Hierarchical predictive control. Application to reheating furnaces in the steel industry.

Xuan Manh Nguyen

N° d'ordre : 2015-09-TH

CentraleSupélec

ECOLE DOCTORALE STITS
« Sciences et Technologies de l'Information des Télécommunications et des Systèmes »

THÈSE DE DOCTORAT

DOMAINE : STIC
Spécialité : Automatique

Soutenue le 18 Mai 2015

par:

Xuan Manh NGUYEN

Commande prédictive hiérarchisée. Application à la commande de fours de réchauffage sidérurgiques

Directeur de thèse :	Didier DUMUR	Professeur, CentraleSupélec
Co-encadrant :	Fayçal LAWAYEB	Ingénieur, ArcelorMittal
Co-encadrant :	Alain MOUCHETTE	Ingénieur, ArcelorMittal
Co-encadrant :	Pedro RODRIGUEZ-AYERBE	Professeur Adjoint, CentraleSupélec

Composition du jury :

Président du jury :	Dorothée NORMAND-CYROT	Directeur de Recherche CNRS, L2S Gif sur Yvette
Rapporteurs :	Edouard LAROCHE	Professeur des Universités, Université de Strasbourg
	Pierre RIEDINGER	Professeur des Universités, CRAN Nancy
Examinateurs :	Estelle COURTIAL	Maitre de Conférences, Université d'Orléans
	Mohammed M'SAAD	Professeur des Universités, GREYC Caen

Acknowledgement

I would like first to thank my supervisors Mr. Didier Dumur, Mr. Pedro Rodriguez-Ayerbe, Mr. Fayçal Lawayeb, and Mr. Alain Mouchette who have guided and encouraged me since the beginning of my doctoral program. I am grateful for their help, their patience, their availability to correct my last minute papers and presentations.

I am very thankful to Mr. Edouard Laroche and Mr. Pierre Riedinger for reviewing my thesis and to Ms. Dorothée Normand-Cyrot, Ms. Estelle Courtial and Mr. Mohammed M'saad for being part of the examination committee of my thesis.

I also would like to thank all professors and personnel of the Automatic Control Department of CentraleSupélec, especially to Ms. Josiane Dartron, Ms. Pascale Lepeltier and Mr. Léon Marquet for their help during my three years at CentraleSupélec. I will not forget my colleagues who supported me with their pleasant presence during coffee breaks and lunch time, specially: Dung, Tri, Ngoc Anh, Binh, Djawad, Imad, Séif and Quang.

It is also a pleasure for me to thank the personnel of the Measurement and Control Department of ArcelorMittal R&D Center, especially to Ms. Laurette Duarte Domingues for her administrative support during my stays at the company.

I am particularly thankful to the colleagues of automatic control team from whom I have learned a lot about technical issues, about organizational and professional skills.

Finally, I am grateful to my girlfriend, my parents, my sister and my friends for their support and their encouragements.

Publications

Les travaux effectués lors de cette thèse, exposés dans ce mémoire, ont donné lieu à trois publications lors des congrès internationaux avec actes :

Conférences internationales avec actes

- X.M. Nguyen, P. Rodriguez-Ayerbe, D. Dumur, A. Mouchette, F. Lawayeb, “Distributed Model Predictive Control of Steel Slab Reheating Furnace”, Proceedings of the IFAC Workshop on Automation in the Mining, Mineral and Metal Industries, MMM 2012, Nagaragawa, Gifu Japon, 10-12 September 2012, pp. 175-180.

- X.M. Nguyen, F. Lawayeb, A. Mouchette, “Advanced Model Predictive Control "MPC" for steel reheating furnace: performance improvement and energy savings”, European Steel Technology & Application Days, Paris, France, 7-8 April 2014, pp. 196-197.

- X.M. Nguyen, F. Lawayeb, A. Mouchette, P. Rodriguez-Ayerbe, D. Dumur, “Nonlinear model predictive control of steel slab walking-beam reheating furnace based on a numerical model”, Proceedings of the IEEE Multi-Conference on Systems and Control 2014 (IEEE MSC 2014), Nice/Antibes, France, 8-10 October 2014, pp. 191-196.

- X.M. Nguyen, P. Rodriguez-Ayerbe, D. Dumur, A. Mouchette, F. Lawayeb, “Temperature control of reheating furnace based on distributed model predictive control”, Proceedings of 18th International Conference on System Theory, Control and Computing ICSTCC 2014, Sinaia, Romania, 17-19 October 2014, pp. 732-737.

