

# Simulation et contrôle de la température d'un hall industriel

Ir. T. DELAITE  
Ing. C. PHILIPPE  
Dr Ir. V. HANUS  
PIERRARD – Virton  
ULB BEAMS Energy

*Ce projet vise à optimiser la régulation de la température d'un hall industriel en contrôlant, à la fois, la température de la soufflerie mais également son débit de ventilation. Le choix de ces deux paramètres, pour obtenir la température voulue dans le hall, sera effectué en utilisant une optimisation des coûts de fonctionnement afin d'obtenir un système robuste, capable de contrôler une large gamme de perturbations tout en réduisant au maximum les coûts de fonctionnement sur une année entière grâce au logiciel Matlab et à son extension Simulink.*

*Mots-clefs : simulation, contrôle de température, modélisation, optimisation, Matlab*

*The aim of this project is to optimise the temperature control of an industrial hall by controlling both the temperature of the blower and its ventilation rate. The design of these two parameters will be made using an optimization of the operating costs to obtain the desired temperature in the hall. The main goal is to obtain a robust system, capable of controlling a wide range of disturbances while reducing the operating costs as much as possible over a full year using the Matlab software and its Simulink extension.*

*Keywords : simulation, temperature control, modelization, optimization, Matlab*

## 1. Introduction

La température de très nombreux halls industriels est actuellement régulée via une ventilation pulsant de l'air chaud ou froid en fonction des besoins. Dans ceux-ci, la régulation s'effectue généralement en contrôlant la température de ventilation via un régulateur PID tandis que le débit est géré en « tout ou rien », en désactivant les différentes gaines de ventilation, ou en les activant à leur débit maximal.

Cet article propose une optimisation du contrôle en régulant, en même temps, les deux variables affectant la température interne via des contrôleurs PID (en étudiant préalablement l'influence et le contrôle de chacune de ces variables de manière indépendante). Puisqu'il existe deux variables d'entrée pour une seule de sortie, il sera nécessaire d'introduire des équations supplémentaires pour résoudre le système. Ce projet utilisera une optimisation des coûts de fonctionnement de la soufflerie afin de trouver une solution permettant de garantir la température désirée dans le hall tout en minimisant les coûts de l'installation.

L'ensemble de ce projet a été réalisé sous le logiciel *Matlab/Simulink* avec les extensions *Control and Design Toolbox*, *Simulink Control Design* et *Guide*. Celles-ci permettent de réaliser un logiciel automatisé, couplé à des interfaces graphiques pour faciliter son utilisation par d'autres utilisateurs dans des configurations diverses.

## 2. Modélisation thermique

Le système thermique de l'installation est donné par son équation différentielle [1] :

$$\rho c_p V \frac{dT_{in}}{dt} = \rho c_p \dot{V} (T_{su} - T_{in}) - AK(T_{in} - T_{out}) - \rho c_p \dot{V}_{inf} (T_{in} - T_{out}) + \dot{Q} \quad (2.1)$$

Avec

$\rho$	Masse volumique de l'air	$[kg/m^3]$
$A$	Surface d'échange entre le bâtiment et l'extérieur	$[m^2]$
$c_p$	Capacité thermique massique	$[J/kg^\circ C]$
$K$	Coefficient de transfert thermique surfacique	$[W/m^2^\circ C]$
$\dot{Q}$	Flux thermique interne	$[W]$
$T_{in}$	Température interne du bâtiment	$[^\circ C]$
$T_{out}$	Température extérieure au bâtiment	$[^\circ C]$
$T_{su}$	Température de la soufflerie	$[^\circ C]$
$V$	Volume du bâtiment	$[m^3]$
$\dot{V}$	Débit total de la soufflerie	$[m^3/s]$
$\dot{V}_{inf}$	Débit d'infiltration	$[m^3/s]$

### 3. Contrôle modèle SISO $T_{su}$

#### 3.1. Modèle mathématique

Afin de réaliser le contrôle simultané de la température de ventilation  $T_{su}$  et du débit  $\dot{V}$ , il a été choisi de procéder par étapes successives afin de progressivement complexifier l'étude du système. La première étape consiste donc à uniquement contrôler la température de pulsation  $T_{su}$ . Pour ce faire, il suffit de démarrer de l'équation (2.1) en isolant  $T_{in}$  comme variable d'état et  $T_{su}$  comme variable de commande :

$$\frac{dT_{in}}{dt} = -\left(\frac{\dot{V}}{V} + \frac{AK}{\rho c_p V} + \frac{\dot{V}_{inf}}{V}\right)T_{in} + \frac{\dot{V}}{V}T_{su} + \left(\frac{AK}{\rho c_p V} + \frac{\dot{V}_{inf}}{V}\right)T_{out} + \frac{\dot{Q}}{\rho c_p V} \quad (3.1)$$

Cette équation (3.1) est totalement linéaire et peut également se mettre sous la forme d'un modèle à retour d'état simple  $\dot{x} = Ax + Bu + E$ . Ceci explique pourquoi la plupart des entreprises utilisent actuellement cette méthode. En effet, en l'associant à un débit de ventilation  $\dot{V}$  en « tout ou rien », il est théoriquement aisé de réguler la température interne des halls de production industriels.

#### 3.2. Contrôleur PI

Le modèle mathématique étant un système du premier ordre, il est dès lors réglable via un simple contrôleur PI de la forme  $C(s) = K_p + K_i \frac{1}{s}$ . Afin de paramétrer ce dernier, un ensemble de tests a été réalisé pour différentes valeurs de  $T_{out}$  et  $\dot{V}$ <sup>1</sup>.

	$T_{out,m}$	$\dot{V}_m$	$T_{out,M}$	$\dot{V}_M$	$T_{out,m}$	$\dot{V}_M$	$T_{out,M}$	$\dot{V}_M$	$T_{out,moy}$	$\dot{V}_{moy}$
$K_p$	6,2138		6,2138		4,332		4,332		4,6042	
$K_i$	0,0023		0,0023		0,0029		0,0029		0,0025	

Ces tests montrent directement que la valeur de la température extérieure n'intervient pas dans le paramétrage du contrôleur. Il suffira dès lors de le paramétrer en fonction du débit de ventilation  $\dot{V}$ .

Les paramètres choisis pour le paramétrage de ce contrôleur sont les suivants :

- Une bande passante 2 fois plus importante que celle du système initial pour diminuer le temps de réponse (les variations de température de ce système sont assez lentes pour permettre un meilleur confort aux occupants) ;
- Une marge de phase maximale de 90° afin de garantir un système le plus stable possible en boucle fermée.

<sup>1</sup>  $m = \text{minimum}$ ,  $M = \text{maximum}$  et  $moy = \text{moyenne arithmétique}$

A partir de ces résultats, un ensemble de contrôleurs a été automatiquement paramétré grâce au module *Control and Design Toolbox* de *Matlab* pour toutes les valeurs possibles de  $\dot{V}$  variant de sa valeur minimale à maximale. Ces résultats suivant, dans l'ensemble, une évolution plutôt linéaire, il a été décidé de réaliser une interpolation du premier ordre afin de pouvoir établir une loi aisément implémentable. Les résultats de ces simulations sont illustrés à la figure 3.1.

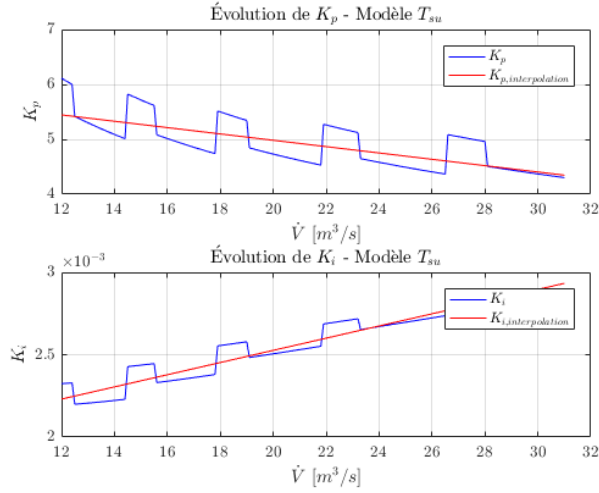


Figure 3.1 : Évolutions des paramètres du PI du modèle  $T_{su}$

Les lois liant les paramètres du contrôleur au débit de ventilation donnent :

$$\begin{cases} K_p = -0,0577\dot{V} + 6,1372 \\ K_i = 0,000037\dot{V} + 0,0018 \end{cases} \quad (3.2)$$

On observe directement qu'il est même possible d'approximer ces lois par des constantes afin d'obtenir un système nécessitant moins de calculs au sein du contrôleur en échange d'une assez faible perte de précision.

Par ailleurs, une boucle anti-windup a été ajoutée au contrôleur afin d'éviter tout emballement du terme intégral [2] :

$$K_s = \frac{1}{\sqrt{K_i}} = \frac{1}{\sqrt{|0,000037\dot{V} + 0,0018|}} \quad (3.3)$$

### 3.3. Simulations et résultats SISO $T_{su}$

Afin d'analyser les résultats des simulations contrôlant  $T_{su}$ , il a été nécessaire d'analyser les variations de la température extérieure  $T_{out}$  et du débit de ventilation  $\dot{V}$ . Ces

simulations montreront uniquement les résultats dans les cas extrêmes, aussi bien pour  $T_{out}$  que pour  $\dot{V}$ .

### ***Température extérieure variable***

Pour ce faire, une première simulation, présentée aux figures 3.2 et 3.3, montre l'évolution de  $T_{in}$  et de  $T_{su}$  pour les valeurs extrêmes de  $T_{out}$  (récupérées parmi des mesures au sein de l'entreprise durant une année complète) tout en faisant varier  $\dot{V}$  de sa valeur minimale à maximale. Celles-ci permettent donc de voir comment le contrôleur réagit aux variations de température externe  $T_{out}$ .

