

Modélisation d'une tour de refroidissement à contact indirect dans Thermoptim

Une tour de refroidissement à contact indirect présente la particularité d'être traversée par deux flux séparés : de l'air et de 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.

Pour la représenter dans Thermoptim, on forme ce 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.

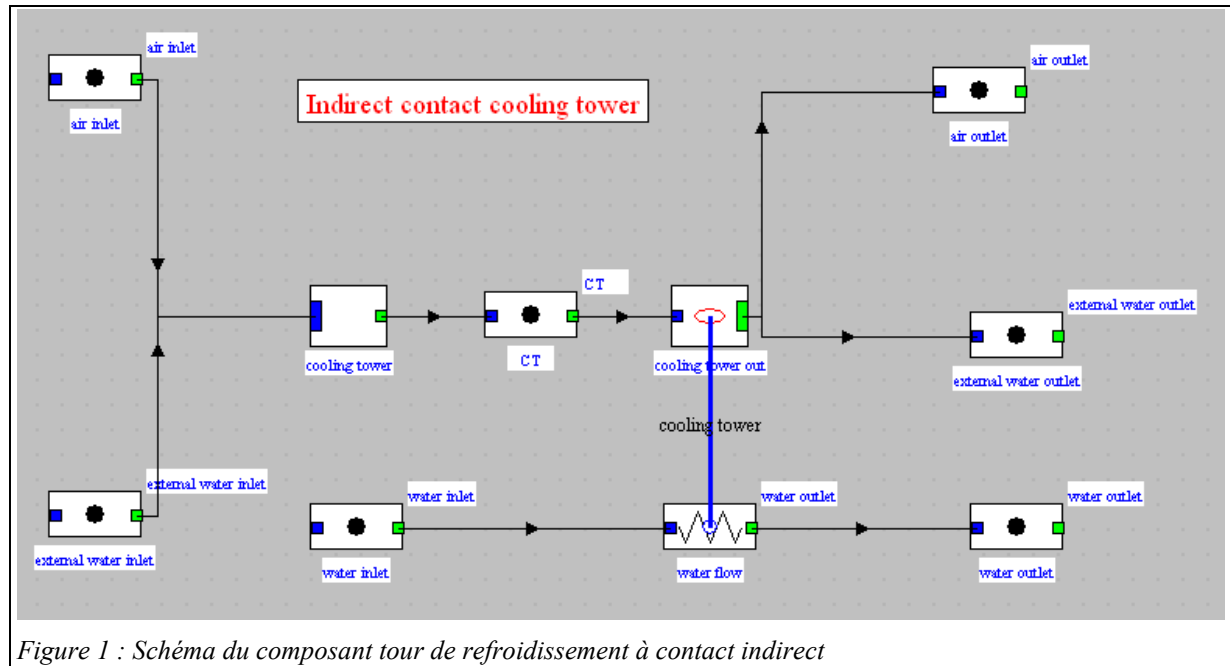


Figure 1 : Schéma du composant tour de refroidissement à contact indirect

Modèle de la tour de refroidissement à contact indirect

Le modèle est basé sur un raisonnement enthalpique global où sont supposées connues les conditions d'entrée et de sortie côté air et celles d'entrée côté eau, et les résultats du modèle sont cohérents avec les valeurs expérimentales et celles fournies par les constructeurs.

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 refroidissement (jusqu'à la température humide de l'air entrant) permette de fournir à l'air l'enthalpie requise, un message 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 externes¹" pour une présentation détaillée des méthodes disponibles.

Le modèle que l'on peut retenir est alors le suivant, lorsque l'on connaît la température de sortie de l'air :

- 1) on commence par calculer les propriétés humides de l'air entrant, et on détermine le débit-masse de gaz sec à partir de celui du gaz humide ; l'humidité relative ε en entrée de tour est affichée à l'écran
- 2) l'humidité relative ε en sortie de tour est lue à l'écran, et les propriétés humides de l'air de sortie sont calculées, ce qui fournit les enthalpies spécifique et totale à apporter à l'air
- 3) le débit d'eau emporté par l'air est déterminé et la composition de l'air humide en sortie est modifiée

¹ <http://www.thermoptim.org/sections/base-methodologique/extensions-thermoptim/calculs-gaz-humides>

- 4) le bilan enthalpique fournit la charge du thermocoupleur, dont le calcul donne la température de sortie du fluide à refroidir
- 5) les valeurs en aval du nœud sont mises à jour

Ce modèle, très simple, ne comporte aucune estimation du dimensionnement de la tour, du type calcul d'un NUT. Il serait possible de le compléter sur ce point, de manière analogue à ce qui est présenté dans le modèle de tour de refroidissement à contact direct.