Résumé de thèse

COMMANDE PRÉDICTIVE HIÉRARCHISÉE. APPLICATION À LA COMMANDE DE FOURS DE RÉCHAUFFAGE SIDÉRURGIQUES

Table des matières

1. Introduction

La sidérurgie a en particulier pour but de transformer le minerai et la ferraille de fer en acier. Pour cela, le fer doit subir plusieurs processus thermiques, mécaniques et thermomécaniques avant de devenir adaptés à d'autres applications industrielles telles que la construction, l'automobile ou l'industrie spatiale. Ces processus peuvent être regroupés en quatre étapes : la fabrication du fer, la fabrication de l'acier, la coulée continue et le laminage à chaud. Au cours de la fabrication du fer, le liquide de fer est produit à partir du minerai et de la ferraille. A ce liquide est ensuite ajouté différents éléments comme le nickel, ou le chrome pour fabriquer l'acier liquide. La coulée continue permet ensuite d'obtenir à partir de l'acier liquide des brames d'acier qui seront transformées selon les besoins du client. Enfin, la fonction principale du processus de laminage à chaud est de réchauffer les brames d'acier à une température proche de leur point de fusion, puis de les laminer par passage dans plusieurs laminoirs successifs et enfin d'enrouler les tôles d'acier sous forme de bobines, voir Fig. 1.

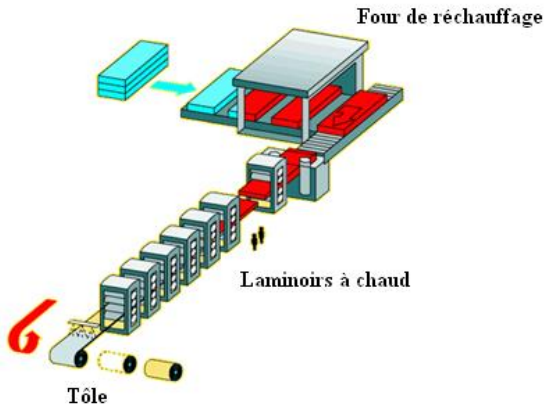

Fig. 1 : Laminage à chaud

Les fours de réchauffage sont positionnés juste avant les laminoirs à chaud, et utilisés pour le réchauffage, ainsi que pour le traitement métallurgique des brames. Les brames d'acier sont habituellement chauffées de la température ambiante jusqu'à une température visée de 1200°C. La performance d'un four de réchauffage est évaluée par la précision de la température de défournement, par l'homogénéité de la température à l'intérieur des brames et par le rendement énergétique du four. Certains phénomènes tels que la formation de calamine, la décarburisation, la fonte de surface ou la distorsion géométrique des brames doivent être réduits autant que possible.

Dans une usine sidérurgique, les fours de réchauffages sont classés deuxième consommateurs d'énergie avec une consommation représentant 17% de l'énergie utilisée dans la production de l'acier. La consommation de gaz du four considéré dans l'étude est équivalente à la consommation d'électricité de 100,000 foyers. De fait, l'efficacité énergétique du four de réchauffage a une influence directe sur le coût de production. Par ailleurs, une réduction de la consommation de gaz entraîne une diminution des émissions de gaz à effet de serre ainsi que du CO_2, N_2O, etc, émissions objet de régulations de plus en plus restrictives. Plusieurs facteurs de conception affectent l'efficacité énergétique du four, tels que la géométrie du four, le système d'isolation thermique, et en particulier, la régulation de la température dans le four.

De nos jours, la diversification des produits fait intervenir une grande variété de brames en termes de matériaux, de dimensions, et de températures visées. Par conséquent, une mauvaise régulation de la température du four oblige les opérateurs à imposer de grandes marges de température afin d'assurer un fonctionnement satisfaisant du four. Cela induit une augmentation importante de la consommation en termes d'énergie.

Pour toutes ces raisons, l'objectif des travaux de cette thèse est d'améliorer la commande de la température du four afin d'en augmenter son efficacité énergétique et la qualité de réchauffage des produits.

Pour réaliser la tâche de réchauffage, les fours sont habituellement construits en plusieurs zones équipées de bruleurs à gaz comme indiqué Fig. 2 (typiquement récupération, préchauffage, chauffage, égalisation). Les longerons permettent de déplacer et donc de transporter les brames à travers le four. Le mur réfractaire est l'élément d'isolation thermique de four. La fumée issue de la combustion des gaz est évacuée à l'entrée du four, cela permet de récupérer une partie de la chaleur afin de chauffer les brames situées dans la zone de récupération. Le chauffage est réalisé dans les zones de préchauffage et de chauffage principal. La température à l'intérieur des brames est finalement homogénéisée dans la zone dite d'égalisation.