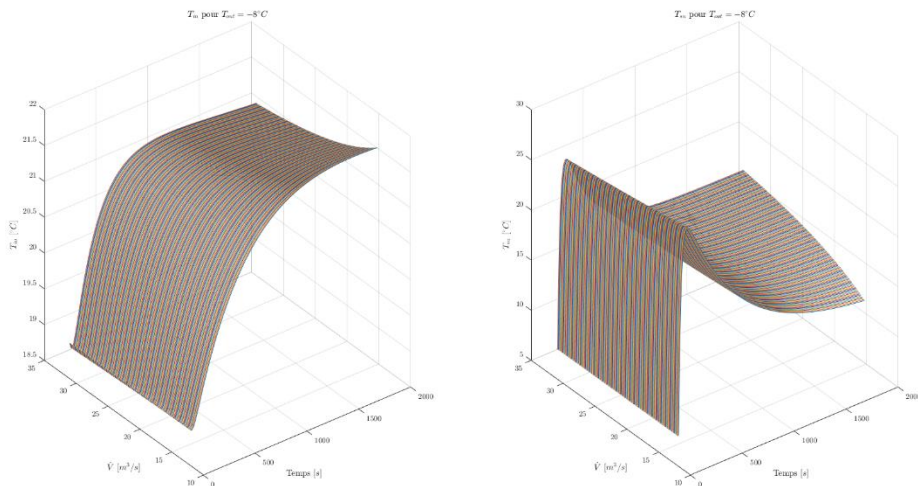


Figure 3.2 : Evolutions de  $T_{in}$  et  $T_{su}$  à  $T_{out,min}$  du modèle  $T_{su}$

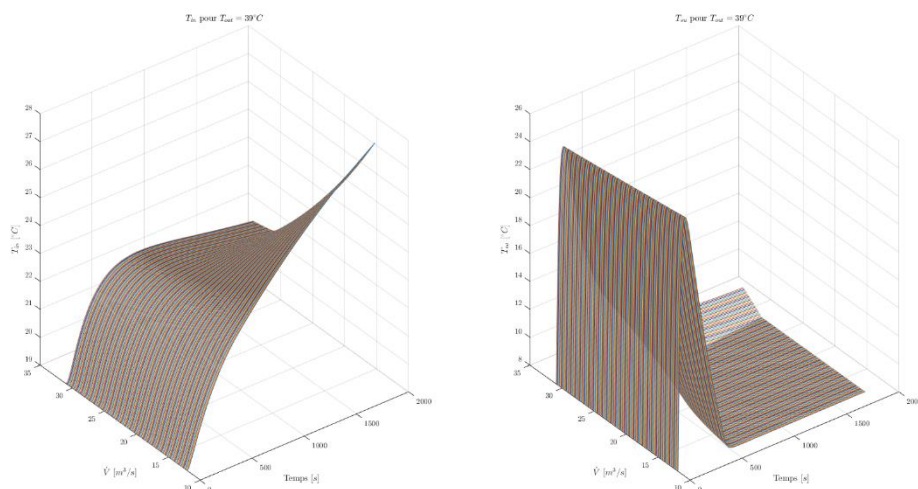


Figure 3.3 : Evolutions de  $T_{in}$  et  $T_{su}$  à  $T_{out,max}$  du modèle  $T_{su}$

Sur le graphique 3.2, il est possible d’observer qu’à basse température extérieure ( $-8^{\circ}\text{C}$ ), ce système simple arrive aisément à contrôler la température interne du hall aux  $21^{\circ}\text{C}$  désirés. La température de ventilation diminue simplement en même temps que  $\dot{V}$  puisqu’il faut toujours autant refroidir la pièce. Il est également important de noter que la température interne  $T_{in}$  diminue au démarrage avant de remonter. En effet, la simulation suppose un air pulsé à la température initiale du bâtiment ( $19^{\circ}\text{C}$ ) et il est donc nécessaire d’attendre un certain temps avant que celui-ci ne soit réchauffé et que l’énergie quitte alors le hall de production (dépendant de la constante de temps du système de chauffe  $\tau_{T_{su}}$ , très courte au vu de la durée de simulation).

A contrario, à haute température extérieure ( $39^{\circ}\text{C}$ ), le système n’est presque jamais capable de réguler. En dessous d’un débit de  $28,1 \text{ m}^3/\text{s}$ , la température de soufflerie sature déjà et le système n’est plus capable de refroidir le bâtiment. Cette limite est due au système qui fixe le débit de ventilation  $\dot{V}$ . Il arrive un moment où le débit ne permet plus d’amener suffisamment de calories froides. Le chapitre 5 permettra de solutionner ce problème en contrôlant à la fois les deux variables de commandes.

### ***Débit de ventilation variable***

Une seconde simulation a ensuite été réalisée. Celle-ci présente les évolutions de  $T_{in}$  et de  $T_{su}$  pour les valeurs extrêmes de  $\dot{V}$  tout en faisant varier  $T_{out}$  de sa valeur minimale à maximale comme illustré aux figures 3.4 et 3.5. De cette manière, les quatre cas extrêmes  $T_{out,min}$ ,  $T_{out,max}$ ,  $\dot{V}_{min}$  et  $\dot{V}_{max}$  ont pu être analysés. Si ces quatre possibilités sont fonctionnelles, toute situation entre les minimum et maximum le sera aussi.

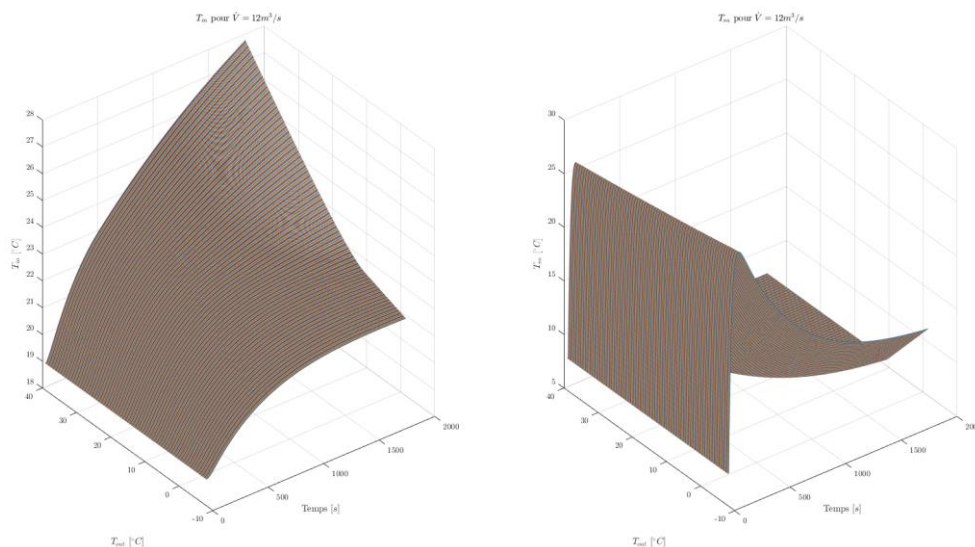


Figure 3.4 : Evolutions de  $T_{in}$  et  $T_{su}$  à  $\dot{V}_{min}$  du modèle  $T_{su}$

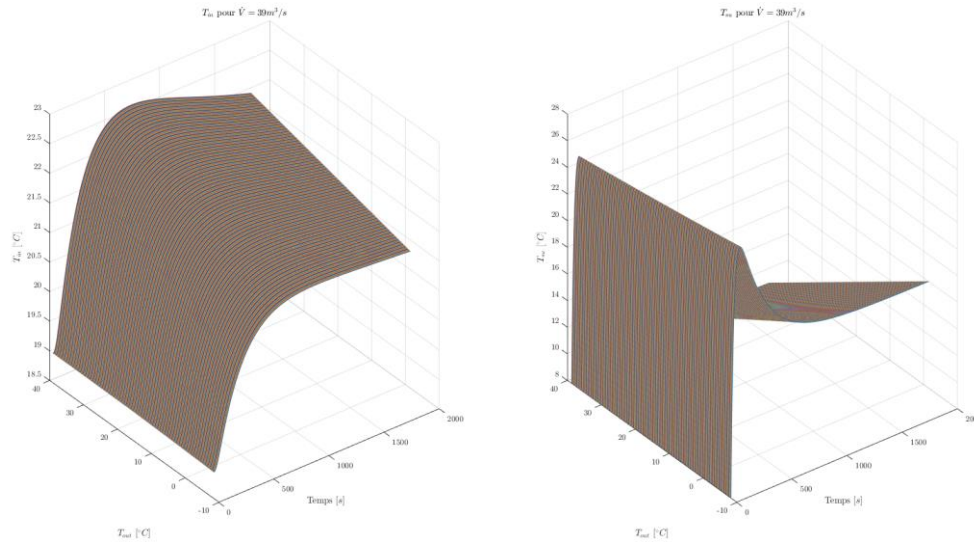


Figure 3.5 : Evolutions de  $T_{in}$  et  $T_{su}$  à  $\dot{V}_{max}$  du modèle  $T_{su}$

A faible débit de ventilation ( $12m^3/s$ ), le système est toujours incapable d'amener la température interne aux  $21^\circ C$  désirés. Puisque le débit est très faible, l'apport de calories froides l'est également et il est dès lors impossible de contrer les apports du flux interne  $\dot{Q}$  et du débit d'infiltration  $\dot{V}_{inf}$  vers l'ambiance extérieure. La température de ventilation  $T_{su}$  diminue au fur et à mesure que la température extérieure augmente mais finit par saturer à partir d'une température extérieure de  $4,6^\circ C$  à partir de laquelle le système n'est plus capable de correctement réguler la température interne  $T_{in}$  puisque le débit n'est plus suffisant pour contrer les pertes vers l'extérieur

En échange, à haut débit ( $30m^3/s$ ), le système est presque toujours capable de réguler la température interne à la consigne désirée (à condition de laisser un temps de simulation suffisamment important). De manière pratique, la température de soufflerie  $T_{su}$  augmente simplement pour contrer les pertes de chaleur vers l'ambiance extérieure lorsque  $T_{out}$  diminue.

