a_1]) + Q_0 \tag{6.4}$$

The resulting air mass flow is shown in Figure 6.17.

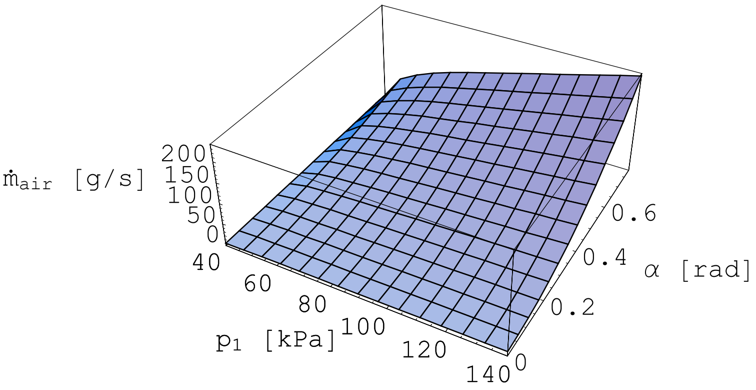
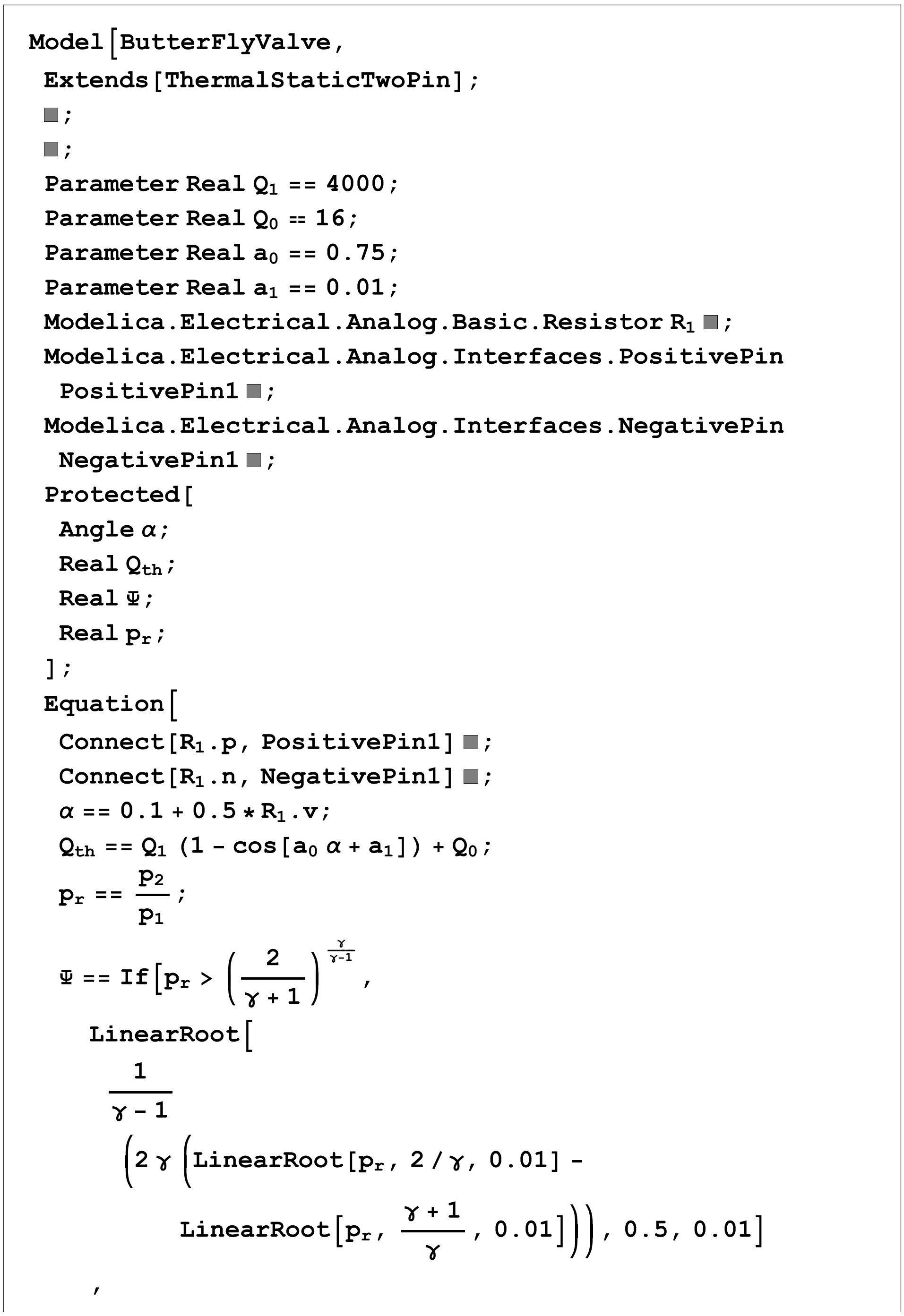


Figure 6.17. Resulting air mass flow as a function of inlet pressure and throttle angle.

To have a complete model, an equation for the temperature change is needed to. A simple model (with no physical background) that describes the temperature change qualitatively is therefore introduced

$$T_2 = T_1 + \frac{100}{1 + w_1} \qquad (6.5)$$

MathModelica Model



```
Model[ButterFlyValve,
 Extends[ThermalStaticTwoPin];
 ■;
 ■;
 Parameter Real Q1 == 4000;
 Parameter Real Q0 == 16;
 Parameter Real a0 == 0.75;
 Parameter Real a1 == 0.01;
 Modelica.Electrical.Analog.Basic.Resistor R1 ■;
 Modelica.Electrical.Analog.Interfaces.PositivePin
  PositivePin1 ■;
 Modelica.Electrical.Analog.Interfaces.NegativePin
  NegativePin1 ■;
 Protected[
  Angle α;
  Real Qth;
  Real Ψ;
  Real pr;
 ];
 Equation[
  Connect[R1.p, PositivePin1] ■;
  Connect[R1.n, NegativePin1] ■;
  α == 0.1 + 0.5 * R1.v;
  Qth == Q1 (1 - cos[a0 α + a1]) + Q0;
  pr == p2/p1;
  Ψ == If[pr > (2/(γ + 1))^(γ/(γ-1)),
    LinearRoot[
     1/(γ - 1)
      (2 γ (LinearRoot[pr, 2 / γ, 0.01] -
          LinearRoot[pr, (γ + 1)/γ, 0.01])), 0.5, 0.01]
    ,
```

$$\sqrt{\frac{2\gamma\left(\left(\frac{2}{\gamma+1}\right)^{\frac{2}{\gamma-1}}-\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}\right)}{\gamma-1}}$$

];

$$w_1 == \frac{p_1\, Q_{th}\, \Psi}{\texttt{LinearRoot}[R\, T_1,\ 0.5,\ 0.01]};$$

$$T_2 == T_1 + \frac{100}{1 + w_1};$$

]

]

6.4.1 Validation

Air Mass Flow

As described above the air mass flow depends on the Q-function. Figure 6.18 shows that the modeled Q_{th} is close to the measured. This means that the air mass flow model is also good.

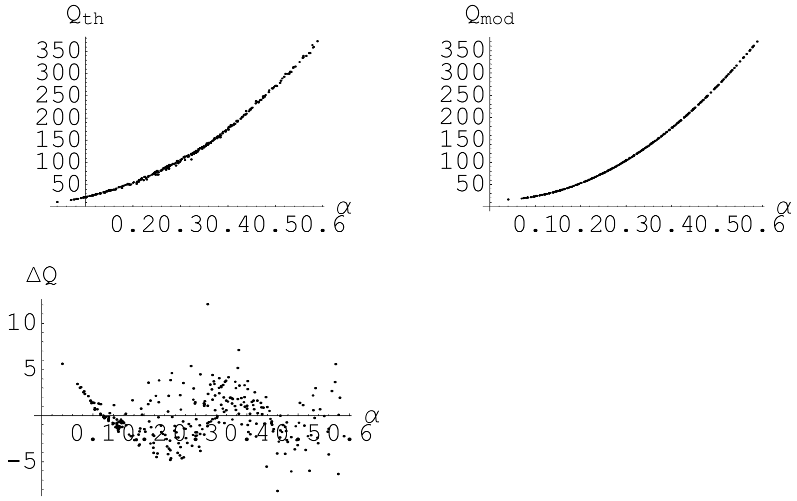


Figure 6.18. Upper left Q-function calculated from measured data. Upper right Q-function according to model.