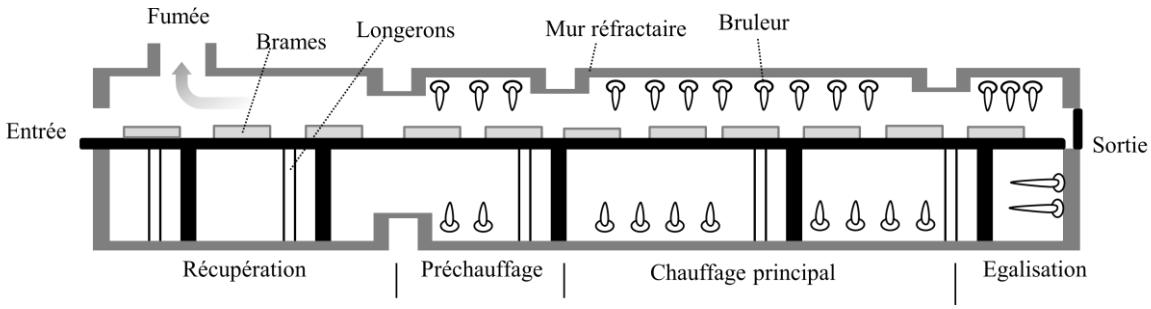

Fig. 2 : Vue latérale du four de réchauffage

Le but de la régulation de la température des fours de réchauffage peut être décrit de façon simplifiée comme étant la stratégie à développer pour assurer la meilleure distribution de l'énergie thermique le long du four afin de chauffer les brames à la bonne température de défournement. Les variables disponibles pour effectuer cette régulation de la température sont les débits de gaz et d'air des bruleurs et, plus indirectement, les consignes de température de zones. Dans certains cas, il est également possible d'ajuster la cadence des brames dans le four à condition que cela ne dépasse pas par la suite la capacité des laminoirs. Pour gérer la complexité du système, la structure actuelle de commande du four exploite l'architecture cascade hiérarchisée représentée Fig. 3. Au niveau 2, la température visée de brames est contrôlée à l'aide d'un modèle de transfert thermique du four, à ce niveau, les consignes de température de zones sont calculées et envoyées ensuite à niveau 1. A partir des consignes calculées par le niveau 2, le niveau 1 détermine les puissances nécessaires de zones permettant d'atteindre ces consignes.

Dans le four considéré, le niveau 2 du système de commande actuel est fondé sur un modèle de transfert thermique stationnaire. Ce modèle ne permet pas d'estimer la consommation du four. A cet effet, un choix lié à la précision de la température de défournement est réalisé, et aucun critère d'efficacité énergétique n'est considéré. Au niveau 1, des régulateurs PID traditionnels sont mis en œuvre pour le suivi des températures des zones. Cependant, ces régulateurs PID ne peuvent pas prendre en compte l'interaction entre les zones du four, qui représente pourtant un phénomène non négligeable à l'intérieur du four. Par conséquent, le chauffage des brames n'est pas optimisé, ce qui induit un faible rendement énergétique du four.

Afin d'améliorer la qualité de chauffage et de réduire la consommation d'énergie, ce travail propose une nouvelle structure de commande illustrée Fig. 3. Les régulateurs PID de niveau 1 y sont remplacés par une stratégie de commande prédictive distribuée permettant la prise en compte de l'interaction entre les zones du four. En outre, au niveau 2, le modèle d'état stationnaire est remplacé par un modèle dynamique non linéaire ; ce modèle peut prédire le comportement transitoire du four, permettant une meilleure prévision et l'estimation de la température des brames. Le travail vise principalement la qualité de chauffage et la réduction de la consommation énergétique du four. Toutefois, une réduction de la calamine et de la décarburisation peut être également espérée. De plus, la flexibilité de production est étudiée afin d'augmenter la productivité du four.