### 3.4. Conclusion modèle SISO $T_{su}$

Le modèle contrôlant la température de ventilation  $T_{su}$ , tout en imposant le débit de soufflerie donne dans l'ensemble de très bons résultats. Cela explique pourquoi la plupart des entreprises actuelles se dirigent essentiellement vers cette méthode de contrôle pour son efficacité remarquable tout en étant extrêmement simple à mettre en œuvre. Il suffit de contrôler la température de soufflerie et d'augmenter ou diminuer le nombre de gaines actives en fonction des besoins afin de garantir une température interne sans grande variation. La seule difficulté apparaît lors de l'allumage complet d'une gaine ce qui provoque une grande variation de  $\dot{V}$  et donc de  $T_{su}$ .

## 4. Contrôle modèle SISO $\dot{V}$

### 4.1. Modèle mathématique

Dans un second temps, ce projet s'attachera à réaliser le contrôle de la température interne  $T_{in}$  en régulant uniquement le débit de la ventilation  $\dot{V}$ . Pour ce faire, il suffit de redémarrer de l'équation (2.1) en isolant  $\dot{V}$  comme variable de commande :

$$\frac{dT_{in}}{dt} = -\left(\frac{\dot{V}}{V} + \frac{AK}{\rho c_p V} + \frac{\dot{V}_{inf}}{V}\right)T_{in} + \frac{\dot{V}}{V}T_{su} + \left(\frac{AK}{\rho c_p V} + \frac{\dot{V}_{inf}}{V}\right)T_{out} + \frac{\dot{Q}}{\rho c_p V} \quad (4.1)$$

Malheureusement, la variable de commande  $\dot{V}$  de (4.1) agit directement sur la variable d'état  $T_{in}$  via un produit. Le système est non **non-linéaire**, ce qui va fortement compliquer sa résolution puisqu'il ne sera plus possible d'utiliser les algorithmes et ressources classiques de *Matlab/Simulink* [2]. Pour résoudre ce problème, il a été choisi d'utiliser une linéarisation du modèle autour d'un point de fonctionnement.

### 4.2. Contrôleur PI

A nouveau, les simulations de ce système du premier ordre ont orienté le régulateur vers un simple contrôleur PI. Pour paramétrer ce dernier, un ensemble de tests a été réalisé pour différentes valeurs de  $T_{out}$  et  $T_{su}$ <sup>2</sup>.

	$T_{out,m}$	$T_{su,m}$	$T_{out,M}$	$T_{su,m}$	$T_{out,m}$	$T_{su,M}$	$T_{out,M}$	$T_{su,M}$	$T_{out,moy}$	$T_{su,moy}$
$K_p$	-5,6266		-5,6266		5,6266		5,6266		0	
$K_i$	-0,0026		-0,0026		0,0026		0,0026		-1	

A nouveau, la température extérieure  $T_{out}$  n'influe aucunement sur les valeurs du régulateur, ce qui simplifiera grandement le paramétrage de ce contrôleur PI puisqu'il suffira de le régler en fonction de la température de soufflerie  $T_{su}$ . De plus, il existe une symétrie évidente autour des valeurs moyennes. **Néanmoins, d'autres simulations ont montré que cette symétrie n'est pas centrée sur la moyenne arithmétique  $T_{su,moy}$  mais sur la température initiale du système  $T_{in,initial}$ .** Ces deux températures étant simplement confondues dans les différentes simulations illustrées dans cet article.

Puisque l'évolution, en se basant uniquement sur ces cinq points, semble linéaire, diverses autres simulations ont été réalisées afin de pouvoir vérifier que la forme réelle de cette évolution est bien une droite. Il est à noter que ces simulations sont effectuées sur le modèle mathématique **linéarisé**. Cette linéarisation introduit des

<sup>2</sup>  $m = minimum$ ,  $M = maximum$  et  $moy = moyenne arithmétique$



imperfections dans le paramétrage du contrôleur. Néanmoins, aucune méthode actuelle ne permet d'obtenir un contrôleur optimal pour des systèmes non-linéaires tel que celui de ce projet. Certaines techniques approchées existent néanmoins [3].

A nouveau, les contrôleurs PI du modèle linéarisé ont été paramétrés grâce au module *Control and Design Toolbox* de *Matlab* pour toutes les valeurs possibles de  $T_{su}$  variant de sa valeur minimale à maximale (résultats illustrés à la figure 4.1) avec les mêmes conditions que précédemment :

- Une bande passante 2 fois plus importante que celle du système initial pour accélérer le système ;
- Une marge de phase maximale de  $90^\circ$  pour garantir une stabilité maximale au système en boucle fermée.

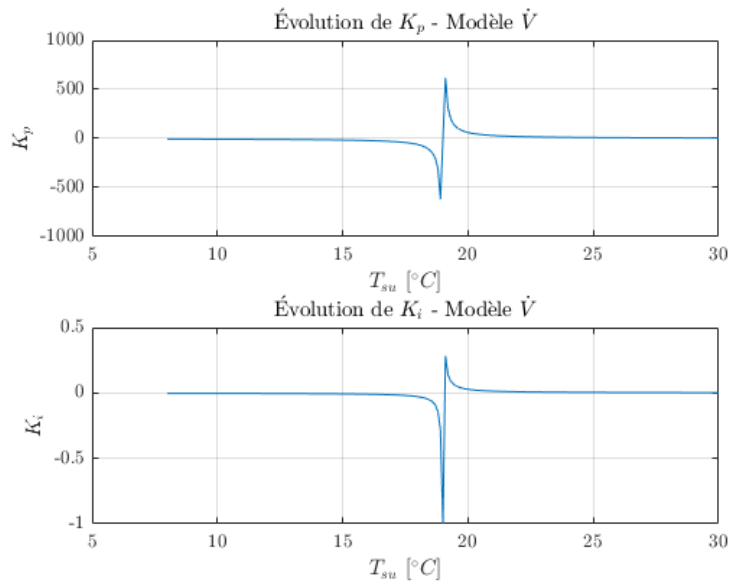


Figure 4.1 : Évolutions des paramètres du PI du modèle  $\dot{V}$

En analysant la figure 4.1, il apparaît un passage de valeurs négatives à positives autour de la température initiale du bâtiment. Ce changement de signe correspond en réalité au moment où l'air dans la ventilation est trop chaud pour pouvoir refroidir le hall (cette valeur est dictée par la température initiale du hall de production). Le régulateur PI veut donc changer le sens de la ventilation afin de continuer à refroidir puisque celui-ci n'a aucune idée de la réalité physique du système, il ne s'agit que de nombres dont la valeur négative inverse le processus. Il a dès lors été décidé d'approximer ces paramètres suivant des lois de la forme :

$$K_{p,i} = \frac{C_{p,i}}{T_{su} - T_{in,initiale}} \quad (4.2)$$

Malheureusement, la présence d'un saturateur dans le modèle pose un problème au-delà de la température initiale du système. En effet, le contrôleur veut inverser le sens de la soufflerie et le rendre négatif (les termes proportionnel et intégral sont dès lors positifs). Le saturateur limite ce débit minimum à  $\dot{V}_{min}$  (constant et positif) alors qu'il faut justement souffler plus fort pour tenter de refroidir.

Pour pallier ce problème, il a été décidé de transformer les équations du contrôleur PI afin de les rendre toujours négatives :

$$\begin{cases} K_p = - \left| \frac{61,8922}{T_{su} - T_{in,initiale}} \right| \\ K_i = - \left| \frac{0,0283}{T_{su} - T_{in,initiale}} \right| \end{cases} \quad (4.3)$$

Pour finir, une boucle anti-windup a de nouveau été ajoutée afin d'éviter l'emballement du terme intégral [2] :

$$K_s = \frac{1}{\sqrt{K_i}} = \frac{1}{\sqrt{\left| \frac{0,0283}{T_{su} - T_{in,initiale}} \right|}} \quad (4.4)$$

Les résultats finaux de ces interpolations sont visibles à la figure 4.2 où l'on peut observer la précision des interpolations finales utilisant le modèle d'équation (4.2).

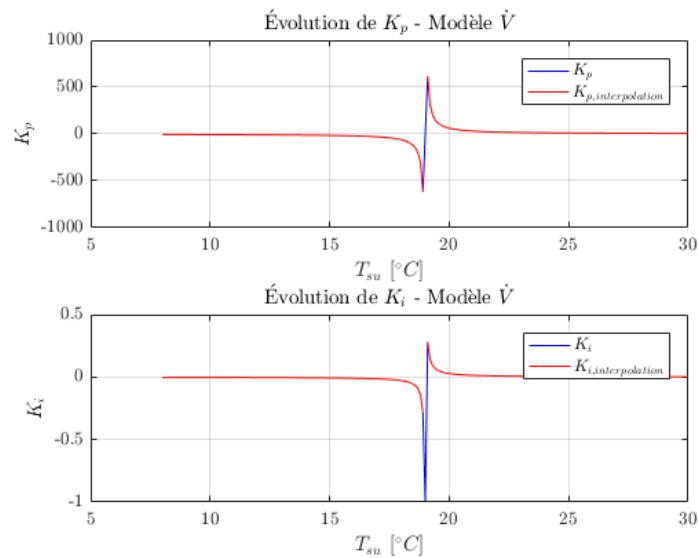


Figure 4.2 : Interpolations des paramètres du PI du modèle  $\dot{V}$

Ces interpolations seront dès lors utilisées dans le contrôleur afin de permettre une évaluation mathématique simple et efficace des paramètres  $K_p$  et  $K_i$  du régulateur PI. De plus, ce modèle de contrôleur permet aussi bien de réchauffer que de refroidir la pièce, en fonction des températures extérieures (été ou hiver) et du flux thermique interne dépendant du nombre et de la puissance thermique dissipée par les différentes machines en fonctionnement. Il est donc valable dans la plupart des cas réels.