La tour est comme on l'a dit représentée par un mélangeur externe connecté à un diviseur externe, les calculs étant effectués par ce dernier. Les classes s'appellent InDirectCoolingTowerInlet et InDirectCoolingTower.

The screenshot shows a software interface for configuring a cooling tower component. At the top, there are input fields for 'noeud' (set to 'cooling tower out') and 'type' (set to 'external divider'). Below these are fields for 'veine principale' (set to 'CT') and 'isobare' (unchecked). Global parameters are listed: 'm global' (0,927439481), 'h global' (56,9361), and 'T global' (20). A table displays stream data:

nom transfo	m abs	m rel	T (°C)	H
air outlet	0,92	0,92	20	-4,92
external water...	0,49256	0,49256	20	83,99

Below the table, there are buttons for 'ajouter une branche' and 'supprimer une branche'. Further down, there are input fields for 'inlet rel. humidity' (0.500), 'outlet rel. humidity' (0.99), 'ΔO' (20.742), and 'water involved' (0.00744). The interface also includes navigation buttons (< >), 'Dupliquer', 'Supprimer', 'Sauver', and 'Fermer', and a 'Calculer' button.

Figure 2 : Ecran du composant tour de refroidissement

Les écrans du composant tour de refroidissement et de son thermocoupleur sont donnés figure 2 et 3. On désire ici refroidir 1kg/s d'eau de 30 °C à 25 °C. Avec un débit de 0,92 kg/s d'air ambiant à 18 °C et d'humidité relative égale à 0,5, la température de l'air sortant est de 20 °C et 7,4 g/s d'eau sont évaporés. La puissance de la tour est de 20,7 kW.

nom	cooling tower	type	contre-courant	<	>	Sauver
cooling tower			Supprimer	Fermer		
thermal fluid		process		Calculer		
water flow	afficher	cooling tower out	afficher			
Te	30	Te	18			
Ts	25,04048362	Ts	20	<input type="radio"/> fluide méthode pinct. <input checked="" type="radio"/>		
m	1	m	0,49628026	pincement minimum 0		
Cp	4,18216259	Cp	10,37069658			
m ΔH	-20,74150387	m ΔH	20,74150387			
<input checked="" type="radio"/> calculate exchange epsilon 0,413293032		UA	2,76714341			
		R	0,81257973			
		NUT	0,661653714			
		DTML	8,43387568			

Figure 3 : Ecran du thermocoupleur

Les figures 4 et 5 fournissent les propriétés humides de l'air en entrée et en sortie de la tour, et la figure 6 un synoptique du modèle.

projet	cooling tower	<input type="checkbox"/> observée	<	>
point	air inlet			
corps	air_humide	afficher	Dupliquer	Sauver
<input type="checkbox"/> mélange externe		Supprimer	Fermer	
Système ouvert (T,P,h)		Système fermé (T,v,u)		Mélanges humides
<input type="checkbox"/> imposer w <input type="checkbox"/> imposer epsi <input type="checkbox"/> imposer l'humidité du gaz		valeurs spécifiques (rapportées à 1 kg de gaz sec)		
w (kg/kg)	0,00639642899	q' (kJ/kg)	34,2469	
epsi	0,500000034	v (m3/kg)	0,8334655	
condensats	0	t' (°C)	12,132	
p (bar)	1,01325	tr (°C)	7,431	
T (°C)	18			
T (K)	291,15			

Figure 4 : Ecran du point d'entrée

projet observée

point

corps

mélange externe

valeurs spécifiques (rapportées à 1 kg de gaz sec)

w (kg/kg)	<input type="text" value="0,0145345452"/>	q' (kJ/kg)	<input type="text" value="56,9361"/>
epsi	<input type="text" value="0,9899999999"/>	v (m3/kg)	<input type="text" value="0,8500568"/>
condensats	<input type="text" value="0"/>	t' (°C)	<input type="text" value="19,8885"/>
p (bar)	<input type="text" value="1,01325"/>	tr (°C)	<input type="text" value="19,838"/>
T (°C)	<input type="text" value="20"/>		
T (K)	<input type="text" value="293,15"/>		