Temperature

Even though the temperature model has no physical background it still gives a quite good description of the temperature change as can be seen in Figure 6.19.

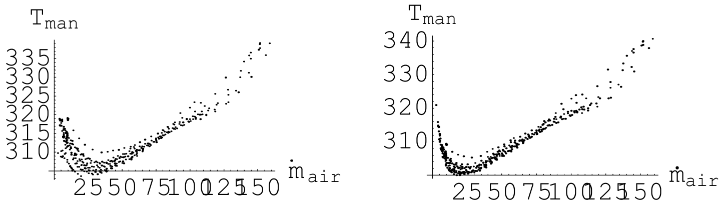


Figure 6.19. Measured and modeled temperature at the throttle

6.5 Base Engine

The actual work of the engine is performed in a four stroke cycle in the cylinders. During the *intake stroke* the cylinder is filled with a fresh air/fuel charge. A *compression stroke* follows, where the air/fuel mixture is compressed to higher temperature and pressure. At the end of this stroke a spark ignites the mixture and initiates the combustion. During the *expansion stroke,* that follows, work is produced by the expanding fluid. At the end of this stroke the exhaust port is opened and a *blowdown* process begins. In the final stroke, the *exhaust stroke*, the rest of the gas in the combustion chamber is pushed out into the exhaust system. Then a new cycle starts.

The thermodynamic properties for different processes in the **ideal** gas standard cycle is given Table 4. Due to e.g. heat loss the actual combustion process will not follow the ideal cycle exactly.

Air Flow into the Cylinders

The air mass flow passing the intake valves, $\dot{m}_{\text{air}}$, out of the intake manifold into the cylinders depends on many parameters, but the engine speed, intake manifold pressure and air temperature are the most important. The parameter volumetric efficiency, η_{vol}, is used to describe the effectiveness of the engine's ability to induct new air into the cylinders. The volumetric efficiency depends on engine speed and intake manifold pressure, however in this model η_{vol} is set as a constant for simplicity.

Process	Model Properties	Equations
Compression (1 - 2)	Isentropic (adiabatic and reversible)	$\delta q = ds = 0$
Combustion (2 - 3)	Adiabatic heat addition a. Constant Volume b. Constant Pressure c. Partly constant volume and partly constant pressure	 $dq = c_v\,dT$ $dq = c_p\,dT$
Expansion (3 - 4)	Isentropic (adiabatic and reversible)	$\delta q = ds = 0$
Heat exchange (4 - 1)	Constant volume	$dV = 0$

Ideal gas standard cycle

Table 4

The air density, ρ, in the cylinder can be calculated using the ideal gas law, equation (1.1) , assuming intake conditions in the cylinder

$$\rho = \frac{p_1}{R\,T_1}$$

If the volumetric efficiency is given, the air mass flow into the cylinders can be calculated as

$$\dot{m}_{\text{air}} = \eta_{\text{vol}} \frac{V_d\,N\,p_1}{n_r\,R\,T_1} \tag{6.6}$$

where V_d is the total cylinder volume, N the engine speed, and n_r number of engine revolutions per cycle .

Temperature

The temperature change is easy to calculate from an ideal combustion cycle, however the combustion cycle is fare from ideal, due to e.g. temperature losses by conduction. The actual temperature losses are very hard to calculate, and are out of the scope for this work. Therefore a very simple temperature model that states that the temperature increases with a constant of 600 Kelvin is used.

Torque

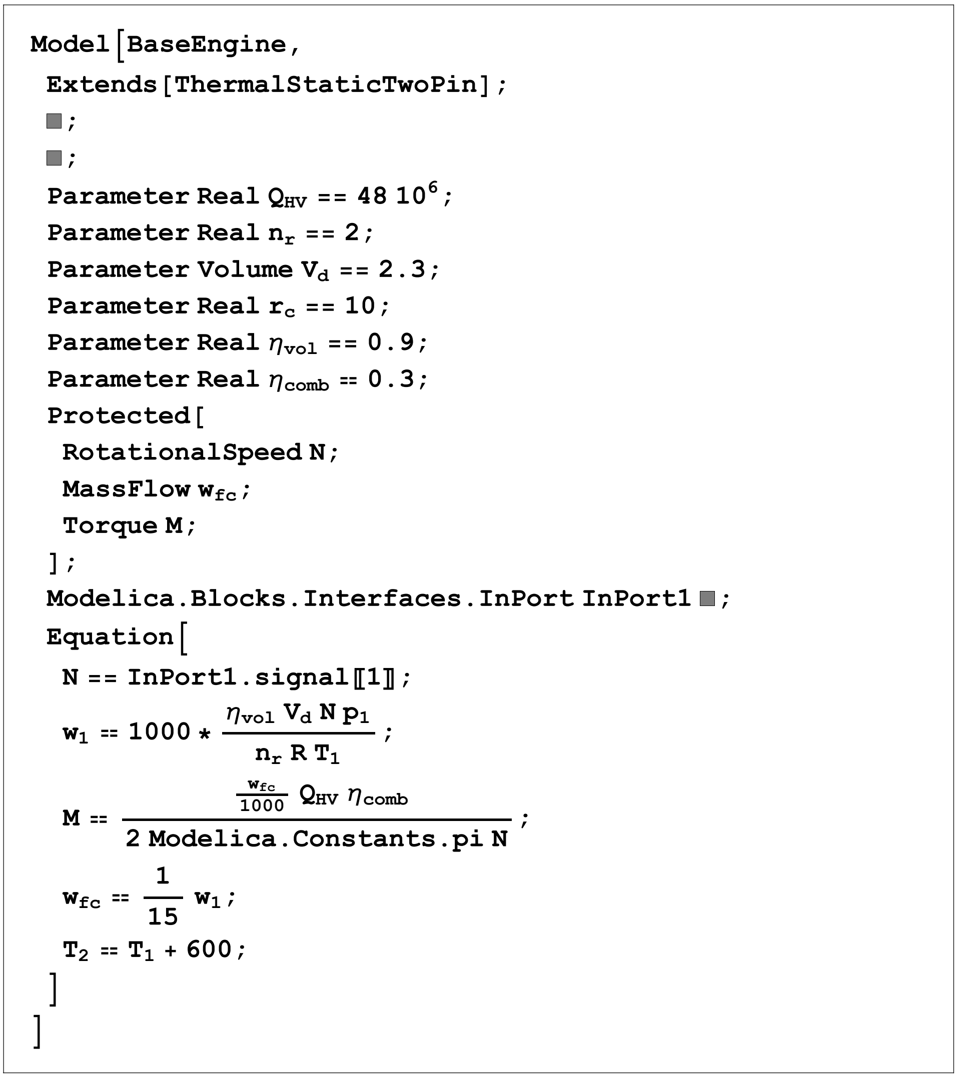
The engine torque, M, is a function of the mass flow of fuel into the cylinders, $\dot{m}_{\text{fc}}$, the heating value of the fuel, Q_{HV}, combustion efficiency, η_{comb}, and engine speed, N

$$M = \frac{\dot{m}_{\text{fc}}\,Q_{\text{HV}}\,\eta_{\text{comb}}}{2\,\pi\,N} \tag{6.7}$$

The stoichiometric air to fuel ratio $(A\,/\,F)_s$ is used to calculate the amount of fuel that should be injected into the engine for optimal combustion

$$\dot{m}_{\text{fc}} = \frac{1}{(A/F)_s}\,\dot{m}_{\text{air}}$$

MathModelica Model



```
Model[BaseEngine,
 Extends[ThermalStaticTwoPin];
 ■;
 ■;
 Parameter Real Q_HV == 48 10^6;
 Parameter Real n_r == 2;
 Parameter Volume V_d == 2.3;
 Parameter Real r_c == 10;
 Parameter Real η_vol == 0.9;
 Parameter Real η_comb == 0.3;
 Protected[
  RotationalSpeed N;
  MassFlow w_fc;
  Torque M;
 ];
 Modelica.Blocks.Interfaces.InPort InPort1 ■;
 Equation[
  N == InPort1.signal[[1]];
  w_1 == 1000 * (η_vol V_d N p_1)/(n_r R T_1);
  M == ((w_fc/1000) Q_HV η_comb)/(2 Modelica.Constants.pi N);
  w_fc == (1/15) w_1;
  T_2 == T_1 + 600;
 ]
]
```

6.5.1 Validation

As the volumetric efficiency is set constant there is no real meaning in validating the result. Furthermore the nature of the BaseEngine is that the air mass flow through may never be negative, see equation (6.6). The air mass flow is proportional to both engine speed and inlet pressure, while the output torque, M, is only a function of the inlet pressure, see Figure 6.20. Note that in reality the volumetric efficiency, η_{vol}, is a function of both inlet pressure and engine speed, $\eta_{\mathrm{vol}}(p_1, N)$. this means that if a proper efficiency map is introduced the engine characteristics would change somewhat.