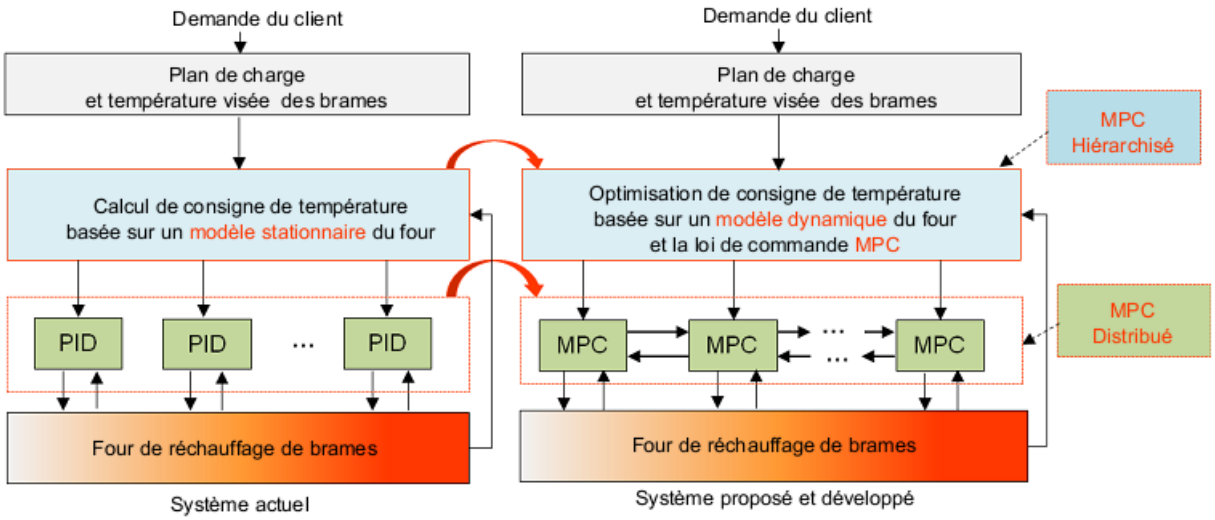

Fig. 3 : Approche proposée pour la commande du four

2. Modélisation du four

Au sein du four, les variables à contrôler sont principalement les profils de températures de brames. Ces variables ne peuvent être mesurées directement. La modélisation du four est une étape fondamentale permettant par la suite une estimation la meilleure possible des températures de brames. Un modèle de transfert thermique est donc nécessaire, se fondant sur la connaissance des phénomènes de conduction, de rayonnement et de convection dans le four.

2.1. La conduction thermique

Sachant que la largeur et la longueur des brames sont largement plus importantes que l'épaisseur, seul le problème de conduction dans l'épaisseur de la brame est considéré. La Fig. 4 illustre les coordonnées associées à la brame *i*, de dimensions (W_i x L_i x D_i). La température en un point de la brame dépend uniquement de sa position *y* et du temps *t*, notée donc $T_i(y, t)$ (avec $-D_i/2 \leq y \leq D_i/2$) pour désigner la température d'un point de la brame *i*.

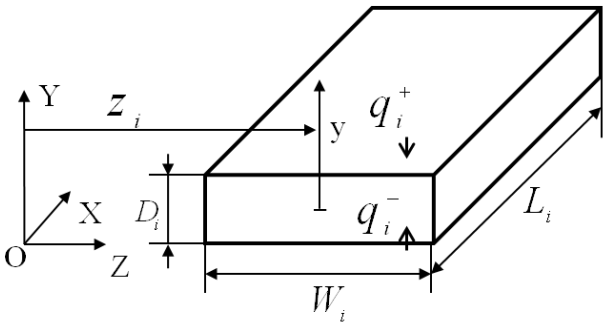

Fig. 4 : Coordonnées de la brame *i*

Les termes $q_i^+(t)$ et $q_i^-(t)$ désignent les débits thermiques absorbés par les surfaces supérieure et inférieure de la brame, respectivement. Le débit thermique $q_i(y,t)$ à l'intérieur de la brame dépend de la capacité thermique du matériau, de la conductivité thermique, du gradient de température à la position *y*, et des conditions aux limites aux deux surfaces de la brame. Le problème de conduction suit la loi de *Fourier* et la loi de diffusion thermique :

$$q_i(y,t) = -\lambda_i \frac{\partial T_i(y,t)}{\partial y} \quad (1)$$

$$\nabla q_i\,(y,t) + \rho_i c_i \frac{\partial T_i(y,t)}{\partial y} = 0 \tag{2}$$

A partir des équations (1) et (2), nous obtenons l'équation décrivant la conduction de la chaleur dans l'épaisseur de la brame :

$$\rho_i c_i \frac{\partial T_i(y,t)}{\partial t} = \frac{\partial}{\partial y}\left(\lambda_i \frac{\partial T_i(y,t)}{\partial y}\right) \tag{3}$$

Les conditions initiales concernent le profil de température de la brame au moment de son enfournement $t_{i,0}$:

$$\begin{aligned} &T_i(y,t_{i,0}) = T_{i,0}(y), \\ &y \in [-D_i/2, D_i/2] \end{aligned} \tag{4}$$

Les conditions aux limites sont exprimées par :

$$\begin{aligned} -\lambda_i \frac{\partial T_i}{\partial y}\bigg|_{y=D_i/2} &= q_i^{+}(t), \\ -\lambda_i \frac{\partial T_i}{\partial y}\bigg|_{y=-D_i/2} &= q_i^{-}(t). \end{aligned} \tag{5}$$

Afin de résoudre l'équation (3), la variable de *Kirchhoff* $\theta(T_i)$ et l'enthalpie $H(T_i)$ sont utilisées et définies par :

$$\theta(T_i) = \lambda_i \int_0^{T_i} dT \tag{6}$$

$$\rho_i \frac{\partial H}{\partial t} = \frac{\lambda_i \partial\left(\partial T_i/\partial y\right)}{\partial y} \tag{7}$$

Les équations (3) et (5) deviennent alors :

$$\rho_i \frac{\partial H}{\partial t} = \nabla^2 \theta \tag{8}$$

$$q_i^{+}(t) = \frac{\partial \theta}{\partial y}\bigg|_{y=D_i/2}$$

$$q_i^{-}(t) = \frac{\partial \theta}{\partial y}\bigg|_{y=-D_i/2} \tag{9}$$

Une discrétisation spatiale et temporelle est nécessaire pour résoudre (8) et (9). Par exemple, si nous calculons la température en $2n+1$ points dans l'épaisseur de la brame, l'échantillonnage spatial est défini par $\Delta y = D_i/2n$. La méthode aux différences finies peut être ensuite utilisée pour déterminer *H* en fonction du temps et la variable de *Kirchhoff*.

