Modeling of a moist gas saturation in Thermoptim

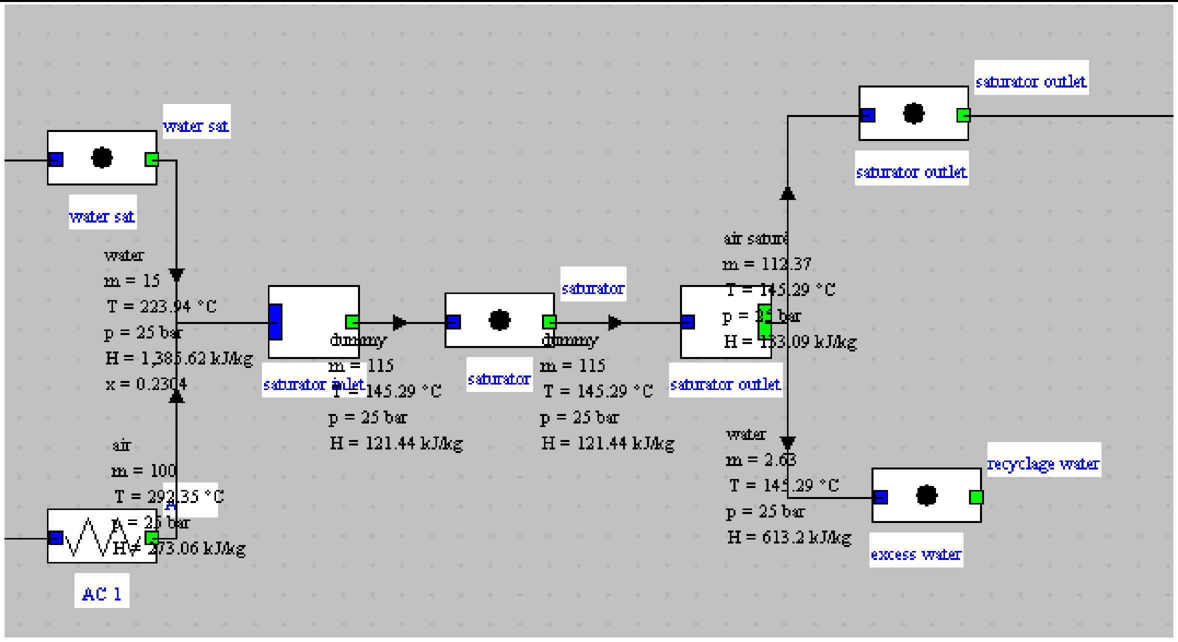
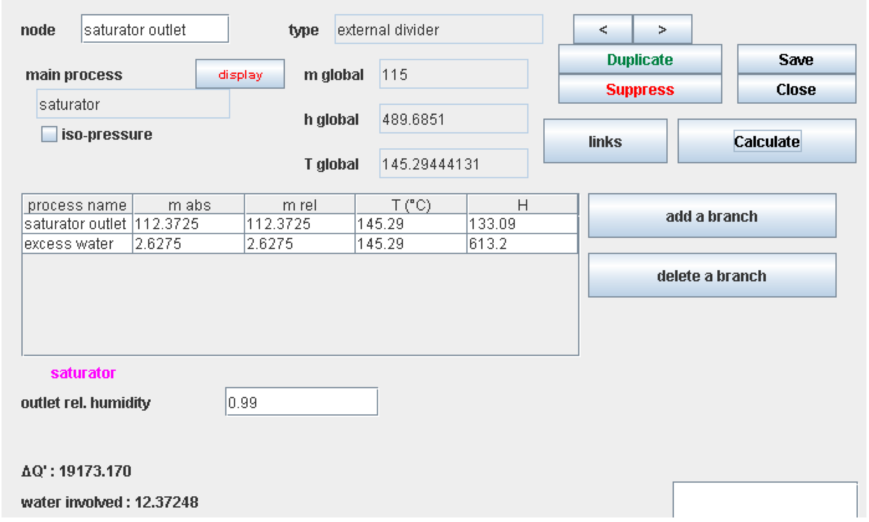


Figure 1: Saturator

A saturator has the distinction of being crossed by two separate streams: air and water, that exchange matter and energy through an interface. It behaves like a quadrupole receiving two input fluid, and of out which come two others (Figure 1).