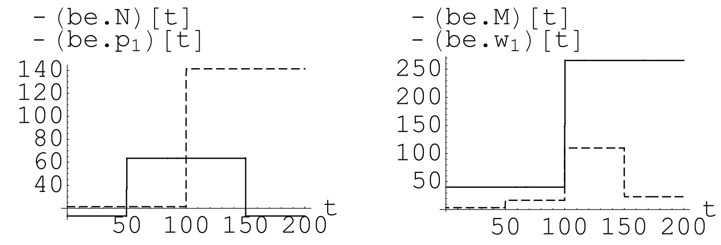


Figure 6.20. *The right hand figure shows that torque, M, does only depend on inlet pressure, p_1, while air mass flow is a function of both p_1 and engine speed, N.*

6.6 Turbine

The kinetic energy of the exhaust gases is used by the turbine to propel the turbine shaft which supplies the compressor with the energy needed for the compression process. The turbine works in a similar way as the compressor but in the opposite direction. However instead of using a temperature model, as we did for the compressor, a pressure model is used for the turbine. This pressure model is calculated from turbine maps supplied by the manufacturer

$$p_r = 1 + r_1\, w_1 + r_2\, w_1{}^2$$

This model could be improved by substituting the air mass flow with a dimensionless or normalized quantity. In this way the model could take the large density variations of the turbine flow into account.

If the turbine is supplied with to much air it will overspead. Therefore one uses a waste gate, that bypasses some of the air, or a variable geometry turbine (VGT), that varies the inlet blade angle to control the turbine efficiency. Neither of these are modeled here, instead a upper limit for the power output, W_{dot}, is set.

MathModelica Model

```
Model[Turbine,
 Extends[ThermalStaticTwoPin];
 ■;
 ■;
 Flange_aa c ■;
 Parameter Real r₁ == 1;
 Parameter Real r₂ == 140;
 Parameter Real ηt == 0.6;
 Protected[
  Real pr;
  RotationalSpeed NT;
 ];
 Equation[
                    c.ω
  NT == ───────────────────────── ;
        2 Modelica.Constants.pi
        p1
  pr == ── ;
        p2
                            w1    2      w1          w1
  pr == 1 + If[w1 < 40, r2 (────)  + r1 ──── , 4 r1 ──── ];
                            1000         1000        1000
                                  p2   γ - 1
  T2 == T1 ηt LinearRoot[ ── , ───── , 0.01];
                                  p1     γ
             Min[30, abs[w1]]
  c.Wdot == ──────────────── cp T1 ηt
                 1000
                    p2   γ - 1
     (LinearRoot[ ── , ───── , 0.01] - 1);
                    p1     γ
 ]
]
```

6.6.1 Validation

It is not possible to perform any appropriate measurements for validation of the turbine at our test bench. This is mainly because lack of control of how much of the air that passes through the turbine and how much is bypassed through the waste gate. Furthermore the temperatures around the turbine are so high that the sensors have to work outside their working range, so precision is lost. Comparing Figure 6.21 and 6.22 shows that the model behaves approximately like the real turbine.

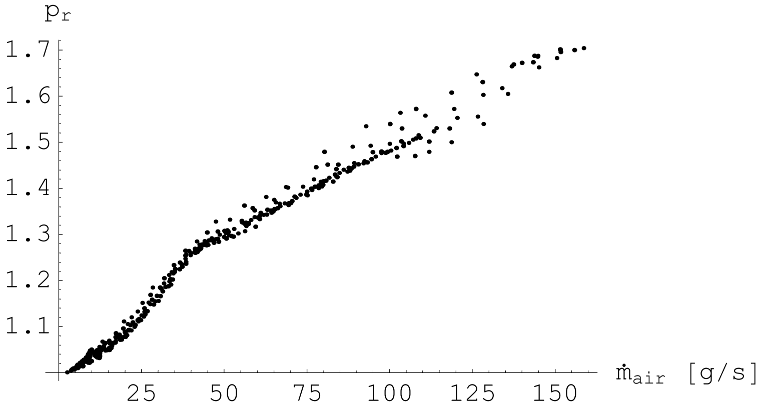


Figure 6.21. Measured turbine pressure ratio, $p_r = p_1 / p_2$, as a function of air mass flow. Around 40 g/s the waste gate is opened.

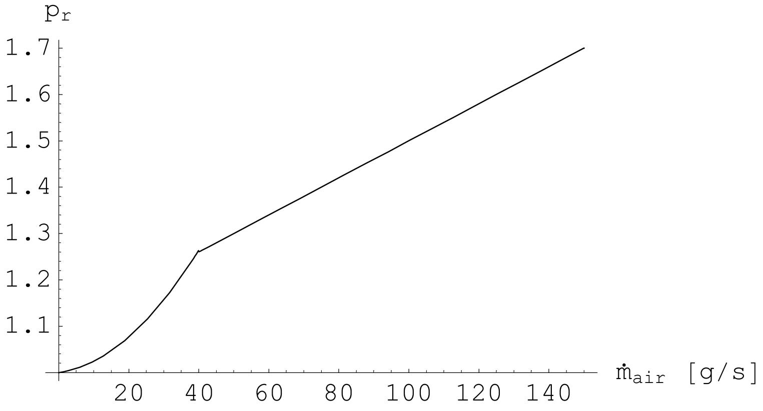


Figure 6.22. Pressure ratio according to the turbine model.

Independence of Flow Direction

Simulations that show that the model works independently of flow direction where done and results of these tests can be seen in Figure 6.23.

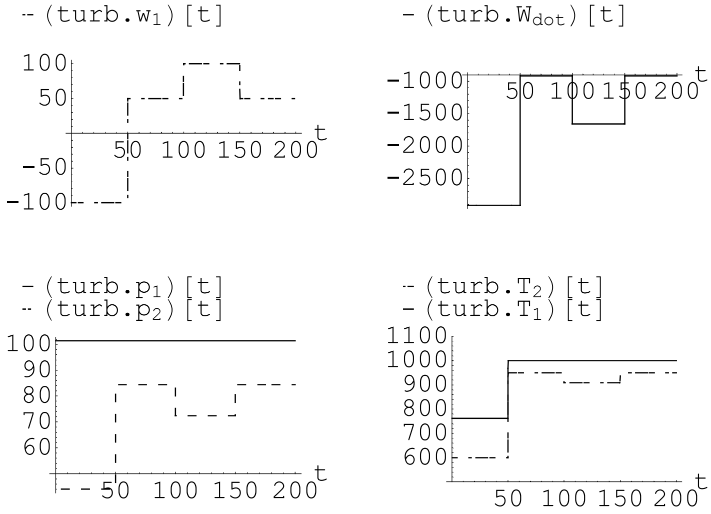
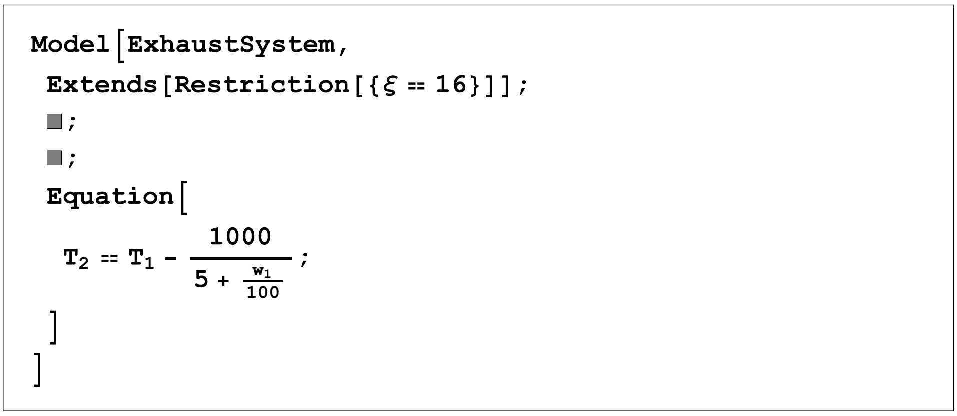


Figure 6.23. *Simulation of turbine. The pressure at inlet is set to constant at 101.3 kPa while air mass flow is varied over time, upper left figure. Lower left hand figure shows the resulting pressure and the figure at lower right hand side shows the temperature change over time.*

6.7 Exhaust System

The catalyst decreases the amount of pollutants in the exhaust gases. As it works at high temperatures (around 800K) temperature decreases due to heat exchange with the surroundings. Just as the air filter the exhaust system works as a restriction for the fluid which causes a pressure drop. A model for the efficiency ϵ should be developed in the future (compare with the intercooler model). The temperature model is chosen so it describes the temperature loss qualitatively.

