Modélisation dans Thermoptim de la trempe d'un gaz (water quench)

La trempe d'un gaz de synthèse (water quench") est une forme d'humidification de gaz par lavage qui permet à la fois de le nettoyer d'impuretés du type cendres et goudrons du fait qu'il se refroidit et de le saturer d'eau.

Cette opération présente la particularité de mettre en jeu deux flux séparés : le gaz de synthèse et l'eau, qui échangent de la matière et de l'énergie par l'intermédiaire d'une interface. Elle se comporte donc comme un quadripôle recevant deux fluides en entrée, et dont en sortent deux autres.

Ceci pose une petite difficulté pour en réaliser un modèle dans Thermoptim, étant donné que les seuls composants disponibles sont soit des transfos, soit des nœuds. La solution est de former le quadripôle en associant un mélangeur en entrée et un diviseur en sortie, les deux étant reliés par une transfo-point qui joue un rôle purement passif.

Pour que le modèle soit bien cohérent, on synchronise les calculs effectués par les deux nœuds. Plus précisément le diviseur de sortie prend le contrôle du mélangeur, dont le rôle se limite à effectuer une mise à jour des variables de couplage associées aux flux d'entrée. La structure du modèle est donnée figure 1.

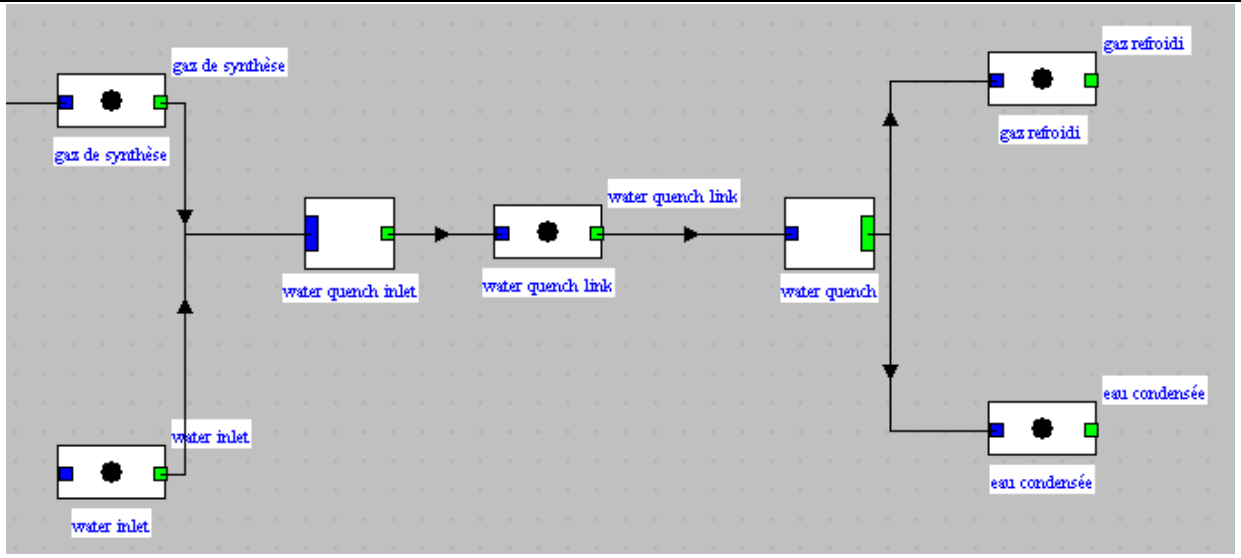

Figure 1 : Schéma du composant water quench

Modèle de la trempe du gaz de synthèse

Le modèle que nous développons ici est particulièrement simple : il se contente d'établir les bilans massiques et énergétiques, en supposant connues la température du gaz et son humidité relative en sortie du composant. Les débits des deux flux entrants sont imposés par les conditions en amont du composant et non pas recalculés. Si le débit d'eau est insuffisant pour que son échauffement (jusqu'à la température du gaz sortant) permette d'extraire du gaz l'enthalpie requise, un message en avertit l'utilisateur.

Les fonctions de calcul des propriétés humides des gaz et des points de Thermoptim ont été rendues accessibles depuis les classes externes. Nous vous conseillons de vous référer à la note : "Calculs des gaz humides depuis les classes externes" pour une présentation détaillée des méthodes disponibles.