Pour résoudre l'équation de conduction dans l'épaisseur de la brame, on peut également utiliser la méthode des volumes finis en intégrant l'équation (3) dans l'espace et dans le temps (Fig. 5), à savoir :

$$\rho_i c_i \int_t^{t+\Delta t}\int_w^e \frac{\partial T_i(y,t)}{\partial t} dy dt = \int_w^e \int_t^{t+\Delta t} \frac{\partial}{\partial y}\left(\lambda_i \frac{\partial T_i(y,t)}{\partial y}\right) dt dy \tag{10}$$

Cette méthode conserve la masse, et par conséquent est plus précise que la méthode des différences finies.

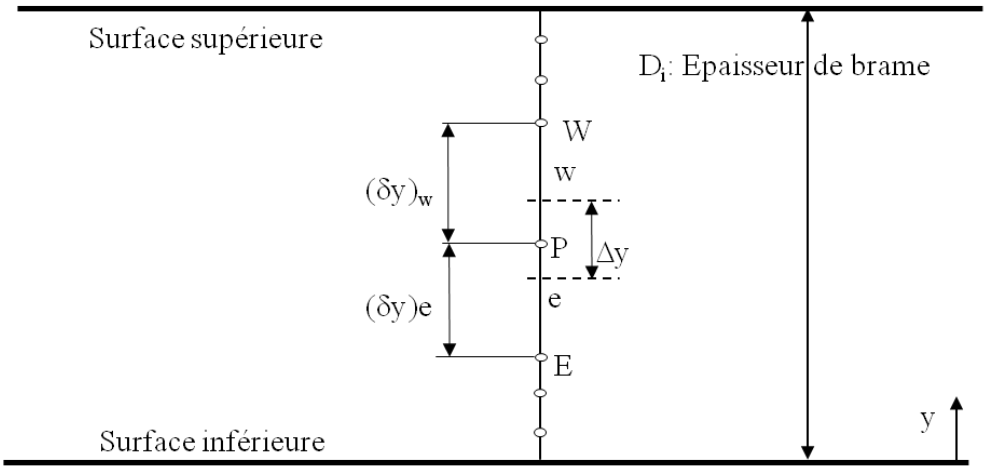

Fig. 5 : Discrétisation spatiale dans l'épaisseur de la brame

2.2. Le rayonnement thermique

Le rayonnement est le phénomène de transfert thermique dominant dans le four de réchauffage, comme indiqué Fig. 6. Environ 90% du débit thermique absorbé par les surfaces des brames est issu du rayonnement. Actuellement, le modèle du four considéré utilise une méthode de calcul stationnaire pour le rayonnement thermique. Les hypothèses de cette méthode sont :

- seules les surfaces supérieure et inférieure sont prises en compte dans l'échange thermique
- l'épaisseur des brames n'est pas prise en compte
- une surface d'échange thermique est homogène en température, et réflecteur idéal
- la température de fumée est homogène le long du four
- tout échange radiatif se produit entre la fumée, les brames et la zone correspondante

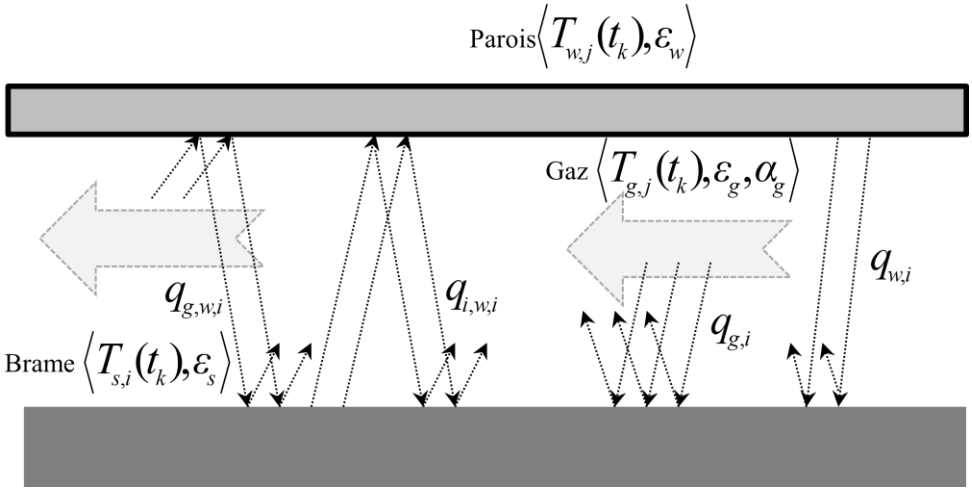

Fig. 6 : Rayonnement thermique dans un four de réchauffage

Avec ces hypothèses, le problème de rayonnement thermique dans une zone est le même que celui dans un espace fermé, comme illustré Fig. 7.