### 4.3. Simulations et résultats SISO $\dot{V}$

Afin d'analyser les résultats des simulations contrôlant  $\dot{V}$ , il a été nécessaire d'analyser toutes les variations possibles de la température extérieure  $T_{out}$  et de la température de soufflerie  $T_{su}$ . Ces simulations montreront uniquement les résultats dans les cas extrêmes, aussi bien pour  $T_{out}$  que pour  $T_{su}$ . Tout comme précédemment, si les quatre graphiques présentés dans cette section sont fonctionnels, toute utilisation d'une valeur intermédiaire le sera également (en dehors de la singularité mathématique apparaissant lorsque  $T_{su} = T_{in,initiale}$ ).

#### *Température extérieure variable*

La première simulation (présentée aux figure 4.3 et 4.4) montre l'évolution de la température interne  $T_{in}$  et du débit  $\dot{V}$  pour les valeurs extrêmes de  $T_{out}$  tout en faisant varier la température de soufflerie entre ses valeurs extrêmes. Ces simulations montrent ainsi comment le contrôleur PI est capable de réguler la température interne en fonction de l'ambiance extérieure en régulant uniquement le débit  $\dot{V}$  alors que la température de soufflerie  $T_{su}$  est imposée<sup>3</sup>.

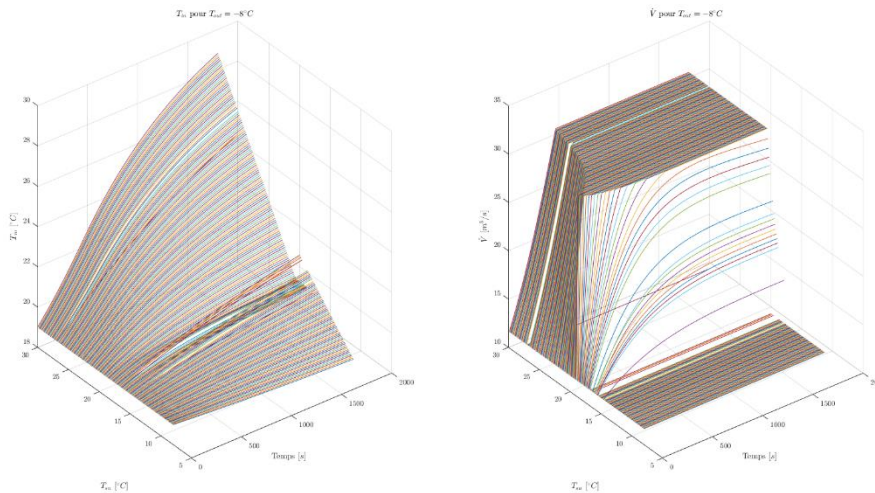


Figure 4.3 : Evolutions de  $T_{in}$  et  $\dot{V}$  à  $T_{out,min}$  du modèle  $\dot{V}$

<sup>3</sup> Le débit de ventilation  $\dot{V}$  est inversement proportionnel à la température interne  $T_{in}$ .

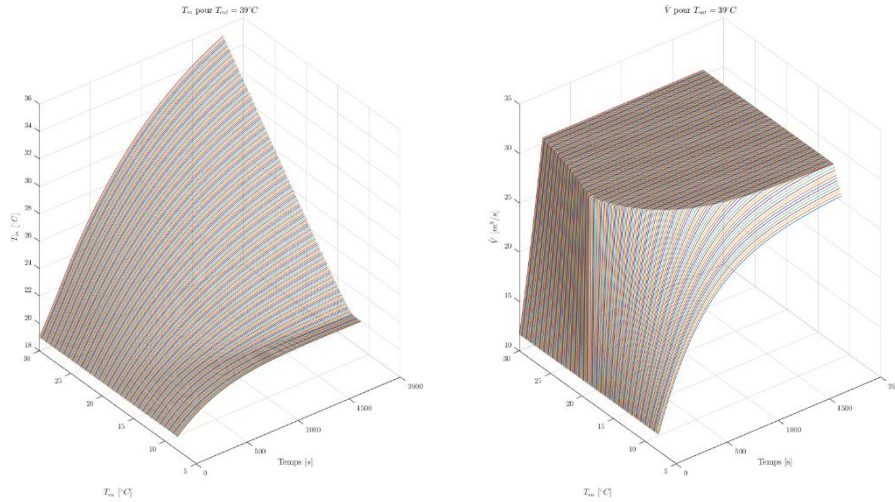


Figure 4.4 : Evolutions de  $T_{in}$  et  $\dot{V}$  à  $T_{out,max}$  du modèle  $\dot{V}$

Ces graphiques montrent directement qu'à basse température extérieure ( $-8^{\circ}\text{C}$ ), le système est d'abord incapable de réguler la température interne puisque la température de soufflerie n'est pas assez élevée. Par la suite, le contrôleur devient capable d'assurer sa fonction et le débit de ventilation  $\dot{V}$  augmente en même temps que  $T_{su}$  puisqu'il faut toujours autant refroidir le hall. Malheureusement, cette commande réalisant bien le contrôle voulu n'est efficace que durant une très courte plage de température de soufflerie (de  $14,6^{\circ}\text{C}$  à  $17,2^{\circ}\text{C}$ ). Une fois cette plage dépassée, il y a saturation du débit de ventilation  $\dot{V}$  et la température interne  $T_{in}$  augmente très fortement puisque le système pulse de l'air chaud dans une pièce qui a besoin d'être refroidie, ce qui va à l'encontre du but recherché.

A haute température extérieure ( $39^{\circ}\text{C}$ ), le système n'est quasiment jamais capable de réguler. A partir d'une température de soufflerie de  $9,3^{\circ}\text{C}$ , le débit  $\dot{V}$  sature déjà et le système n'est pas capable de remplir sa fonction. Lorsque l'ambiance extérieure est particulièrement chaude, ce système ne peut réguler efficacement la température interne que pour une température de soufflerie de  $8$  à  $9,3^{\circ}\text{C}$ . Une solution sera apportée lors du contrôle simultané des différentes variables de commande où il sera possible de baisser la température de soufflerie tout en augmentant son débit.

### **Température de soufflerie variable**

La seconde simulation de cette section présente l'évolution de la température interne  $T_{in}$  et du débit  $\dot{V}$  pour les valeurs extrêmes de  $T_{su}$  en faisant varier  $T_{tout}$  entre ses valeurs minimale et maximale relevées sur le site de production. Ces résultats présenteront d'autres difficultés que celles présentées ci-avant à cause du contrôle d'une seule variable non-linéaire. Les résultats de ces simulations sont présentés aux figures 4.5 et 4.6.

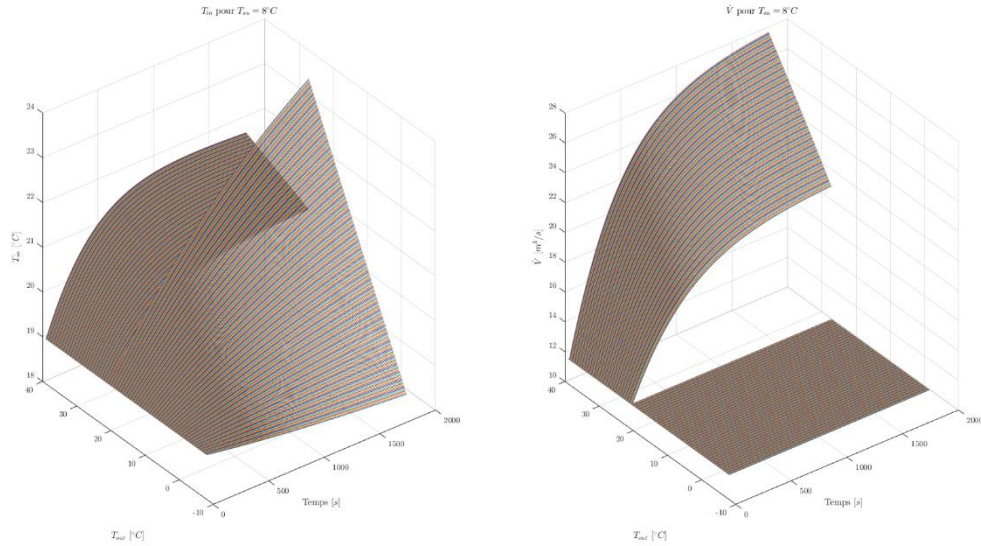


Figure 4.5 : Evolutions de  $T_{in}$  et  $\dot{V}$  à  $T_{su,min}$  du modèle  $\dot{V}$