MathModelica Model



```
Model[ExhaustSystem,
 Extends[Restriction[{ξ == 16}]];
 ■;
 ■;
 Equation[
   T2 == T1 - 1000/(5 + w1/100);
 ]
]
```

6.7.1 Validation

A least square fit to measured data, gives the constant ξ for equation (6.8). In Figure 6.24 the model is represented by a solid line, with validation data superimposed. For small mass flows the model shows excellent agreement with measured data, while for larger gas mass flows the actual loss fluctuates somewhat more from the predicted loss. This could possibly be explained with insecurity in our measurements, as the temperature at the turbine outflow exceeds the temperature range for the pressure transmitters. The average error is 0.5 kPa and the maximum error is less then 4.2 kPa.

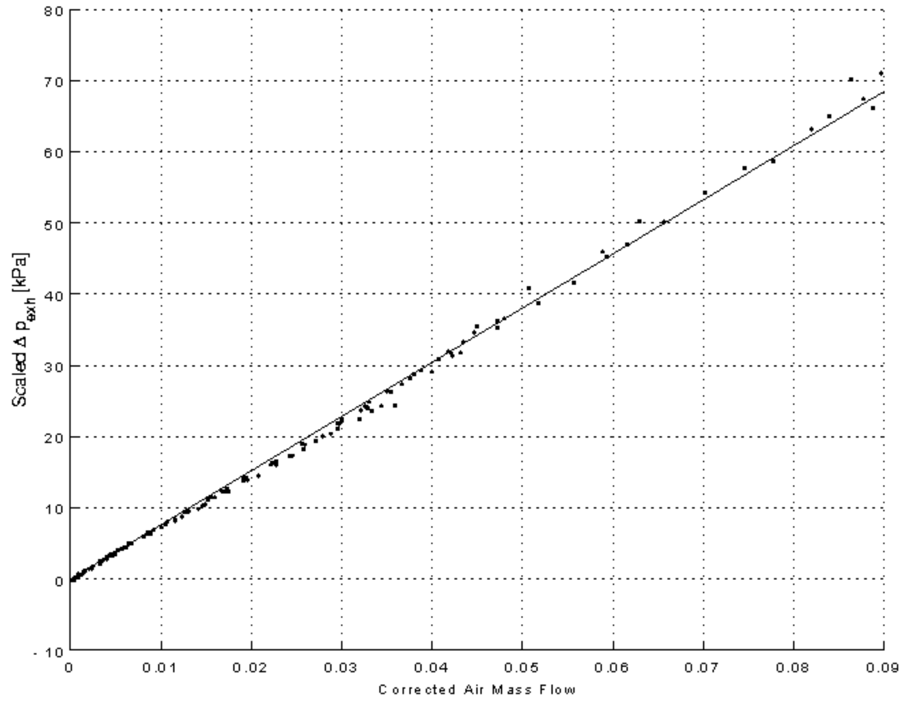


Figure 6.24. *Validation of pressure loss over exhaust system, the corrected air mass flow is given by (T_{amb} $\dot{m}_{air}^2$) / p_{amb}. Note that the pressure loss has been scaled.*

Independence of Flow Direction

Simulations that show that the model works independently of flow direction and chosen input signal where also done and the results are shown in Figures 6.25 and 6.26.

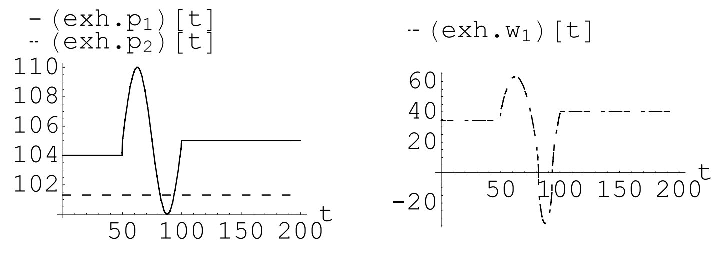


Figure 6.25. *Simulation of exhaust system. The pressure at outlet is held constant at 101.3 kPa while inlet pressure is varied over time. Left figure shows the resulting air mass flow through the exhaust system.*

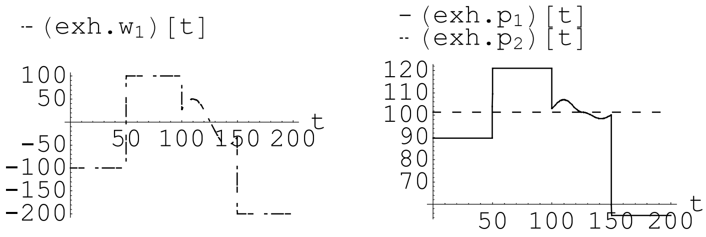


Figure 6.26. Simulation of exhaust system. The outlet pressure is held constant at 101.3 kPa while the air mass flow is varied over time. Note that flow is negative except for 50 to 100. The right hand picture shows the resulting pressure.

6.8 Volume

The volume stores mass and energy and introduces dynamics to the model. This two state model is modeled according to [Eriksson 2000] by using the ideal gas law, `p V = m R T`, and a temperature rate equation derived from energy balance. The volume can easily be extended to handle external heat, Q, and work, W, by adding a heat and energy connector and then adding the heat and energy flow to the temperature rate equation.

Example, Pressure in a Volume

Consider the one-dimensional flow of air through the volume in Figure 6.27.

The mass flow rate into and out of the reservoir is denoted with $\dot{m}_{in}$ and $\dot{m}_{out}$ respectively and m is the stored mass in the system. Equation (1.9) and (1.10) can be combined to the following relation

$$\frac{d}{\mathrm{dt}} m = \dot{m}_{\mathrm{in}} - \dot{m}_{\mathrm{out}} \tag{6.8}$$

Using the ideal gas equation of state (1.1) for the volume, m can be expressed in terms of the volumes pressure, p, temperature, T and volume, V

$$m = \frac{V}{R\,T}\,p \tag{6.9}$$

If equation (6.9) is differentiated, the left hand side in (6.8) can be determined

$$\frac{d}{\mathrm{dt}} m = \frac{V}{R}\,\frac{d}{\mathrm{dt}}\left(\frac{p}{T}\right) = \frac{V}{R\,T}\left[\frac{d}{\mathrm{dt}}\,p - \frac{p}{T}\,\frac{d}{\mathrm{dt}}\,T\right]$$

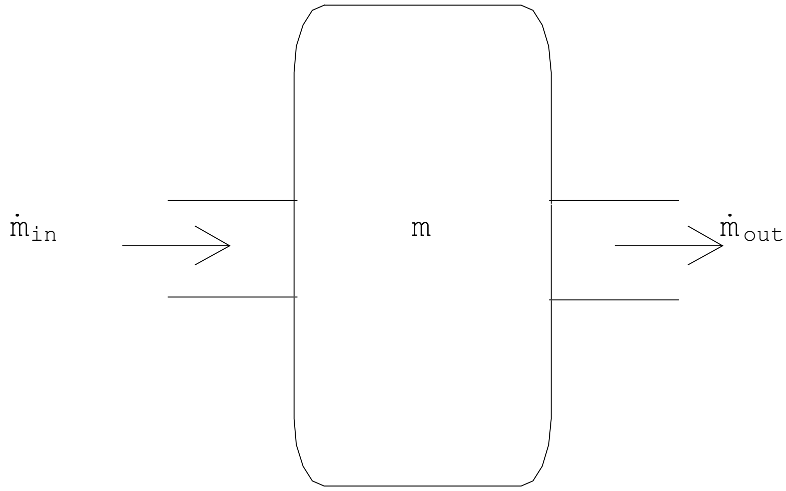


Figure 6.27. Mass flow in to and out of a volume

or

$$\frac{d}{\mathrm{dt}}\,p = \frac{R\,T}{V}\,\frac{d}{\mathrm{dt}}\,m + \frac{p}{T}\,\frac{d}{\mathrm{dt}}\,T = \frac{R\,T}{V}\,\frac{d}{\mathrm{dt}}\,m + \frac{m\,R}{V}\,\frac{d}{\mathrm{dt}}\,T \tag{6.10}$$