Le modèle que l'on peut retenir est alors le suivant :

1) le seul paramètre est la valeur de l'humidité relative ε en sortie de composant, lue à l'écran (*a priori* très proche de 1) ;
2) on commence par calculer l'humidité absolue du gaz entrant, et on détermine le débit-masse de gaz sec à partir de celui du gaz humide ;
3) l'humidité relative ε en sortie est imposée, et les propriétés humides du gaz de sortie sont calculées, ce qui fournit les enthalpies spécifique et totale à extraire du gaz
4) le débit d'eau fourni par le gaz est déterminé et la composition du gaz humide en sortie est modifiée

5) le bilan enthalpique sur l'eau fournit sa température de sortie, qui doit être inférieure à celle du gaz sortant
6) les valeurs en aval du noeud sont mises à jour

Dans Thermoptim, le composant est comme on l'a dit représenté par un mélangeur externe connecté à un diviseur externe, les calculs étant effectués par ce dernier. Les classes s'appellent WaterQuenchInlet et WaterQuench. L'écran du composant est donné figure 2.

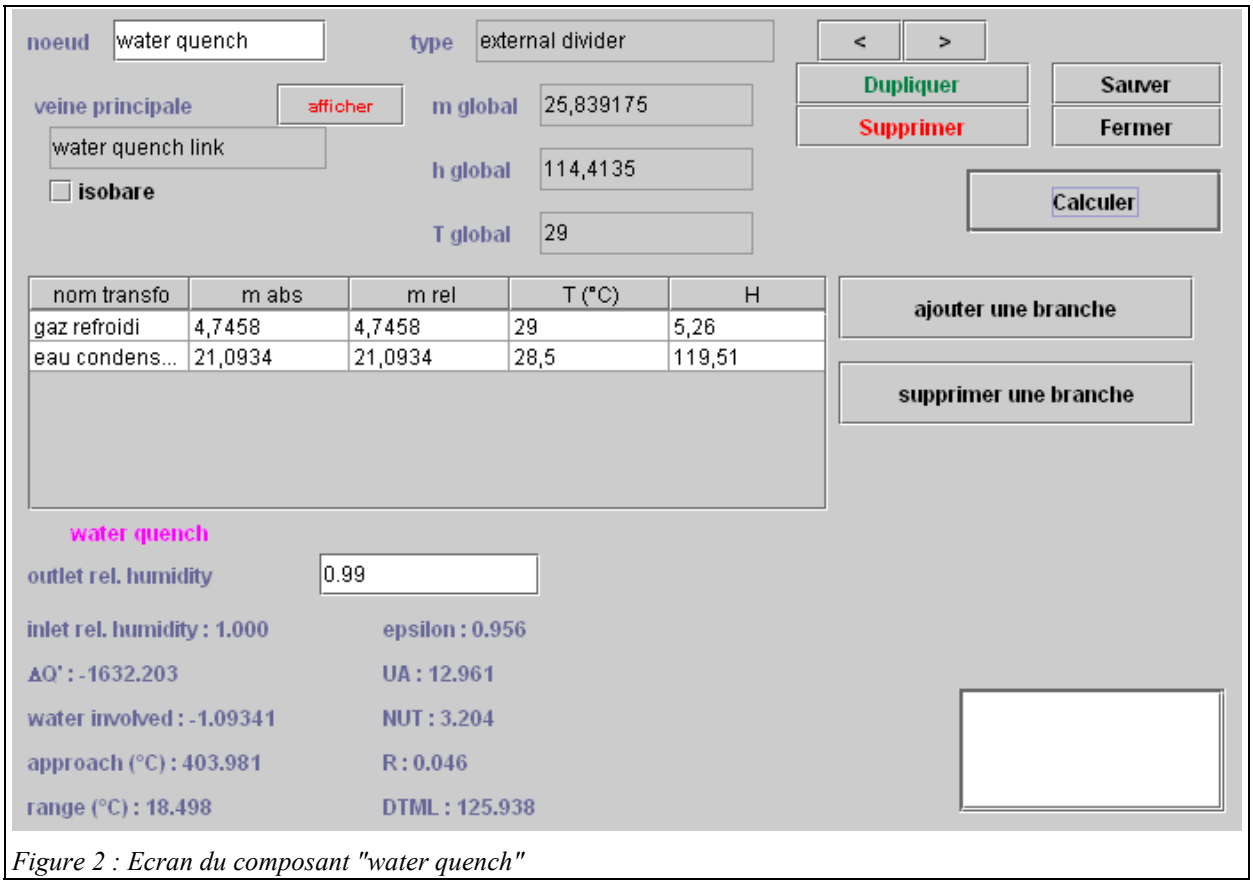

Figure 2 : Ecran du composant "water quench"