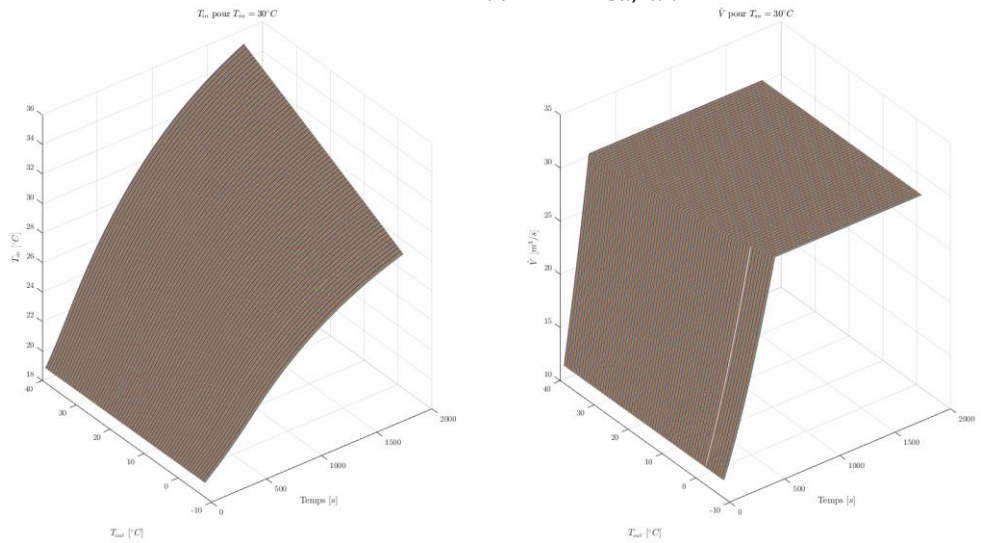


Figure 4.6 : Evolutions de  $T_{in}$  et  $\dot{V}$  à  $T_{su,max}$  du modèle  $\dot{V}$

A faible température de ventilation ( $8^{\circ}\text{C}$ ), le système est parfois capable d'amener la température finale à la consigne de  $21^{\circ}\text{C}$  désirée. Le débit de ventilation  $\dot{V}$  augmente simplement pour contrer l'augmentation de la température extérieure. Néanmoins, en dessous d'une ambiance extérieure de  $20,7^{\circ}\text{C}$ , le système n'est plus capable de réguler puisque la température de soufflerie est fixée à sa valeur minimale de  $8^{\circ}\text{C}$ . Ceci signifie que même un débit minimum fournit trop d'énergie froide. Pour contrer ce problème, il faudrait augmenter la température de soufflerie  $T_{su}$  comme cela sera développé dans le chapitre 5.

Pour finir, à haute température de soufflerie, ( $30^{\circ}\text{C}$ ), le système n'arrive simplement jamais à réguler. Le débit  $\dot{V}$  est directement saturé à son maximum et la température finale est très largement supérieure à la consigne désirée (le maximum atteint étant même de  $35,27^{\circ}\text{C}$ ). Cette simulation est en réalité peu utile puisqu'elle suppose souffler de l'air à  $30^{\circ}\text{C}$  dans un hall devant être refroidi à  $21^{\circ}\text{C}$ . Son seul intérêt réside dans des températures extérieure très froides où il est nécessaire de réchauffer le hall.

#### 4.4. Conclusion modèle SISO $\dot{V}$

Ce modèle contrôlant le débit de ventilation  $\dot{V}$  tout en imposant la température de soufflerie donne dans l'ensemble des résultats assez médiocres. Cela s'explique par la non-linéarité du système original ainsi que de l'influence bien plus importante de la température de soufflerie plutôt que du débit. Cette méthode de contrôle n'est donc absolument pas adaptée à un hall de production et la méthode classique consistant à uniquement contrôler la température de soufflerie  $T_{su}$  est bien plus efficace.

### 5. Contrôle modèle MISO $T_{su}$ et $\dot{V}$

#### 5.1. Modèle mathématique

Les chapitres 3 et 4 réalisaient le contrôle d'une des deux variables de commande disponible  $T_{su}$  ou  $\dot{V}$  pour réguler la température interne du hall. Malheureusement, les résultats ont montré que, bien que parfois efficaces, ces méthodes étaient loin d'être optimales.

Le but de cet ultime chapitre est donc de combiner les deux systèmes précédents afin d'établir un contrôleur capable de gérer **simultanément** ces deux variables. Puisqu'il existe deux variables de commande pour une seule de sortie, il est nécessaire d'ajouter une équation supplémentaire pour pouvoir résoudre le système d'équations. Cette article utilisera une optimisation des coûts de fonctionnement afin d'obtenir un système capable de contrôler la température interne tout en diminuant au maximum les coûts d'utilisation du système de ventilation de l'entreprise.

#### 5.2. Optimisation des variables de commande $T_{su}$ et $\dot{V}$

Pour optimiser les variables de commandes  $T_{su}$  et  $\dot{V}$ , il est tout d'abord nécessaire de modéliser l'évolution de leur coût. Il est à noter que, dans la pratique, l'évolution du coût de  $\dot{V}$  est proportionnelle au débit (plus ce débit est haut, plus cela est onéreux) tandis que l'évolution du coût de  $T_{su}$  est inversement proportionnelle à cette température. Inversement, par rapport à la température interne, ces règles s'inversent puisqu'il faut souffler plus fort pour mieux refroidir tandis qu'il suffit d'augmenter

la température de soufflerie pour réchauffer le bâtiment. Ces particularités ont posé de nombreuses complexités lors de la réalisation du code *Matlab* optimisant le système puisqu'il a été nécessaire de changer le signe de très nombreuses variables tout au long des différents scripts.

L'évolution du coût d'une variable suit généralement une loi quadratique<sup>4</sup> comme montré aux équations (5.1) et à la figure 5.1 illustrant les résultats numériques :

$$\begin{cases} C_{T_{su}}(T_{su}) = \frac{a_{T_{su}}}{2} T_{su}^2 + b_{T_{su}} T_{su} + c_{T_{su}} \\ C_{\dot{V}}(\dot{V}) = \frac{a_{\dot{V}}}{2} \dot{V}^2 + b_{\dot{V}} \dot{V} + c_{\dot{V}} \end{cases} \quad (5.1)$$

L'évolution du coût de  $\dot{V}$  est donnée par des données expérimentales présentant l'évolution du coût horaire par pas de  $10m^3/h$ . Celles-ci sont issues du prix au *kWh* couplé à la loi de similitude liant le débit à la puissance électrique du ventilateur [5].

$$\frac{P_{nominale}}{P_{actuelle}} = \left( \frac{\dot{V}_{nominale}}{\dot{V}_{actuel}} \right)^3 \quad (5.2)$$

Pour  $T_{su}$ , il a été nécessaire de choisir des valeurs arbitraires. Celles-ci s'accordent avec l'expérience pratique où il revient moins cher de d'abord diminuer la température de soufflerie avant d'augmenter le débit, les différences devenant minimales lorsque ces commandes arrivent proches de leur coût maximum.

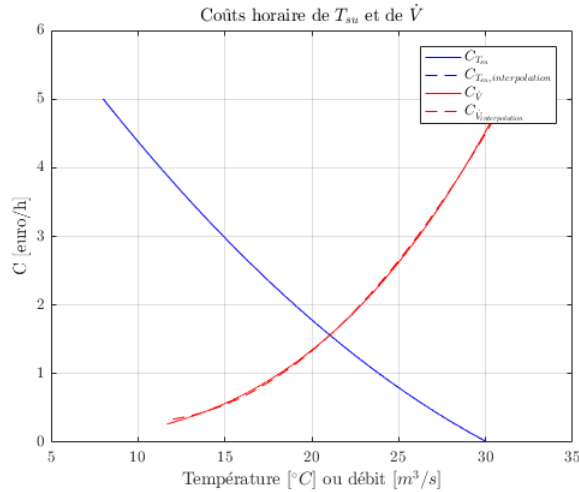


Figure 5.1 : Coûts horaires de  $T_{su}$  et  $\dot{V}$

<sup>4</sup> Quadratique : de la forme du second ordre  $\frac{a}{2}x^2 + bx + c$ .

Le but est dès lors d'optimiser le coût total de fonctionnement de la ventilation aussi bien au niveau de la température de soufflerie  $T_{su}$  que de son débit  $\dot{V}$  en prenant en compte les variations quadratiques de ces coûts avec une méthode similaire à [4] :

$$C_{total} = C_{T_{su}}(T_{su}) + C_{\dot{V}}(\dot{V}) = \frac{a_{T_{su}}}{2} T_{su}^2 + b_{T_{su}} T_{su} + c_{T_{su}} + \frac{a_{\dot{V}}}{2} \dot{V}^2 + b_{\dot{V}} \dot{V} + c_{\dot{V}} \quad (5.3)$$

Tout en garantissant la température finale dans le hall en régime  $\left(\frac{dT_{in}}{dt} = 0\right)$  :

$$T_{in} = \frac{\dot{V} T_{su} + \left(\frac{AK}{\rho c_p} + \dot{V}_{inf}\right) T_{out} + \frac{\dot{Q}}{\rho c_p}}{\dot{V} + \frac{AK}{\rho c_p} + \dot{V}_{inf}} \quad (5.4)$$

Il est dès lors possible d'obtenir la fonction Lagrangienne du système [6] :

$$\mathcal{L}(T_{in}, \lambda) = \sum_{i=1}^n C_i(x_i) - \lambda(T_{in} - T_{in,ref}) \quad (5.5)$$

Où  $n = 2$  :  $x_1 = T_{su}$  et  $x_2 = \dot{V}$

Les conditions optimales sont donc :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial C_i} = IC_i(x_i) - \lambda = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda} = -T_{in} + T_{in,ref} = 0 \end{cases} \quad (5.6)$$