node: saturator outlet

type: external divider

main process: saturator

☐ iso-pressure

m global: 115

h global: 489.6851

T global: 145.29444131

process name	m abs	m rel	T (°C)	H
saturator outlet	112.3725	112.3725	145.29	133.09
excess water	2.6275	2.6275	145.29	613.2

saturator

outlet rel. humidity: 0.99

ΔQ' : 19173.170

water involved : 12.37248

Figure 1: Saturator component screen

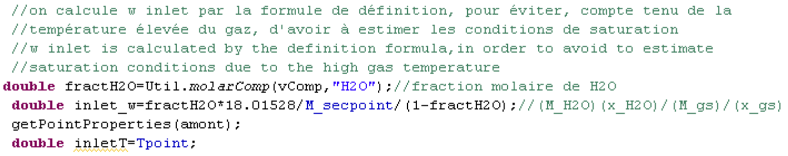
The saturator behaves like a moist mixer, and is calculated as such.

The class code (Saturator.java) is as follows:

1) we begin by getting the incoming gas composition, and we update the composition of the outlet gas, a precaution in case the two dry gases would be different.

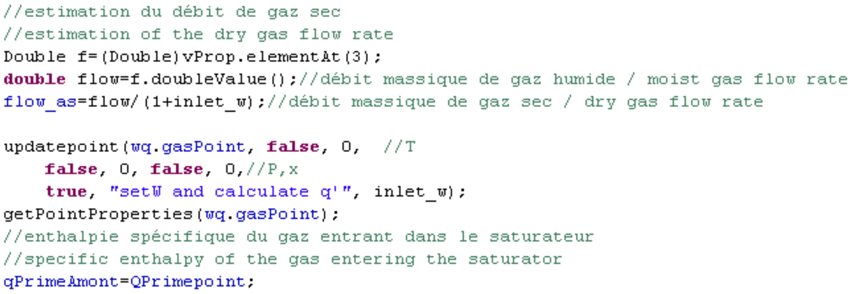
```
//Propriétés humides du gaz entrant
//moist properties of the inlet gas
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=wq.gasProcess;//name of the process (see method getProperties(String[] args))
Vector vProp=proj.getProperties(args);
String amont=(String)vProp.elementAt(2);//gets the downstream point name
getPointProperties(amont);//direct parsing of point property vector
getSubstProperties(nomCorps);
double M=Msubst;
Vector vComp=lecorps.getGasComposition();
outletGasSubstance.updateGasComposition(vComp);//on met à jour la composition du gaz en sortie
//the outlet gas composition is updated
```

2) we calculate the inlet absolute humidity with the definition formula, to avoid, given the high temperature of gas, having to estimate the saturation conditions.



```
 //on calcule w inlet par la formule de définition, pour éviter, compte tenu de la
 //température élevée du gaz, d'avoir à estimer les conditions de saturation
 //w inlet is calculated by the definition formula,in order to avoid to estimate
 //saturation conditions due to the high gas temperature
double fractH2O=Util.molarComp(vComp,"H2O");//fraction molaire de H2O
 double inlet_w=fractH2O*18.01528/M_secpoint/(1-fractH2O);//(M_H2O)(x_H2O)/(M_gs)/(x_gs)
 getPointProperties(amont);
 double inletT=Tpoint;
```

3) the dry gas flow and the inlet gas specific enthalpy are determined, using methods updatepoint() and getPointProperties():

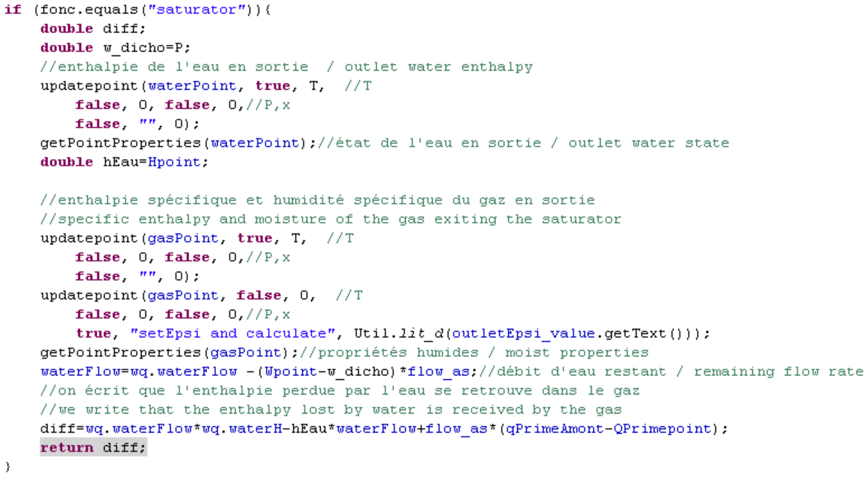


```
//estimation du débit de gaz sec
//estimation of the dry gas flow rate
Double f=(Double)vProp.elementAt(3);
double flow=f.doubleValue();//débit massique de gaz humide / moist gas flow rate
flow_as=flow/(1+inlet_w);//débit massique de gaz sec / dry gas flow rate

updatepoint(wq.gasPoint, false, 0,  //T
    false, 0, false, 0,//P,x
    true, "setW and calculate q'", inlet_w);
getPointProperties(wq.gasPoint);
//enthalpie spécifique du gaz entrant dans le saturateur
//specific enthalpy of the gas entering the saturator
qPrimeAmont=QPrimepoint;
```