If the temperature is constant Equation (6.10) can be written

$$\frac{d}{\mathrm{dt}}\,p = \frac{R\,T}{V}\,\frac{d}{\mathrm{dt}}\,m = \frac{R\,T}{V}\,(\dot{m}_{\mathrm{in}} - \dot{m}_{\mathrm{out}}) \tag{6.11}$$

this equation may be used when a one-state model is sufficient, while Equation (6.10) can be used together with a temperature rate equation. This may be derived from the energy balance which gives,

$$\frac{d}{\mathrm{dt}}\,T = \frac{\dot{m}_{\mathrm{in}}(\,c_v(\,T_{\mathrm{in}}-T\,)+R\,T_{\mathrm{in}}) - \dot{m}_{\mathrm{out}}(\,c_v(\,T_{\mathrm{out}}-T\,)+R\,T_{\mathrm{out}})}{m\,c_v} \tag{6.12}$$

□

MathModelica Model

```
Model[CtrlVolume,
 Extends[ThermalTwoPin];
 ■;
 ■;
 Parameter Volume V == 1;
 Mass m;
 Temperature T;
 Equation[
  p1 == p2;
  m′ == w1 + w2;
        w1 (cv (T1 - T) + R T1) + w2 (cv (T2 - T) + R T2)
  T′ == ------------------------------------------------- ;
                             m cv
         R T m′   m R T′
  p1′ == ------ + ------ ;
           V        V
  q1 == If[w1 < 0,
      1
     ---- w1 cp T
     1000
     ,
      1
     ---- w1 cp T1
     1000
    ];
  q2 == If[w2 < 0,
      1
     ---- w2 cp T
     1000
     ,
      1
     ---- w2 cp T2
     1000
    ]
  ]
 ]
```

6.8.1 Validation

Figure 6.28 shows how temperature and air mass flow build up works for a single volume. If two control volumes, with volume V_1 and V_2, are connected in series they will work as one control volume, with volume $V = V_1 + V_2$.

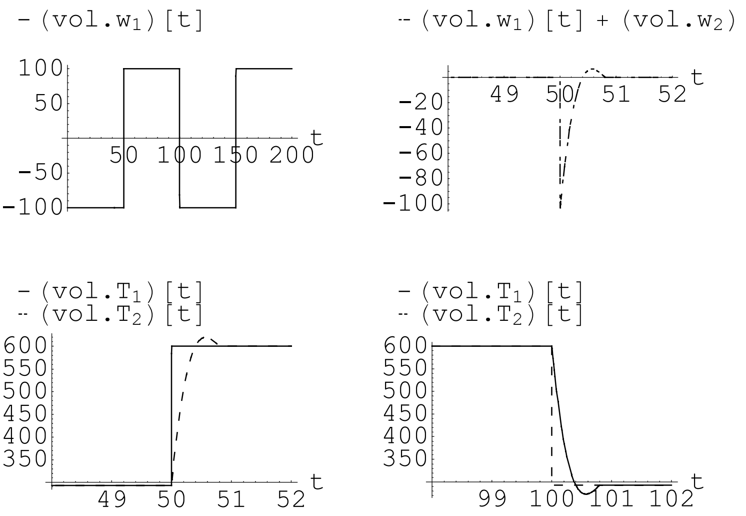


Figure 6.28. Upper left shows the air mass flow which is given as an input signal to the volume. Upper right shows the difference between in- ant outflow. The two lower figures show how the temperature changes over time.

6.9 RotationalAxis

The turbine shaft is used to connect the compressor with the turbine. Instead of using the shaft from Modelicas standard library a shaft with different connector is used.

MathModelica Model

```
Model[TurbineShaft,
 ■;
 ■;
 Flange_aa a ■;
 Flange_bb b ■;
 Parameter Inertia J == 10^-5;
 Protected[
  AngularVelocity ω;
 ];
 Equation[
  ω == a.ω;
  ω == b.ω;
  J ω ω′ == a.Wdot + b.Wdot
 ]
]
```

6.10 Engine

The power in a vehicle engine is produced by a compression cycle in a combustion chamber. Air flows into the system and enters the combustion chamber. During the cycle fuel is added, thus adding energy to the system. This energy is used to produce a torque to the driveline. The produced energy depends on the amount of air that enters the cylinders. To increase the amount of air that enters the compressor is used, this is supplied with energy through the turbine shaft by the turbine, which in it's turn is driven by the exhaust gases.

It is possible to insert volumes between each component to add dynamics to the system. The most common way to do it is to either have no volumes, or to have lumped volumes at the intake and exhaust manifold respectively. These to versions, see Figures 6.32 and 6.38, are presented and simulated below.

Remarks on Stability

Because of numerical instability in the solvers it might be difficult to introduce other volumes to the engine. It is especially cumbersome to connect volumes between the air filter and compressor, and compressor and intercooler. For this cases the initial value problem becomes unstable, and to avoid this the models need to be more carefully developed, to avoid singularities and non linearities.

To avoid numerical problems as far as possible units have been chosen so that the values should be of the same magnitude, see Section 5.2.

MathModelica Model, Three Volume Engine

```
Model[ThreeVolumeEngine,
 ■;
 ■;
 Reservoir ambient_in[{p_res == 101.3`, T_res == 296}] ■;
 AirFilter
  airfilter[{w_1[{Start == 16}], p_1[{Start == 101.3`}],
    k_1 == 1}] ■;
 CtrlVolume
  vol_af[{w_1[{Start == 16}], p_1[{Start == 101.1`}],
    V == 5, T[{Start == 305}], m[{Start == (101 5)/(287 305)}]}] ■;
 Compressor
  compressor[{w_1[{Start == 16}], p_2[{Start == 102}],
    η_c == 0.75`}] ■;
 Intercooler
  intercooler[{w_1[{Start == 16}], p_2[{Start == 102}],
    T_2[{Start == 295}]}] ■;
 ButterFlyValve
  throttle[{w_1[{Start == 16}], p_2[{Start == 50}]}] ■;
 CtrlVolume
  vol_man[{w_1[{Start == 16}], p_1[{Start == 50}], V == 6,
    T[{Start == 300}], m[{Start == (50 6)/(287 300)}]}] ■;
 BaseEngine
  baseengine[{w_1[{Start == 16}], p_1[{Start == 50}],
    p_2[{Start == 107}], T_1[{Start == 300}],
    T_2[{Start == 900}]}] ■;
 CtrlVolume
  vol_exh[{w_1[{Start == 16}], p_1[{Start == 103}],
    c_p == 1020, R == 236, V == 2, T[{Start == 900}],
    m[{Start == (103 2)/(236 1000)}]}] ■;
 Turbine
  turbine[{w_1[{Start == 16}], c_p == 1020, R == 236,
    η_t == 0.6`}] ■;
 ExhaustSystem
```

```
  exhaust[{w_1[{Start == 16}], c_p == 1010, R == 260,
    p_2[{Start == 101.3`}]}] ■;
TurbineShaft shaft[{J == 1/10^5, ω[{Start == 100}]}] ■;
Reservoir ambient_out[{w_1[{Start == 16}], p_res == 101.3`}]
  ■;
Modelica.Electrical.Analog.Basic.Ground Ground1 ■;
Modelica.Electrical.Analog.Sources.RampVoltage
  RampVoltage1[{offset == 0.2`, duration == 50}] ■;
Modelica.Blocks.Sources.Pulse
  Pulse1[{period == {100}, offset == {800/60},
    amplitude == {800/60}}] ■;
Modelica.Electrical.Analog.Sources.PulseVoltage
  Pulse2[{period == 200, startTime == 100, V == -0.8`}] ■;
Equation[
  Connect[ambient_in.a, airfilter.a] ■;
  Connect[turbine.b, exhaust.a] ■;
  Connect[exhaust.b, ambient_out.a] ■;
  Connect[turbine.c, shaft.a] ■;
  Connect[shaft.b, compressor.c] ■;
  Connect[RampVoltage1.p, Pulse2.n] ■;
  Connect[Pulse1.outPort, baseengine.InPort1] ■;
  Connect[airfilter.b, vol_af.a] ■;
  Connect[vol_af.b, compressor.a] ■;
  Connect[compressor.b, intercooler.a] ■;
  Connect[intercooler.b, throttle.a] ■;
  Connect[Ground1.p, RampVoltage1.n] ■;
  Connect[throttle.PositivePin1, Pulse2.p] ■;
  Connect[throttle.NegativePin1, RampVoltage1.n] ■;
  Connect[throttle.b, vol_man.a] ■;
  Connect[vol_man.b, baseengine.a] ■;
  Connect[turbine.a, vol_exh.b] ■;
  Connect[vol_exh.a, baseengine.b] ■
 ]
]
```