Où  $IC_i$  est le coût incrémental [4] :

$$IC_i = \frac{dC_i(x_i)}{dx_i} \quad (5.7)$$

Pour finir, il faut adjoindre les limites opérationnelles du système. Celles-ci sont dictées par les limites physiques du système de ventilation :

$$\begin{cases} T_{su,min} \leq T_{su} \leq T_{su,max} \\ \dot{V}_{min} \leq \dot{V} \leq \dot{V}_{max} \end{cases} \quad (5.8)$$

Les systèmes d'équations (5.1), (5.4), (5.6) et (5.8) mènent finalement au système final constitué de trois équations à trois inconnues non-linéaires couplées à deux inéquations qu'il faut résoudre :



$$\left\{ \begin{array}{l}
\frac{\partial \mathcal{L}}{\partial T_{su}} = a_{T_{su}} T_{su} + b_{T_{su}} - \lambda \left[ \frac{\dot{V}}{\dot{V} + \frac{AK}{\rho c_p} + \dot{V}_{inf}} \right] = 0 \\
\frac{\partial \mathcal{L}}{\partial \dot{V}} = a_{\dot{V}} \dot{V} + b_{\dot{V}} - \lambda \left[ \frac{\left( \frac{AK}{\rho c_p} + \dot{V}_{inf} \right) (T_{su} - T_{out}) + \frac{\dot{Q}}{\rho c_p}}{\left( \dot{V} + \frac{AK}{\rho c_p} + \dot{V}_{inf} \right)^2} \right] = 0 \\
T_{in,ref} = \frac{\dot{V} T_{su} + \left( \frac{AK}{\rho c_p} + \dot{V}_{inf} \right) T_{out} + \frac{\dot{Q}}{\rho c_p}}{\dot{V} + \frac{AK}{\rho c_p} + \dot{V}_{inf}} \\
T_{su,min} \leq T_{su} \leq T_{su,max} \\
\dot{V}_{min} \leq \dot{V} \leq \dot{V}_{max}
\end{array} \right. \quad (5.9)$$

Plusieurs méthodes ont été tout d'abord testées afin de tenter de résoudre ce système d'équation (5.9) de la manière la plus parfaite d'un point de vue mathématique :

- *Une méthode purement analytique* : malheureusement, celle-ci donne des résultats en dehors des inéquations des limites physiques du système. Ceci signifie que la solution idéale économique n'est jamais atteignable à cause des limites physiques du système. Il est dès lors obligatoire de se contenter d'une solution approchée [7].
- *Une méthode linéarisée* : cette technique consiste à linéariser le système d'équations (5.9) en un système d'équations à retour d'état linéaire présentant la forme classique [3] :

$$\{\dot{T}_{in}\} = [a_{lin}]\{T_{in}\} + [b_{T_{su},lin} \quad b_{\dot{V},lin}] \begin{Bmatrix} T_{su} \\ \dot{V} \end{Bmatrix}$$

Malheureusement, cette méthode a de nouveau donné des résultats en dehors des limites physiques du système et n'est donc pas applicable à la résolution du système d'équations non-linéaires de ce projet.

Pour finir, il a été décidé d'utiliser un algorithme de dichotomie<sup>5</sup> pour résoudre le système (5.9). Celui-ci fonctionne comme suit [7] :

- Il estime tout d'abord  $\lambda$  via la dichotomie. Les valeurs initiales de  $\lambda_{min}$  et  $\lambda_{max}$  sont liées aux dérivées des coûts totaux extrêmes  $\frac{dC_{total,min}}{dt}$  et  $\frac{dC_{total,max}}{dt}$  ;

---

<sup>5</sup> De nombreux autres algorithmes permettent de trouver le zéro d'une fonction mathématique. La méthode dichotomique est sans aucun doute la plus simple à mettre en œuvre mais il est également possible d'appliquer des méthodes beaucoup plus efficaces comme les méthodes de Newton-Raphson ou de Newton-Gauss même si celles-ci sont difficiles à appliquer dans un système à deux variables non-linéaires [7] [8].

- Il calcule les nouvelles valeurs de  $T_{su}$  et de  $\dot{V}$  correspondant à ce  $\lambda$  en linéarisant les équations (5.9) (ou avec les non-linéarités pour des solveurs avancés<sup>6</sup>) ;
- Il obtient la valeur de  $T_{in}$  correspondant à ces valeurs de  $T_{su}$  et de  $\dot{V}$  ;
- Il met à jour  $\lambda_{min}$  ou  $\lambda_{max}$  en fonction du résultat. Celui-ci est ajusté en fonction d'une température  $T_{in}$  inférieure ou supérieure à la consigne ;
- Il continue les itérations jusqu'à ce que la tolérance désirée soit atteinte ou qu'un nombre d'itérations maximales soit dépassé<sup>7</sup>.

Malheureusement, les résultats obtenus via cette optimisation n'étaient pas réalistes à cause des non-linéarités du système couplées aux méthodes de résolution du logiciel. Il a dès lors été choisi de négliger certaines parties du système d'équations afin que Matlab puisse effectuer l'algorithme. Les résultats ne seront dès lors pas optimaux mais seront néanmoins bien meilleurs que les utilisations actuelles en « tout ou rien ». **De plus, cette résolution n'est plus « pure » mathématiquement.**

Le nouveau système d'équations qui est finalement simulé simplifie uniquement les parties liées aux liens entre  $\lambda$  et la dérivée du coût de chaque variable de commande :

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial T_{su}} = a_{T_{su}} T_{su} + b_{T_{su}} - \lambda = 0 \\ \frac{\partial \mathcal{L}}{\partial \dot{V}} = a_{\dot{V}} \dot{V} + b_{\dot{V}} - \lambda = 0 \\ T_{in,ref} = \frac{\dot{V} T_{su} + \left( \frac{AK}{\rho c_p} + \dot{V}_{inf} \right) T_{out} + \frac{\dot{Q}}{\rho c_p}}{\dot{V} + \frac{AK}{\rho c_p} + \dot{V}_{inf}} \\ T_{su,min} \leq T_{su} \leq T_{su,max} \\ \dot{V}_{min} \leq \dot{V} \leq \dot{V}_{max} \end{array} \right. \quad (5.10)$$

Comme il est aisé de l'observer dans le système (5.10), la dérivée du Lagrangien de l'équation (5.6) n'est plus parfaitement respectée. Néanmoins, les résultats de  $T_{su}$  et  $\dot{V}$  qui seront obtenus via cette optimisation imparfaite donneront bien la température  $T_{in}$  réelle dans le hall de production puisqu'elle est dictée par l'équation (5.4).

Ces analyses simplifiées mènent aux résultats des figures 5.2 et 5.3 illustrant l'évolution des paramètres  $T_{su}$  et  $\dot{V}$  optimaux en fonction de la température externe ainsi que l'évolution du coût horaire total  $C_{total}$  et de sa dérivée  $\lambda$  :

<sup>6</sup> Il serait possible d'obtenir de bien meilleurs résultats en utilisant des solveurs avancés tels que *Cplex Optimizer* [9] ou *Yalmip* [10] utilisables sous *Matlab*.

<sup>7</sup> Lors de la réalisation de codes présentant des boucles avec une sortie via une tolérance, il est obligatoire de prévoir une sortie supplémentaire après un nombre défini d'itérations.