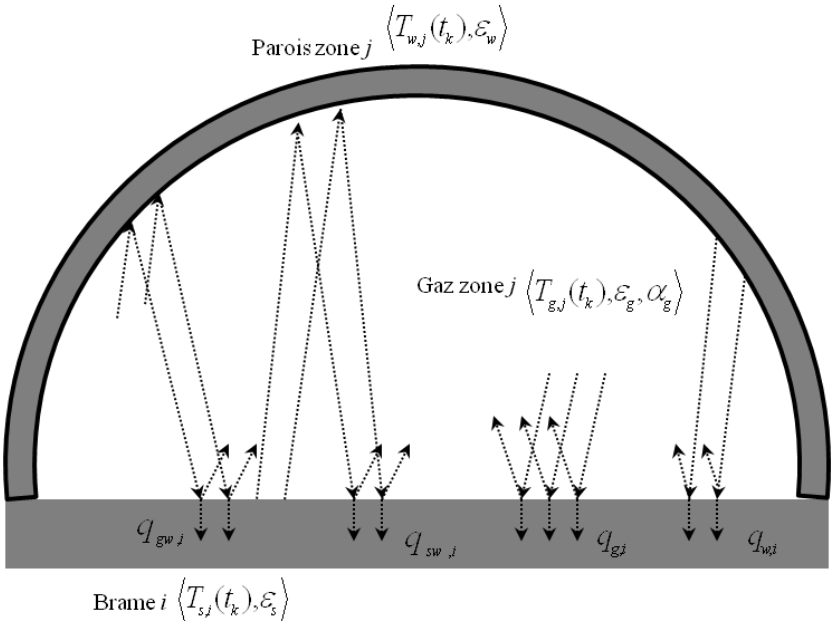

Fig. 7 : Espace fermé équivalent d'une zone utilisé par la méthode stationnaire

Les débits thermiques absorbés par les surfaces d'une brame dans une zone sont constitués des éléments suivants :

- débit issu du mur reçu par la brame $q_{w,i}$
- débit issu de la fumée, réfléchi sur le mur et reçu par la brame $q_{gw,i}$
- débit issu de la brame, réfléchi sur le mur et reçu par la brame elle-même $q_{sw,i}$
- débit issu de la fumée, reçu directement par la brame $q_{g,i}$

$$q_i^+ = q_i^- = q_i = q_{g,i} + q_{w,i} + q_{gw,i} + q_{sw,i} \tag{11}$$

Le rayonnement suit la loi de *Stefan-Boltzmann* appliquée aux corps gris. Les éléments de (11) sont calculés par :

$$q_{g,i} = \sum_{k=1}^{n} q_{g,i,k} = \varepsilon_s A_g \frac{1}{1-a} = \frac{\varepsilon_s \varepsilon_g \sigma T_{g,j}^4}{1-(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2} \tag{12}$$

$$q_{w,i} = \varepsilon_s A_w \frac{1}{1-a} = \frac{\varepsilon_s (1-\alpha_g) \varepsilon_w \sigma T_{w,j}^4}{1-(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2} \tag{13}$$

$$q_{gw,i} = \varepsilon_s A_{gw,i} \frac{1}{1-a} = \frac{\varepsilon_s (1-\alpha_g)(1-\varepsilon_w) \varepsilon_g \sigma T_{g,j}^4}{1-(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2} \tag{14}$$

$$q_{sw,i} = \varepsilon_s A_{sw,i} \frac{1}{1-a} = \frac{\varepsilon_s (1-\varepsilon_w)(1-\alpha_g)^2 \sigma T_{s,i}^4}{1-(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2} \tag{15}$$

où $\varepsilon_s, \varepsilon_g, \varepsilon_w$ sont les coefficients d'émissivité de la brame, de la fumée et du mur, respectivement. α_g est le coefficient d'absorption de la fumée. $T_{g,j}$, $T_{w,j}$, $T_{s,i}$ sont les température de la fumée, du mur de la zone *j* et de la brame *i*.

La démarche tenant compte du comportement transitoire considère le rayonnement comme un phénomène global. L'intérieur du four est composé de plusieurs surfaces élémentaires grises et diffusantes S_j de température T_j et de coefficient d'émissivité ε_j (Fig. 8). Cette méthode nécessite des calculs numériques, à l'opposé des formules analytiques données par une approche stationnaire.