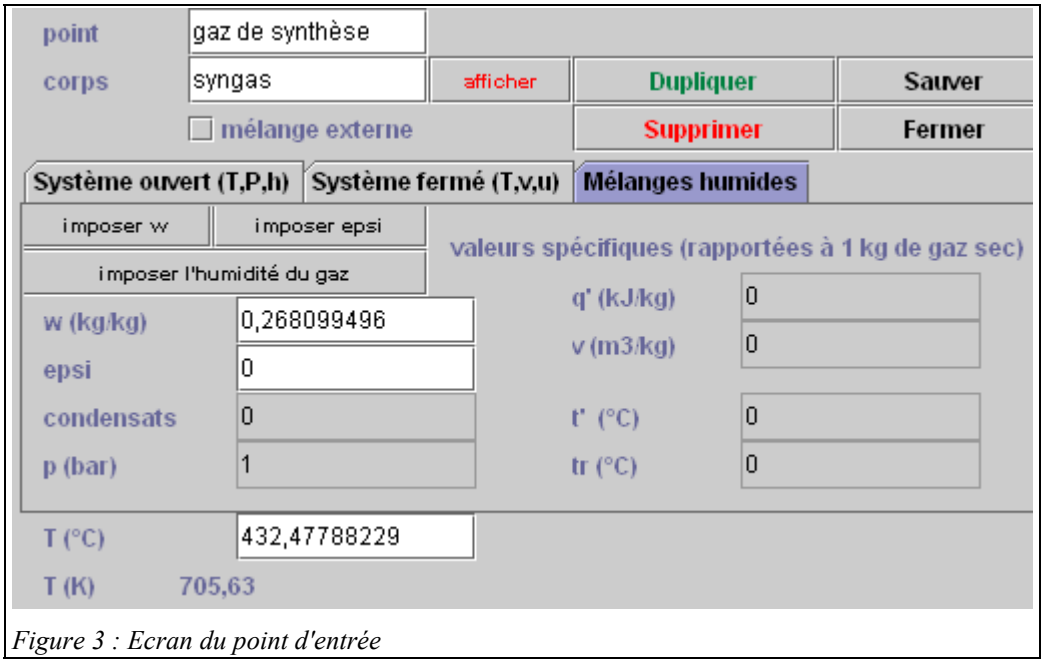

Figure 3 : Ecran du point d'entrée

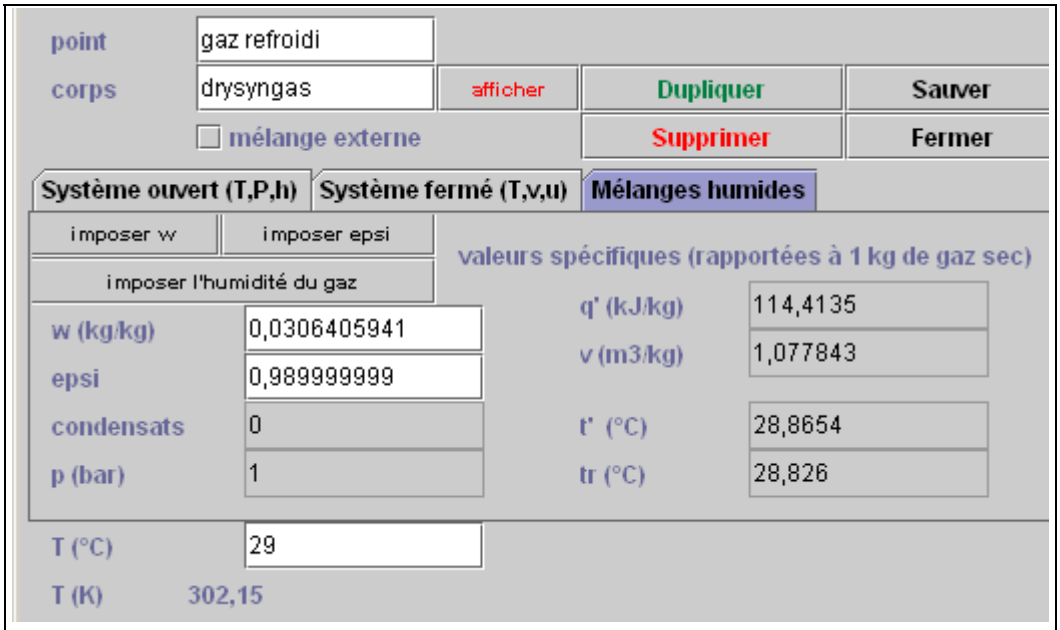

Figure 4 : Ecran du point de sortie

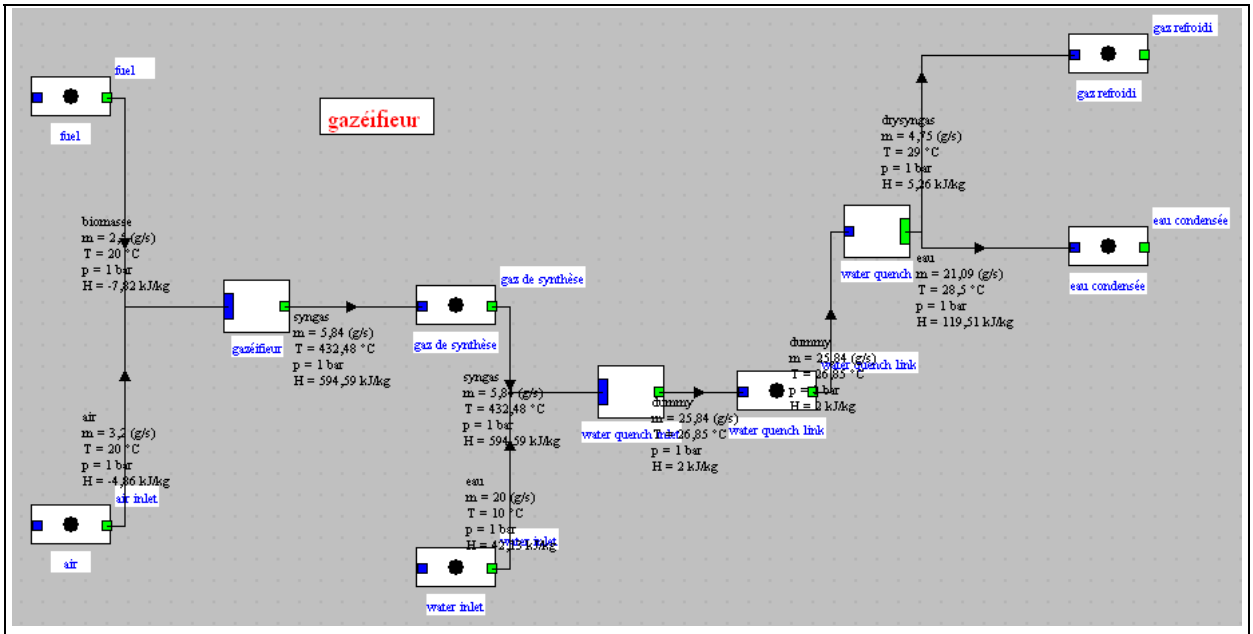

Figure 5 : Schéma du composant "water quench"

La condensation d'une partie de l'eau a pour effet d'augmenter le PCI du gaz (figure 6).

nom du composant	fraction molaire	fraction massique
CO2	0,1663779	0,3048321
H2O	0,03963997	0,02972966
O2	0	0
N2	0,4379945	0,5107991
CO	0,1079439	0,1258732
H2	0,2430099	0,02039408
Ar	0,005033745	0,008371891

Figure 6 : composition du gaz de synthèse refroidi (PCI : 3,72 MJ/kg)

Etude de la classe externe WaterQuench

Pour assurer la cohérence du modèle (éviter que l'on connecte le mélangeur d'entrée à un diviseur de sortie inadéquat), chacun des deux nœuds essaie d'instancier l'autre en recherchant sa classe parmi les composants externes du projet, et vérifie que tous deux sont bien connectés à la même transfo de liaison. Si l'opération échoue, un message avertit l'utilisateur que la construction est incorrecte. Cette vérification est effectuée par les méthodes setupOutlet() et setupInlet().

De surcroît, des tests de cohérence de chaque nœud sont effectués par la méthode checkConsistency() pour vérifier que les fluides connectés sont les bons : dans ce cas, un gaz humide et de l'eau en entrée et en sortie. On se reportera au tome 3 du manuel de référence pour les explications sur ce point, valables pour tous les nœuds externes.