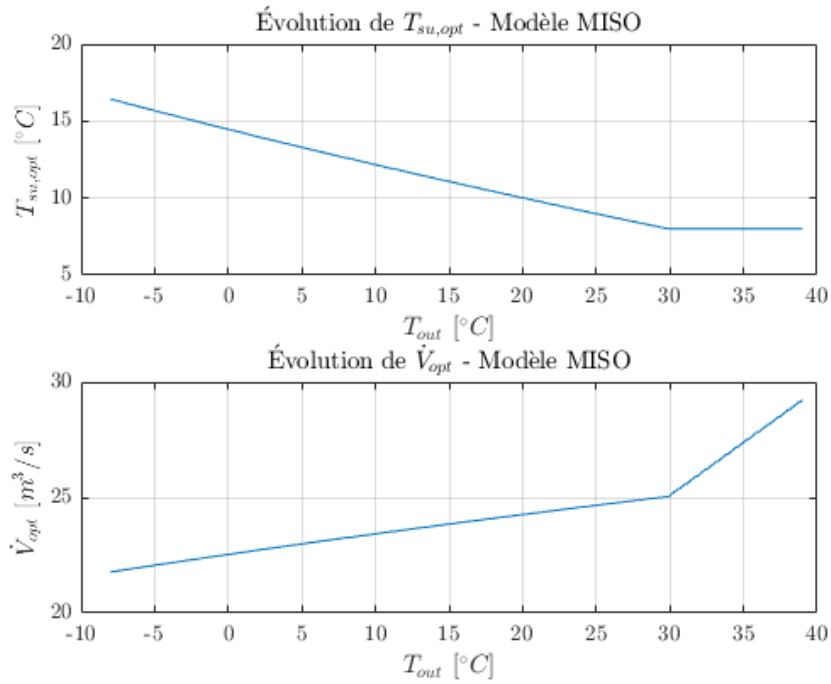


Figure 5.2 : Evolutions des paramètres  $T_{su}$  et  $\dot{V}$  optimaux

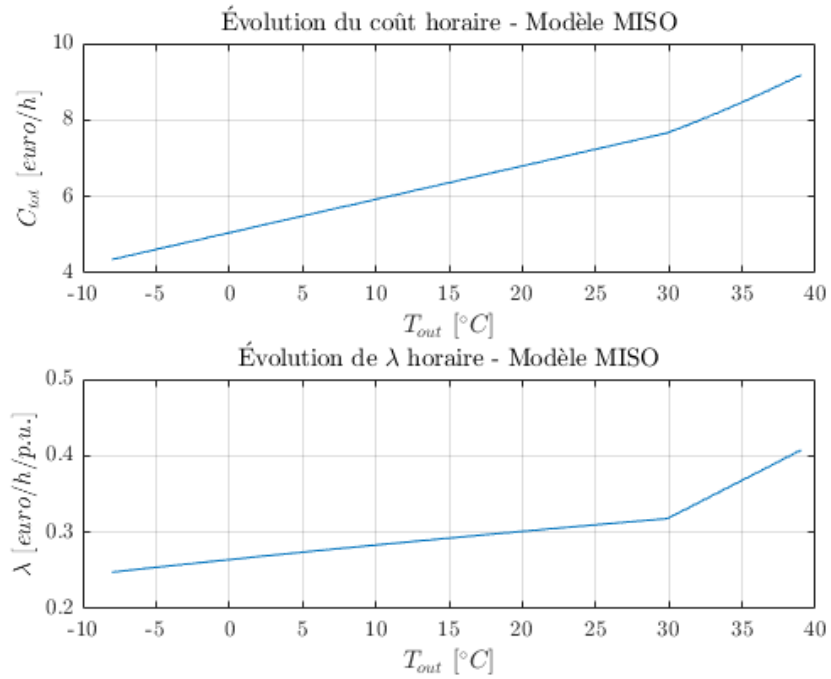


Figure 5.3 : Evolutions de  $C_{total}$  et de  $\lambda$

La figure 5.2 montre la présence d'une température extérieure limite  $T_{out,lim}$  de  $29,7^{\circ}C$  où la température de soufflerie  $T_{su}$  va saturer à la valeur minimale  $T_{su,min}$ . Ceci implique qu'à partir de cette température extérieure, le débit  $\dot{V}$  va devoir augmenter beaucoup plus rapidement pour contrer l'augmentation de température extérieure. Ce résultat est dicté par l'optimisation économique qui pousse, dans ce cas-ci, à plutôt diminuer la température  $T_{su}$  que d'augmenter le débit  $\dot{V}$  afin de garantir un coût total minimum.

De la même manière, la figure 5.3 illustre bien cette température limite où la saturation apparaît. A partir de cette limite, le coût varie différemment et augmente bien plus rapidement pour chaque degré de température extérieure supplémentaire puisque la seule manière d'agir sur la température interne  $T_{in}$  est d'augmenter le débit  $\dot{V}$  qui coûte de plus en plus cher au vu de son évolution quadratique.

### 5.3. Contrôleur PI

Le système simplifié (5.10) montre qu'il existe un lien entre  $T_{su}$ ,  $\dot{V}$  et  $\lambda$  tel que :

$$\begin{cases} T_{su} = \frac{\lambda - b_{T_{su}}}{a_{T_{su}}} \\ \dot{V} = \frac{\lambda - b_{\dot{V}}}{a_{\dot{V}}} \end{cases} \quad (5.11)$$

Il est dès lors possible de ne contrôler que  $\lambda$  via un unique contrôleur PI<sup>8</sup> et les équations (5.11) permettent d'obtenir les valeurs correspondantes de  $T_{su}$  et de  $\dot{V}$ . Le contrôleur de ce système est un peu différent des précédents puisque celui-ci s'attache uniquement à contrôler l'actionneur intermédiaire  $\lambda$ . Le système étant du premier ordre, un contrôleur PI est à nouveau largement suffisant.

Pour une dernière fois, de nombreux cas possibles ont été simulés en fonction de la température extérieure  $T_{out}$ <sup>9</sup>. Puisque le but de ce contrôle est de réguler à la fois  $T_{su}$  et  $\dot{V}$ , il ne reste que la température extérieure comme variable pouvant agir sur l'évolution du système :

	$T_{out,m}$	$T_{out,M}$	$T_{out,moy}$
$K_p$	-0,0325	-0,0325	-0,0325
$K_i$	-0,000013	-0,000013	-0,000013

<sup>8</sup> Une autre solution consiste à réaliser deux contrôleurs PI indépendants, gérant chacun une variable ( $T_{su}$  ou  $\dot{V}$ ). Cette solution a été explorée et donne de très bons résultats mais n'est malheureusement pas capable de prendre en compte l'optimisation économique.

<sup>9</sup>  $m = minimum$ ,  $M = maximum$  et  $moy = moyenne arithmétique$

A nouveau, les contrôleurs PI du modèle simplifié (5.10) ont été paramétrés grâce au module *Control and Design Toolbox* de *Matlab* pour toutes les valeurs possibles de  $T_{out}$  variant de sa valeur minimale à maximale avec les mêmes conditions que celles définies dans les chapitres 3 et 4 :

- Une bande passante du système initial doublée pour accélérer la réponse tout en évitant les évolutions trop brutales pour le confort des occupants ;
- Une marge de phase maximale de  $90^\circ$  pour garantir une stabilité maximale au système boucle fermée.

Encore une fois, la température extérieure n'intervient pas dans le paramétrage du contrôleur PI. Celui-ci sera donc extrêmement simple puisqu'il est uniquement constitué de deux paramètres constants :

$$\begin{cases} K_p = -0,0325 \\ K_i = -0,000013 \end{cases} \quad (5.12)$$

Pour finir, une boucle anti-windup a encore une fois été ajoutée pour éviter tout emballement du terme intégral [2] :

$$K_s = \frac{1}{\sqrt{K_i}} = \frac{1}{\sqrt{-0,000013}} \quad (5.13)$$

Les modèles *Simulink* présentent un saturateur sur  $T_{su}$  et un sur  $\dot{V}$ . Il est obligatoire d'en prévoir un sur chaque variable puisque le saturateur sur  $\lambda$  ne représente que les sommes des pentes aux extremums de  $T_{su}$  et  $\dot{V}$ . Il est donc théoriquement possible d'avoir  $T_{su}$  ou  $\dot{V}$  qui atteint sa valeur limite de saturation sans que  $\lambda$  ne le soit.

Cette saturation ne posera en réalité pas de problème puisque le phénomène de windup n'apparaît que pour des commandes très largement supérieures aux valeurs de saturation des commandes ; ce qui est (presque) impossible en pratique. Un simple saturateur sur la variable intermédiaire  $\lambda$  suffit donc largement à éviter ce phénomène (en plus de pouvoir être aisément réalisé dans le bloc *PID* de *Simulink*).

Pour finir, cette boucle n'agit que sur  $\lambda$ . Pour obtenir une boucle anti-windup parfaite, il serait nécessaire de réaliser une boucle par variable de commande ( $T_{su}$  et  $\dot{V}$ ) qui viendraient indépendamment se sommer sur la partie intégrale. Malheureusement, aucune théorie classique ne traite cette difficulté [3]. De plus, le bloc *PID* de *Simulink* ne permet pas de réaliser cette contrainte. Il serait dès lors nécessaire de réaliser soi-même le contrôleur PID avec ses deux boucles anti-windup de manière manuelle. Le paramétrage de ces deux anti-windup devrait par ailleurs être réalisé de manière expérimentale puisqu'il n'existe encore aucune méthode théorique capable de gérer ce genre de systèmes MISO non-linéaires [3].

#### 5.4. Simulations et résultats MISO $T_{su}$ et $\dot{V}$

Les simulations finales du système simplifié (5.10) contrôlant  $\lambda$  sont illustrées aux figures 5.4 et 5.5. Celles-ci permettent de vérifier si le contrôleur finalement réalisé est bien capable de réguler la température dans le hall de production  $T_{in}$  à la consigne désirée pour l'ensemble des possibilités de température extérieures sur une année complète (de  $-7,6$  à  $38,5^\circ\text{C}$ ).

Même si ces résultats ne sont pas totalement optimaux à cause du passage du système complet non-linéaire (5.9) vers le système simplifié (5.10), les résultats de  $T_{in}$  restent parfaitement valides puisqu'ils sont basés sur l'équation (5.4).

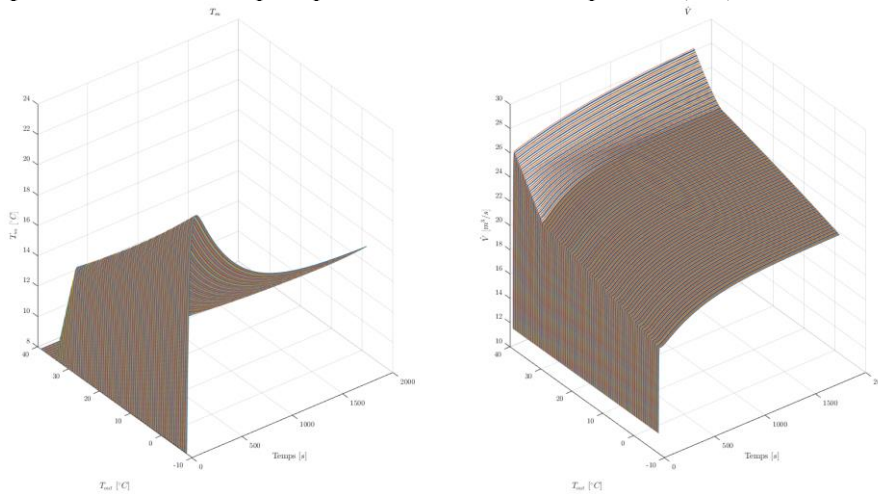


Figure 5.4 : Evolutions de  $T_{su}$  et  $\dot{V}$  en fonction de  $T_{out}$  du modèle MISO