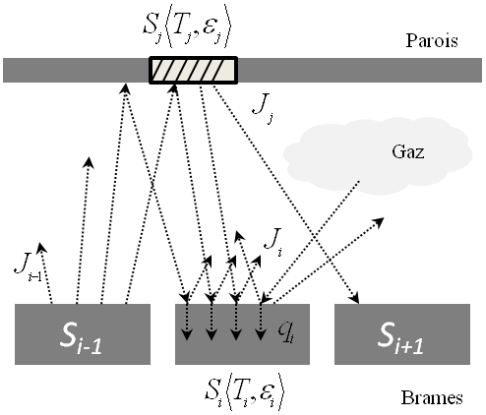

Fig. 8 : Rayonnement thermique transitoire dans un four de réchauffage

2.3. La convection thermique

La convection représente environ 5% des échanges thermiques entre les brames et le four. Elle se produit quand il y a mouvement de la fumée à l'intérieur du four. La loi de refroidissement de *Newton* décrit la convection comme dépendant de la vitesse de la fumée, et de la différence entre la température de la fumée et la température de la brame, à savoir :

$$q_{c,i}^{\pm}(t) = h_{c,j}^{\pm}\left(T_{g,j}^{\pm}(t) - T_{s,i}(\pm D_j/2, t)\right) \tag{16}$$

où les coefficients de convection $h_{c,j}^{\pm}$ sont proportionnels à la vitesse de la fumée.

3. Conception de la commande prédictive

La commande prédictive, souvent appelée MPC, est une méthode de commande avancée qui suscite un engouement fort à la fois du point de vue théorique et par les nombreuses applications industrielles. L'idée de base du MPC consiste à utiliser un modèle du système pour prédire le comportement des sorties à contrôler. En particulier, le MPC est capable de gérer des contraintes du système et de traiter les problèmes de commande multivariable. Le problème de commande est très souvent (en particulier en présence de contraintes) transformé en un problème d'optimisation, faisant intervenir pour notre application des critères économiques comme la réduction de la consommation énergétique ou l'amélioration de la qualité des produits.

Dans le cas de la commande des fours de réchauffage et avec l'approche proposée dans les travaux de thèse, le MPC est utilisé pour optimiser la commande des deux niveaux principaux. Au niveau 1, une structure MPC distribuée est construite comme le montre la Fig. 9. Dans cette structure, chaque zone du four est pilotée par un régulateur MPC local. Malgré tout, les zones interagissent du fait du mouvement de la fumée et de l'effet du rayonnement thermique. En conséquence, à l'opposé des régulateurs PID traditionnels, la structure MPC distribuée est développée pour prendre en compte les interactions entre les zones du four. Deux approches sont envisagées : une structure MPC distribuée coopérative et une stratégie MPC non-coopérative. Dans l'approche non-coopérative, chaque régulateur MPC local élabore la puissance de zone nécessaire afin d'optimiser la température de la zone en prenant en compte les influences des autres zones. Dans l'approche coopérative, chaque régulateur MPC local vise à optimiser non seulement la performance de la zone correspondante mais aussi celles des autres zones, particulièrement les zones voisines ayant des couplages directs avec la zone considérée.

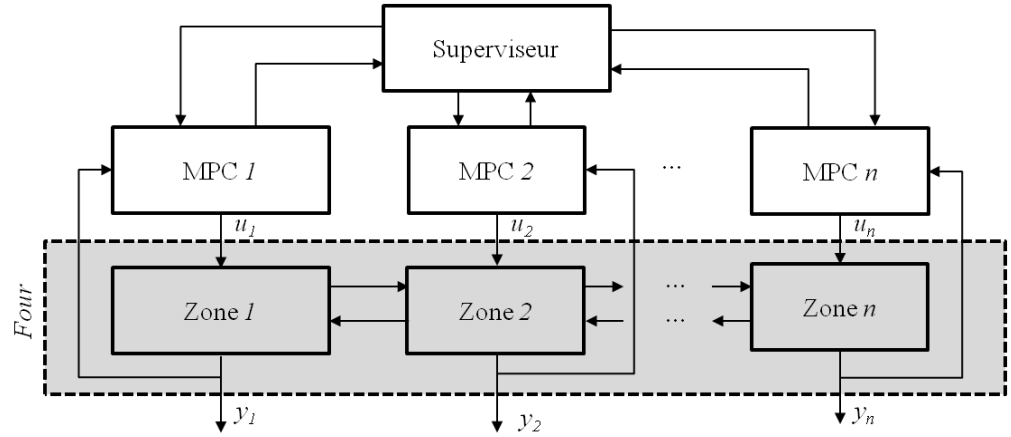

Fig. 9 : Structure distribuée de la commande MPC

L'approche MPC distribuée nécessite tout d'abord une modélisation de la dynamique de la température des zones, ainsi que des couplages entre zones. Les algorithmes permettant la communication et la coopération entre les régulateurs MPC locaux sont ensuite élaborés.

Au niveau 2, le contrôle de température des brames est réalisé par l'optimisation des consignes de température des zones. Différents critères sont pris en compte, comme la précision de la température de sortie des brames, l'homogénéité de la température à l'intérieur des brames, la consommation énergétique. Pour cela, le modèle précédent de rayonnement transitoire est utilisé afin de mieux capturer la dynamique du four et d'être capable également d'estimer la consommation du four, ce qui n'est pas permis par le modèle stationnaire de rayonnement thermique. Les contraintes du four sont prises en compte dans le problème d'optimisation sous forme de pénalités.