4) At this stage, the upstream moist gas properties are perfectly calculated. We must now determine the saturator exit temperature, solving simultaneously:

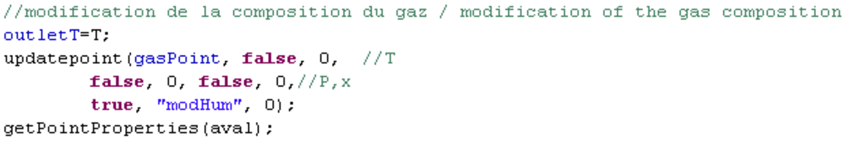
- water balance (the flow of water consumed is equal to the product of the dry gas flow by the gas moisture variation);
- enthalpy balance (the sum of incoming enthalpy flows (specific units for the moist gas) is equal to the sum of outgoing enthalpy flows. Since Ts is unknown, we do a solution search by dichotomy, using generic function Util.dicho_T(), which uses f_dicho(). The code operates as follows:
- humidity is passed as input argument, instead of the pressure, otherwise known;
- we first calculate the enthalpy heau (Ts) of water at the outlet;
- we change the gas outlet temperature, then its moisture from the value read on the screen, and we get the values of its absolute humidity and specific enthalpy;
- we calculate the flow of water left (although we should do a test to make sure it remains positive);
- we write that the enthalpy lost by the water ends up in the air, and calculate residue diff;
- temperature Ts is determined when diff = 0.



```
if (fonc.equals("saturator")){
    double diff;
    double w_dicho=P;
    //enthalpie de l'eau en sortie  / outlet water enthalpy
    updatepoint(waterPoint, true, T,  //T
        false, 0, false, 0,//P,x
        false, "", 0);
    getPointProperties(waterPoint);//état de l'eau en sortie / outlet water state
    double hEau=Hpoint;

    //enthalpie spécifique et humidité spécifique du gaz en sortie
    //specific enthalpy and moisture of the gas exiting the saturator
    updatepoint(gasPoint, true, T,  //T
        false, 0, false, 0,//P,x
        false, "", 0);
    updatepoint(gasPoint, false, 0,  //T
        false, 0, false, 0,//P,x
        true, "setEpsi and calculate", Util.lit_d(outletEpsi_value.getText()));
    getPointProperties(gasPoint);//propriétés humides / moist properties
    waterFlow=wq.waterFlow -(Wpoint-w_dicho)*flow_as;//débit d'eau restant / remaining flow rate
    //on écrit que l'enthalpie perdue par l'eau se retrouve dans le gaz
    //we write that the enthalpy lost by water is received by the gas
    diff=wq.waterFlow*wq.waterH-hEau*waterFlow+flow_as*(qPrimeAmont-QPrimepoint);
    return diff;
}
```

5) Ts being determined, we change the composition of moist gas at the outlet:



```
//modification de la composition du gaz / modification of the gas composition
outletT=T;
updatepoint(gasPoint, false, 0,  //T
        false, 0, false, 0,//P,x
        true, "modHum", 0);
getPointProperties(aval);
```

6) Consistency checks

Using the values that appear on the diagram of Figure 1, we can build the apparent saturator balance:

	kW
total inlet enthalpy flow	65,012
total output enthalpy flow	22,570
apparent discrepancy	42,443
difference per kg/s of water consumed	L_{water} = 2,547

Everything happens as if 42.4 MW of heat disappeared but, as shown in the last row of the table, this value corresponds exactly to the enthalpy of vaporization of water in the moist gas, which is not recognized in the values displayed by Thermoptim given the conventions adopted for ideal gases (zero enthalpy at 25 °C and 1 bar).

Obviously, if we sufficiently cooled exhaust gases for the water they contain to condense, this enthalpy would appear again (with the addition of the water formed in the combustion chamber).