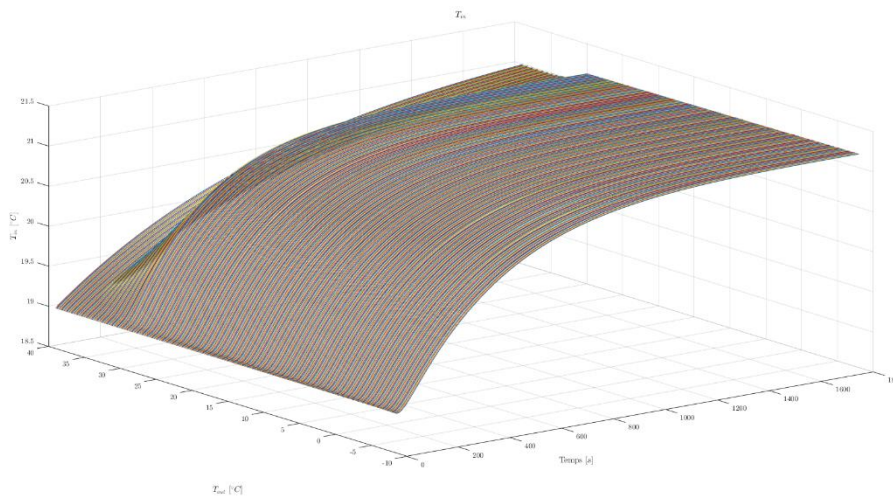


Figure 5.5 : Evolutions de  $T_{in}$  en fonction de  $T_{out}$  du modèle MISO

La figure 5.4 montre bien que  $T_{su}$  et  $\dot{V}$  varient simultanément en suivant les équations d'optimisation économiques définies au système (5.10) afin de garantir une température interne dans le hall passant de  $19^{\circ}\text{C}$  à  $21^{\circ}\text{C}$  quelque soit la température extérieure au bâtiment.

Jusqu'à une limite de  $30,3^{\circ}\text{C}$ , le système est adaptatif et diminue  $T_{su}$  tout en augmentant  $\dot{V}$  afin de contrer l'augmentation de la température extérieure. Cette variation n'est pas la même pour les deux variables puisque le système cherche à garder l'optimum économique. A partir de cette limite  $T_{su}$  est saturé à sa valeur minimum  $T_{su,min}$  et  $\dot{V}$  augmente donc plus rapidement pour contrer l'évolution de la température extérieure puisqu'il est la seule variable qui peut encore influencer sur le résultat du système.

La figure 5.5 montre, quant-à-elle, que le système est capable de réguler la température interne du hall de production aux  $21^{\circ}\text{C}$  désiré dans tous les cas possibles tout en optimisant les coûts. Lorsque la température  $T_{su}$  n'est pas saturée, sa réponse présente un dépassement assez faible durant un court laps de temps au vu des durées de simulation de  $30min$  avant de se stabiliser à une valeur convenable. La température interne, quant-à-elle, ne présente pas de dépassement et permet donc une évolution douce de la température interne garantissant un confort maximum aux occupants.

### 5.5. Conclusion modèle MISO $T_{su}$ et $\dot{V}$

Pour conclure le modèle MISO  $T_{su}$  et  $\dot{V}$ , il a été possible de transformer un système commandant les deux variables de commande  $T_{su}$  et  $\dot{V}$  (système MISO) en un simple système contrôlant uniquement  $\lambda$  (système SISO). Cette technique a permis de fortement simplifier le paramétrage du contrôleur.

Malheureusement, la résolution du véritable système non-linéaire (5.9) n'a pas été réalisable avec les outils disponibles et il a été nécessaire de fortement le simplifier afin d'obtenir le système (5.10) qui n'est plus pur mathématiquement mais qui, en échange, peut être résolu avec des méthodes classiques.

La résolution proposée permet finalement de garantir la température du hall tout en optimisant les coûts liés à ce refroidissement dans toutes les situations possibles de la gamme de température extérieure contrôlable (de  $-24,67^{\circ}\text{C}$  à  $41,91^{\circ}\text{C}$ ) et ce, sans présenter de montées/descentes brusques de la température interne.

L'optimisation des coûts de fonctionnement en se basant sur une méthode analogue à celle de la production/consommation des centrales électriques [4] montre qu'il n'est finalement pas impossible pour les entreprises d'optimiser le coût de leur contrôle de température dans les bâtiments sans nécessiter des modèles trop complexes.

## 6. Conclusion et améliorations

En conclusion, ce projet a permis d'optimiser efficacement les coûts de production de la température interne d'un hall de production  $T_{in}$  en contrôlant à la fois la température de soufflerie  $T_{su}$  mais également son débit  $\dot{V}$  via un simple contrôleur PI en se basant sur un modèle simplifié.

Pour réaliser un tel projet, il a été nécessaire d'augmenter progressivement la difficulté. La première étape consistait à uniquement contrôler la température de soufflerie  $T_{su}$  dans un modèle parfaitement linéaire. Par la suite, l'analyse d'un modèle non-linéaire contrôlant le débit de ventilation  $\dot{V}$  a permis de se rendre compte des difficultés de ces non-linéarités. Pour finir, ces deux modèles ont été associés afin d'obtenir un système contrôlant simultanément les deux variables de commande tout en y ajoutant une optimisation économique.

Ce régulateur final s'auto-paramètre en utilisant une optimisation mathématique des coûts de production d'énergie propre à chaque actionneur, tout en prenant en compte leurs limites de saturation. Le résultat est dès lors un contrôle efficace de la température interne, avec très peu de limites, tout en diminuant au maximum les coûts. Les seules limites sont uniquement les cas où il est de toute façon impossible de réguler avec le système de soufflerie installé.

Plusieurs améliorations peuvent être proposées pour rendre ce projet plus efficace :

- Résoudre réellement les équations non-linéaires du système d'équations (5.9) en utilisant des solveurs plus puissants. Ceci est réalisable étant donné que les équations d'optimisation (5.6) ne font apparaître que  $T_{su}$ ,  $\dot{V}$  et  $\lambda$ . La difficulté est de résoudre ce système d'équation non-linéaire de manière parfaite ;
- Utiliser des méthodes de contrôle plus avancées telles que la méthode discrète RST ou des techniques de contrôle prédictif bien plus efficaces [3] ;
- Coupler plusieurs mesures (actuellement le système ne mesure que  $T_{in}$ ) grâce à un filtre de Kalman pour obtenir une mesure globale bien plus précise [3] [11] ;
- Perfectionner le modèle en prenant en compte la non-homogénéité de la température dans la pièce avec un logiciel d'éléments finis tel qu'*Abaqus* ;
- L'étude économique se base uniquement sur les coûts de production. Il vaudrait mieux optimiser le Social Welfare qui comprend bien plus d'informations pondérées telles que le confort des occupants, l'impact écologique etc. [4] ;
- L'utilisation d'algorithmes avancés permettraient des calculs plus rapides et optimaux tels que la méthode de Newton-Raphson/Newton-Gauss (recherche du zéro d'une fonction multivariable), la méthode de Monte-Carlo (algorithme génétique où des variations aléatoires apparaissent à chaque génération et où l'évolution entre générations successives suit un mécanisme basé sur l'évolution Darwinienne), l'algorithme de Karmarkar (méthode du point intérieur où la solution optimale est obtenue de manière asymptotique) etc.



## 7. Sources

[1] HANUS Vincent, *Génie énergétique*, Notes de cours, Virton, Belgique : Haute École de Namur-Liège-Luxembourg (Hénallux) – Département Ingénieur Industriel de Pierrad-Virton, 2019, pp.159.

[2] LECOINTRE Julien et DELAITE Thomas, *Modélisation des processus*, Notes de cours, Virton, Belgique : Haute École de Namur-Liège-Luxembourg (Hénallux) – Département Ingénieur Industriel de Pierrad-Virton, 2019.

[3] VANDE WOUWER Alain, *Optimal estimation and control with application to bioengineering systems*, Notes de cours, Mons, Belgique : Université de Mons (Umons) – Faculté Polytechnique de Mons – Service d'Automatique, 2014.

[4] HENNEAUX Pierre et GENÊT Benjamin, *Electric Power Systems*, Notes de cours, Bruxelles, Belgique : Université Libre de Bruxelles (ULB) – École Polytechnique de Bruxelles – Department of Bio- Electro- And Mechanical Systems (BEAMS) – Energy Group, 2019.

[5] HERR Horst, *Génie énergétique et climatique - Chauffage, froid, climatisation*, Dunod, 2014.

[6] FORTEMPS Philippe et PIRLOT Marc, *Mathématique pour l'Ingénieur 3*, Notes de cours, Belgique : Université de Mons (Umons) – Faculté Polytechnique de Mons – Service de Mathématique et de Recherche opérationnelle, 2009.

[7] SAUCEZ Philippe Saucez, *Analyse numérique*, Notes de cours, Mons, Belgique : Université de Mons (Umons) – Faculté Polytechnique de Mons – Service de Mathématique et de Recherche opérationnelle, 2011.

[8] TINOCO DE RUBIRA Tomás, *Numerical optimization and modeling techniques for power system operations and planning*, PhD thesis, Stanford, USA – California : Stanford University – Department of electrical engineering, 2015.

[9] INTERNATIONAL BUSINESS MACHINES CORPORATION, *Cplex Optimizer*, 2020.  
Adresse URL : <https://www.ibm.com/analytics/cplex-optimizer>.

[10] LÖFBERG Johan, *Yalmip*, 2020.  
Adresse URL : <https://yalmip.github.io>.

[11] FARAGHER Ramsey, *Understanding the basis of the Kalman filter via a simple and intuitive derivation*  
IEEE Signal processing magazine, 29 (5), septembre 2012, pp. 128-132.