6.10.1 Validation

It is possible to connect volumes between all components along the air flow. A script has been made in *Mathematica* which tests all 32 possible engine setups, with or without volumes between the air filter, compressor, intercooler, throttle, base engine, and turbine. The throttle is supplied with an input voltage according and the engine with an input engine speed according to Figures 6.30 and 6.31 respectively.

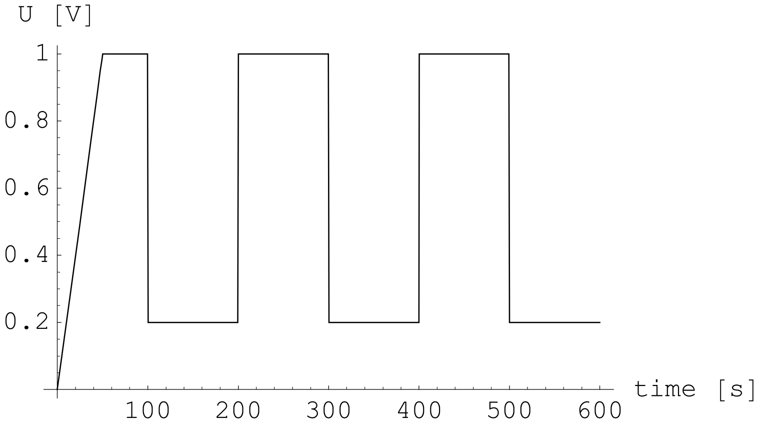


Figure 6.29. Input voltage to throttle at engine validation.

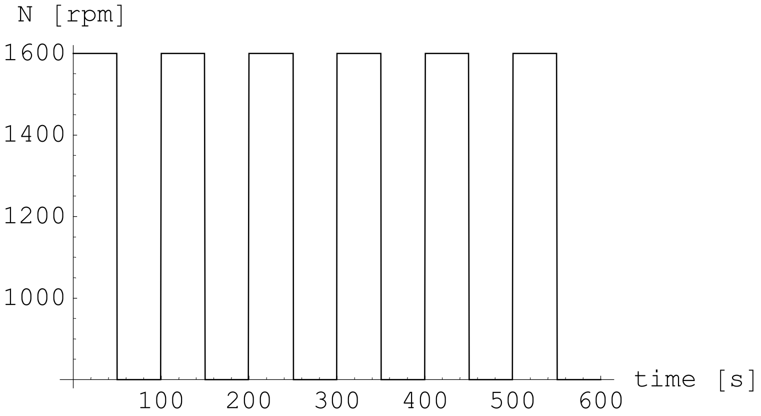


Figure 6.30. Input engine speed to base engine at validation.

It can be noted that as soon as a volume is connected to the compressor numerical problems will appear, but for most other cases the engine works fine. Validations of engines without volume and with volumes at intake and exhaust manifold are showed in the figures below. Note the transients that appears for the latter case. Finally a model without turbo is tested with the same input signals.

No Volumes

In this example an engine without any volumes connected to it is simulated. The connection diagram is shown in Figure 6.31. The following figures, Figure 6.32-6.35, shows simulation results for the test cycle given above.

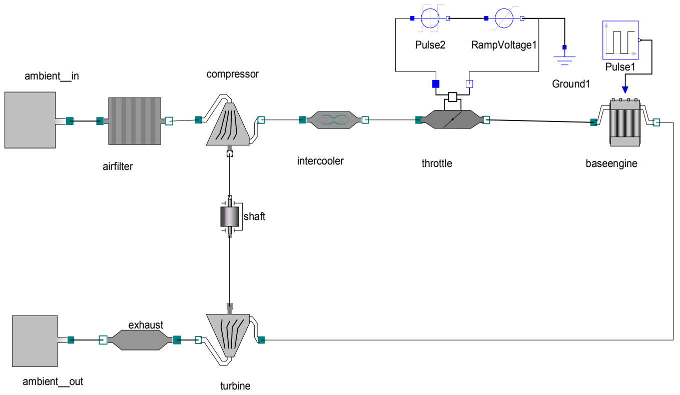


Figure 6.31. Engine Model without volumes.

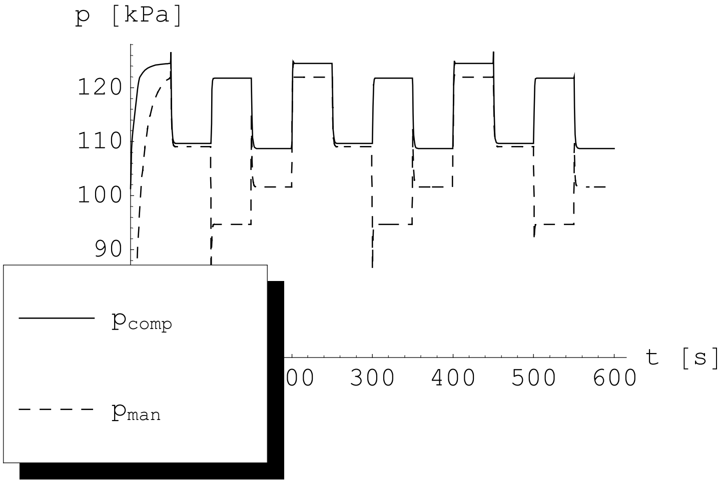


Figure 6.32. Pressure at engine intake side, p_{comp} is the pressure out from the compressor and p_{man} is the pressure in to the cylinders.

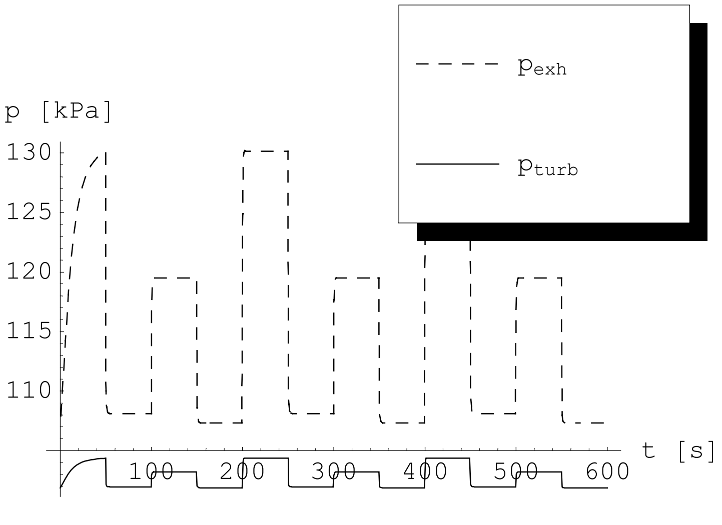


Figure 6.33. Pressure in to the turbine, p_{exh}, and out of it p_{turb}.