Figure 5 : Ecran du point de sortie

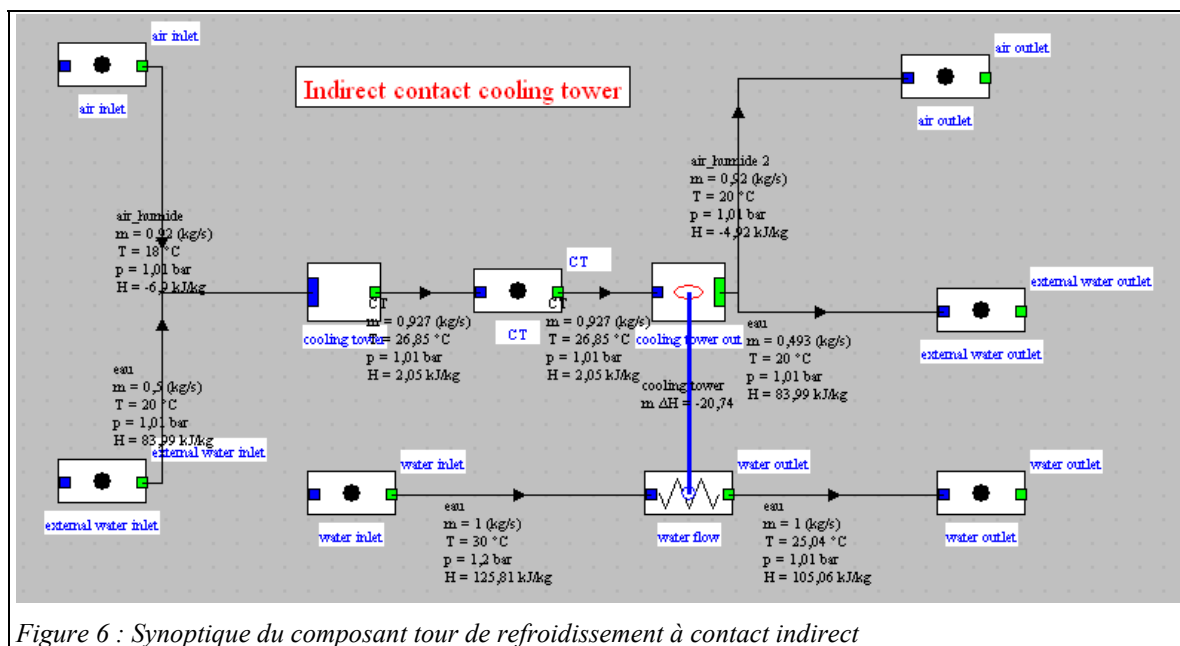


Figure 6 : Synoptique du composant tour de refroidissement à contact indirect

Etude de la classe externe InDirectCoolingTower

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, de l'air 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 InDirectCoolingTower permet de voir comment le modèle a été implémenté. Comme on peut le voir, dans le cas où la température de l'air en sortie est connue, six étapes suffisent pour effectuer les calculs (dans les autres cas, la démarche est analogue) :

- 1) on commence par calculer les propriétés humides de l'air entrant, grâce à la méthode générique `updatepoint()`, puis on initialise le débit d'air sec et l'enthalpie spécifique amont :

```
//imposition de w et calcul des propriétés humides
updatepoint(amont, false, 0, //T
            false, 0, false, 0, //P,x
            true, "setW and calculate all", old_w);

getPointProperties(amont);

Double f=(Double)vProp.elementAt(3);
double flow=f.doubleValue();
double flow_as=flow/(1+Wpoint); //débit massique de gaz sec

inletEpsi_value.setText(Util.aff_d(Epsipoint,3));
double hamont=QPrimepoint; //enthalpie spécifique de l'air entrant
```

- 2) on calcule ensuite les propriétés de l'air humide sortant et on en déduit l'enthalpie totale mise en jeu

```
//Propriétés humides de l'air sortant
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=airProcess;//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 de l'air sortant

double DeltaQprime=flow_as*(haval-hamont); //enthalpie totale acquise par l'air
DeltaQprime_value.setText(Util.aff_d(DeltaQprime,3));
```

- 3) on modifie la composition de l'air humide en sortie, et on calcule les débits-masses sortants :

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

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

double waterFlow=(Wpoint-old_w)*flow_as;
water_value.setText(Util.aff_d(waterFlow,5));

approach_value.setText(Util.aff_d(waterT-outletT,3));
```

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

```
//on considère que l'échange se fait avec le débit d'eau moyen
updateThermoCoupler("cooling tower", Tamont, Tpoint, DeltaQprime, icti.waterFlow-waterFlow/2);
//mise à jour du noeud en utilisant les méthodes génériques
vTransfo= new Vector[nBranches+1];
vPoints= new Vector[nBranches+1];
setupVector(airProcess, airPoint, 0, flow, airT, airP, 0);
setupVector(waterProcess, waterPoint, 1, icti.waterFlow-waterFlow, waterT, airP, 0);
setupVector(mainProcess, aval, 2, outletFlow, outletT, airP, 0);
updateDivider(vTransfo,vPoints,outletT,haval);
```