L'étude de la classe externe DirectCoolingTower permet de voir comment le modèle a été implémenté. Six étapes suffisent pour effectuer les calculs :

1) on commence par calculer l'humidité absolue du gaz entrant, et on détermine le débit-masse de gaz sec à partir de celui du gaz humide puis on initialise l'enthalpie spécifique amont :

```
//Propriétés humides du gaz entrant
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();
drySynGasSubstance.updateGasComposition(vComp);//on met à jour la composition du gaz traité

//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
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)
hamont=wq.hamont/(1+inlet_w);//passage en unités spécifiques
getPointProperties(amont);
double inletT=Tpoint;

//imposition de w et calcul des propriétés humides
Double f=(Double)vProp.elementAt(3);
double flow=f.doubleValue();//débit massique de gaz humide
double flow_as=flow/(1+inlet_w);//débit massique de gaz sec
```

2) on calcule ensuite les propriétés du gaz sortant, en lui imposant l'humidité relative lue à l'écran ; on en déduit l'enthalpie totale mise en jeu

```
//Propriétés humides du gaz sortant
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=gasProcess;//name of the process (see method getProperties(String[] args))
vProp=proj.getProperties(args);
String aval=(String)vProp.elementAt(1);//gets the upstream point name
getPointProperties(aval);
outletT=Tpoint;

double epsi=Util.lit_d(outletEpsi_value.getText());
//imposition de epsilon et mise à jour de w
updatepoint(aval, false, 0,  //T
        false, 0, false, 0,//P,x
        true, "setEpsi and calculate", epsi);
getPointProperties(aval);

double haval=QPrimepoint;//enthalpie spécifique du gaz sortant

double DeltaQprime=flow_as*(haval-hamont);//enthalpie totale acquise par l'gaz
JLabel3.setText("\u0394Q' : "+Util.aff_d(DeltaQprime,3));
```

3) on modifie la composition du gaz en sortie, et on calcule les débits-masses sortant :

```
//modification de la composition du gaz
updatepoint(aval, false, 0,  //T
        false, 0, false, 0,//P,x
        true, "modHum", 0);

//Bilans massiques gaz et eau
double outletFlow=flow_as*(1+Wpoint);

double waterFlow=wq.waterFlow - (Wpoint-inlet_w)*flow_as;
JLabel4.setText("water involved : "+Util.aff_d((Wpoint-inlet_w)*flow_as,5));
```

4) le bilan enthalpique sur l'eau fournit alors la température de l'eau sortante

```
//Bilans enthalpique sur l'eau
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=waterProcess;//name of the process (see method getProperties(String[] args))
vProp=proj.getProperties(args);
String waterOut=(String)vProp.elementAt(1);//gets the upstream point name
getPointProperties(waterOut);
double hAval=wq.waterH-DeltaQprime/waterFlow;
getSubstProperties(nomCorps);

double waterOutT=lecorps.getT_from_hP(hAval,Ppoint);//température de sortie de l'eau

if(waterOutT>outletT){
    String msg = "The water flow rate is too low. It cannot extract the enthalpy requested";
    JOptionPane.showMessageDialog(de, msg);
    isBuilt=false;
}
else{
    JLabel5.setText("approach (°C) : "+Util.aff_d(inletT-waterOutT,3));
    JLabel6.setText("range (°C) : "+Util.aff_d(waterOutT-wq.waterT,3));
```

5) le nœud est mis à jour en utilisant les méthodes génériques décrites dans le manuel de référence

```
//mise à jour du noeud en utilisant les méthodes génériques
vTransfo= new Vector[nBranches+1];
vPoints= new Vector[nBranches+1];
setupVector(gasProcess, aval, 0, outletFlow, outletT, gasP, 0);
setupVector(waterProcess, waterPoint, 1, waterFlow, waterOutT, gasP, 0);
setupVector(mainProcess, aval, 2, outletFlow+waterFlow, outletT, gasP, 0);
updateDivider(vTransfo,vPoints,outletT,haval);
```

6) on estime enfin les caractéristiques globales de l'échangeur

```
if (mCpc>mCpf){
    epsilon=Math.abs((Tfs-Tfe)/(Tfmin-Tce));
    if(epsilon>1)epsilon=Math.abs((Tfs-Tfe)/(Tfmax-Tce));
    R=mCpf/mCpc;
    NUT=NUT_epsi(epsilon,R) ;
    UA=mCpf*NUT;
}
else{
    epsilon=Math.abs((Tcs-Tce)/(Tfmin-Tce));
    if(epsilon>1)epsilon=Math.abs((Tcs-Tce)/(Tfmax-Tce));
    R=mCpc/mCpf;
    NUT=NUT_epsi(epsilon,R) ;
    UA=mCpc*NUT;
}
```