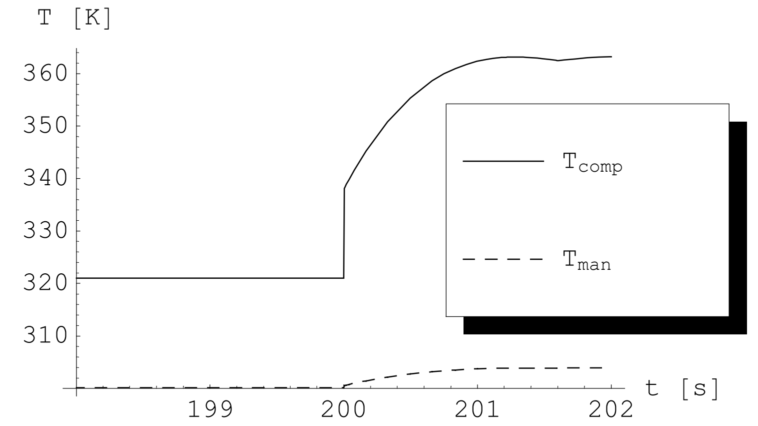


Figure 6.34. Temperature at engine intake side around t = 200 [s]. T_{comp} is the temperature at compressor outlet and T_{man} is the temperature at inlet manifold.

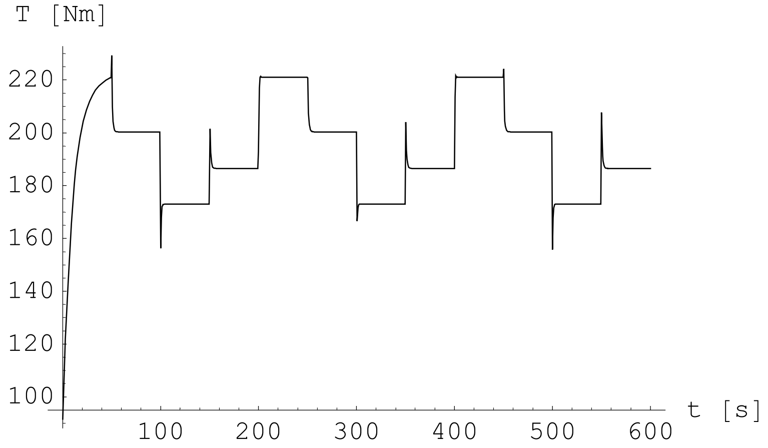


Figure 6.35. Produced engine torque during test cycle.

With Volumes

In this version three volumes are added to the model. The first volume, of 5 liters, is connected between air filter and compressor, the second volume of 6 liters is the manifold volume and finally the exhaust volume of 2 liters is added. The connection diagram is shown in Figure 6.38. The following figures, Figure 6.37-6.41, shows simulation results for the test cycle given above.

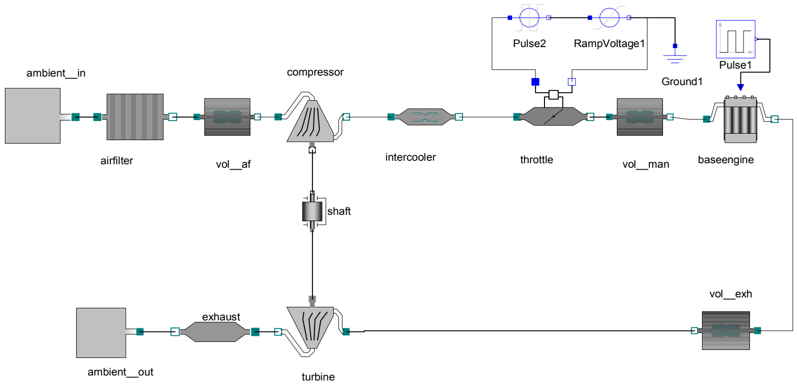


Figure 6.36. Engine Model with three volumes.

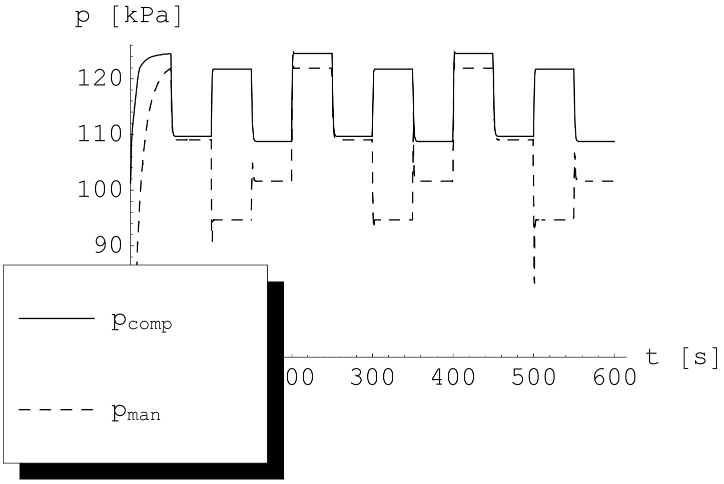


Figure 6.37. Pressure at engine intake side, p_{comp} is the pressure out from the compressor and p_{man} is the pressure in to the cylinders.

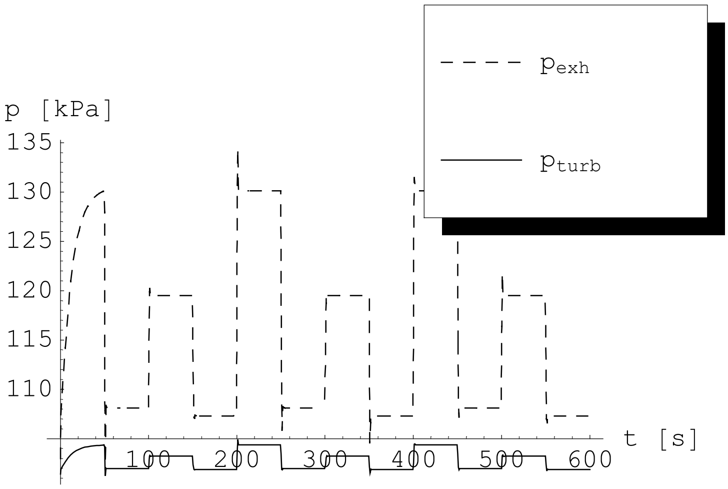


Figure 6.38. Pressure in to the turbine, p_{exh}, and out of it p_{turb}.

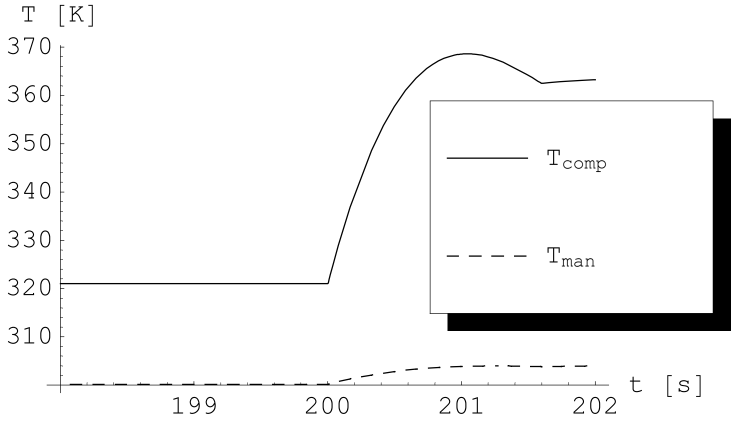


Figure 6.39. Temperature at engine intake side around t = 200 [s]. T_{comp} is the temperature at compressor outlet and T_{man} is the temperature at inlet manifold.

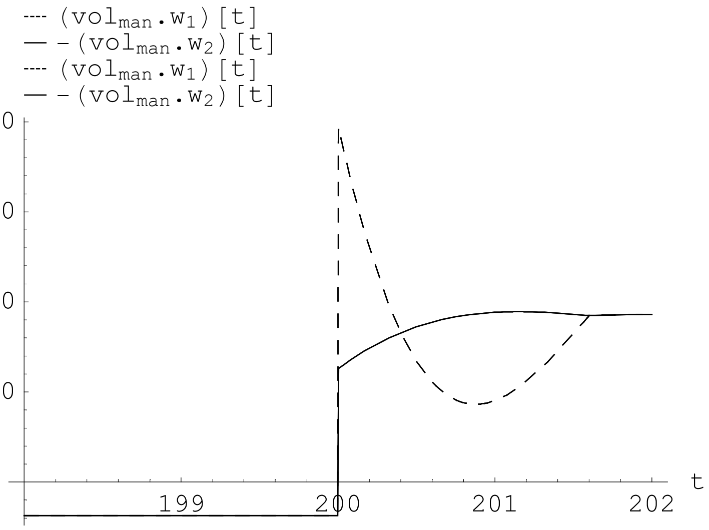


Figure 6.40. Delay of air mass flow through the manifold volume.

```
PlotSimulation[baseengine.M[t], {t, 198, 202},
  PlotStyle → {Dashing[{0.02, 0.02}], Dashing[{0}]},
  SimulationResult → {resNV, resV}];
```

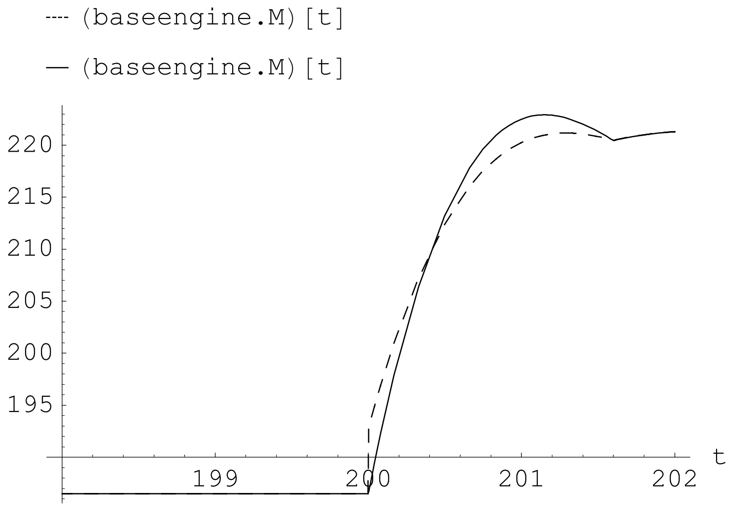


Figure 6.41. *Produced engine torque around t=200 s for the engine without volumes (dotted line) and the engine with volumes.*

No Turbo

A connection diagram of an engine without turbo is shown in Figure 6.42. The only difference between this engine and the turbo engine with volumes is that it lacks the turbo and intercooler, all other components are exactly the same. This should result in an engine that can not produce the same output torque. An engine which has exactly the same components as a turbo engine, except the turbo, should of course produce lower torque, this is shown in Figure 6.43.

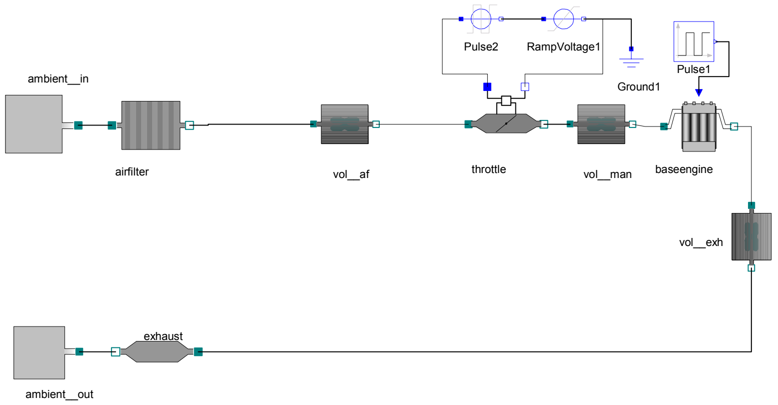


Figure 6.42. Engine Model without turbo.

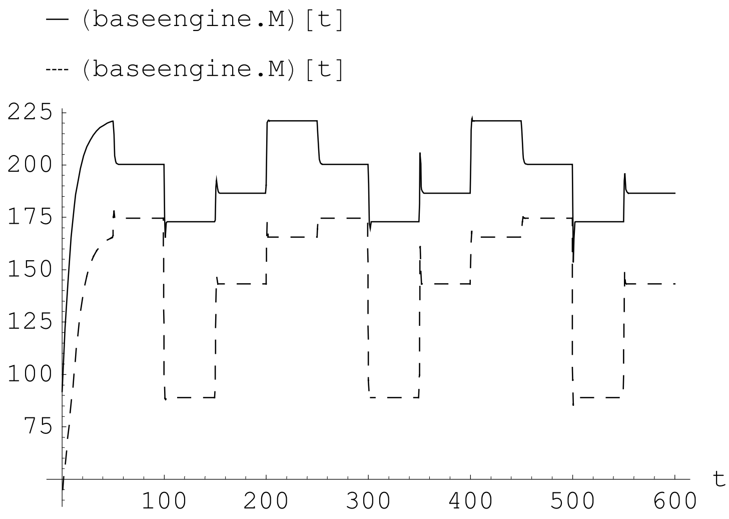


Figure 6.43. Output torque from engine without turbo (dotted) compared to an engine with turbo.

7. Conclusions

A Mean Value Engine Model has been presented in this document. It is developed in Modelica, making it possible to take advantage of a component based approach. It is shown that the engines qualitative behavior corresponds to a real engine behavior. It is shown that the models works independently on flow direction and that it is possible to choose different degrees of complexity by adding or deleting components, such as the entire turbo or volumes.

Components that are not developed -or tested- in this notebook, but that are interesting to look at in future projects are, among others; Exhaust Gas Recirculation (EGR), basically a butterflyvalve and intercooler, waste gate, consisting of a valve and control system, fuel dynamics, lambda controller, ECU. It is also interesting to try to connect the engine to an entire driveline such as the one developed in [Nobrant, 2001].

More general models for the gas properties should also be developed, this is done by [Silverlind, 2001] in his forthcoming master thesis. In this master thesis the approach with pWH-connector, described earlier on in this document is also tested. Furthermore performance maps for e.g. volumetric efficiency and turbine efficiency are also introduced.

As described in Chapter 6 there are a few problems with numerical stability. It is very interesting to study this in detail in the future. One thing that could be done is introduce the noEvent() operator, which is an built-in operator in Modelica, used to prevent time events to be triggered. Introducing Modelicas "*nominal*" attribute would probably make it easier for the simulation engine to solve the initial value problem. Furthermore the restriction should be modeled more carefully, so it can handle both laminar and turbulent flow. This will eliminate the root function for small flows, thus probably increasing numerical stability.

Bibliography

Bergström, Johan & Brugård, Jan
Modeling of a Turbo Charged Ark Ignited Engine
University of Linköping ISY
1999
Reg nr: LiTH-ISY-EX-2081
Download: pdf-file, ps-file

Eriksson, Lars
Thermodynamics of Unsteady Flows in Open System (Title might change)
University of Linköping ISY
2000

Eriksson, Lars & Nielsen, Lars & Brugård, Jan & Bergström, Johan & Pettersson, Fredrik & Andersson, Per
Modeling and Simulation of a Turbo Charged SI Engine
Elsevier Science, 3rd IFAC Workshop "Advances in Automatic Control"
2001

Fritzon, Peter
Forthcoming book; Modelica -the Language
2001

Jones J.B. & Dugan, R.B.
Engineering Thermodynamics
Prentice Hall, Englewood Cliffs
1996
ISBN 0-02-361332-7

Krysander, Mattias
Luftflöde förbi en trottel
University of Linköping ISY
2000
Reg nr: LiTH-ISY-EX-3068
Download: pdf-file, ps-file

Lakshminarayana, Budugur
Fluid Dynamics and Heat Transfer of Turbomachinery
John Wiley & Sons
1996
ISBN 0-471-85546-4

Müller, M. & Hendricks, E. & Sörenson, S.C.
Mean Value Modeling of Turbocharged Spark Ignited Engines
Society of Automotive Engineers
1998
SP-1330

Nakayama, Y. & Boucher, R.F.
Introduction to Fluid Mechanincs
Arnold
1999
ISBN 0-07-114214-2

Nielsen, Lars & Eriksson, Lars
Course Material Vehicular Systems
University of Linköping ISY
2000

Nobrant, Per
Driveline Modeling using MathModelica
University of Linköping ISY
2001
Download: pdf-file, ps-file

Petterson, Fredrik
Simulation of a Turbo Charged Spark Ignited Engine
University of Linköping ISY
2000
Reg nr: LiTH-ISY-EX-3010
Download: pdf-file, ps-file

Silverlind, Daniel
Modeling of a Turbo Engine
University of Linköping ISY
2001
Reg nr: Not available yet.

Tiller, Michael M.
Introduction to Physical Modeling with Modelica
Kluwer Academic Publishers

2001
ISBN 0-7923-7367-7

Tummescheit, Hubertus & Eborn, Jonas
Thrmofluid homepage
http://www.control.lth.se/~hubertus/ThermoFluid/