Les critères correspondant à chaque brame y sont pondérés en fonction de l'ordre de sortie de la brame. Les brames situées dans les zones de fin du four sont considérées plus importantes lors de l'optimisation parce qu'il reste moins de temps pour les chauffer à la bonne température. Par ailleurs, les brames situées dans la zone d'égalisation sont prioritaires pour atteindre l'homogénéité en température. Les puissances consommées des zones sont également pondérées. Dans une simulation du four, plus on avance dans l'horizon de prédiction, moins la puissance estimée par le modèle sera pondérée dans les critères d'optimisation.

Les paragraphes suivants détaillent la construction du MPC pour les deux niveaux.

3.1. Régulation de la température de zone

Des essais expérimentaux réalisés sur le four montrent que l'évolution de la température des zones du four se comporte comme un système du premier ordre retardé comme le montre la Fig. 10 sur un cas particulier. Les puissances u_1, u_3 y sont maintenues constantes pendant que u_2 est alimentée par un échelon. Le système global peut dès lors être identifié par des fonctions de transfert du premier ordre avec retard comme donné ci-dessous :

$$\begin{aligned} Y_1(s) &= \frac{0.0143e^{-60s}}{1+605s}U_1(s) + \frac{0.210e^{-20s}}{1+500s}Y_2(s) \\ Y_2(s) &= \frac{0.012e^{-40s}}{1+742s}U_2(s) + \frac{0.16e^{-20s}}{1+416s}Y_3(s) \\ Y_3(s) &= \frac{0.0066e^{-70s}}{1+415s}U_3(s) \end{aligned} \tag{17}$$

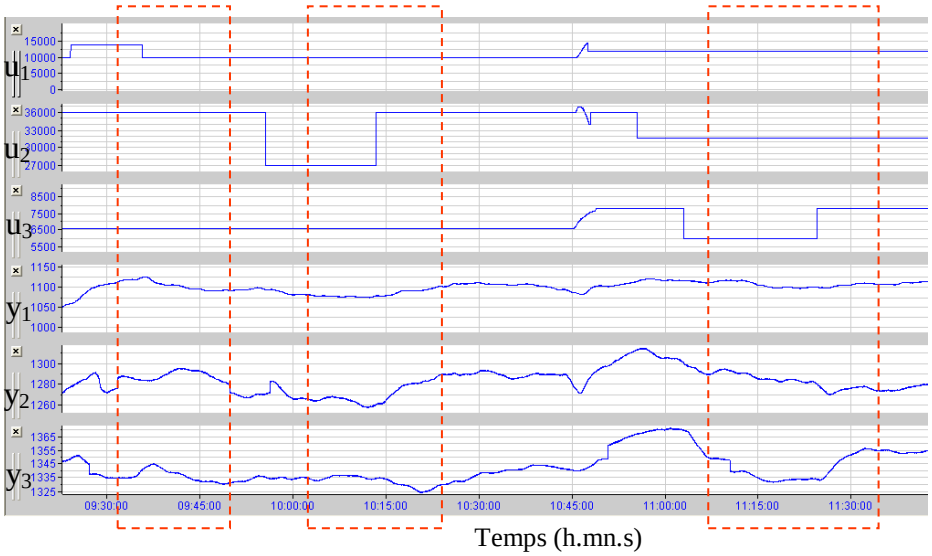

Fig. 10 : Réponse du premier ordre de la température des zones

où y_1, y_2, y_3 et u_1, u_2, u_3 représentent les variations des températures et puissances des zones par rapport au point de fonctionnement du four. Ce modèle traduit un couplage par les sorties. Pour construire la loi de commande MPC distribuée, il convient de transformer (17) sous forme de représentation d'état comme suit :

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) \sum_{j=1, j\neq i}^{M} B_{ij} y_j(k) \\ y_i(k) = C_i x_i(k) \end{cases} \tag{18}$$

où M est le nombre de zones du four. La relation (18) permet alors de prédire les sorties du système sur l'horizon de prédiction N_P à partir d'une séquence de commande sur l'horizon de commande N_u, comme suit :

$$\hat{y}_i(k+s|k) = C_i A_i^s x_i(k) + \sum_{p=1}^{\min(s,N_u)} C_i A_i^{s-p} B_i u_i(k+p-1|k) + \sum_{j\neq i}\sum_{p=1}^{s} C_i A_i^{s-p} B_{ij} \hat{y}_j(k+p-1|k), s = 1,\ldots,N_p \tag{19}$$

Dans une approche non-coopérative, le régulateur local de la zone i résout le problème de minimisation suivant :

$$\min_{u_i(k|k),\ldots,u_i(k+N_u-1|k)} \left\{ J_i(k) = \sum_{s=1}^{N_p} q_{i,s} \left[\hat{y}_i(k+s|k) - y_{i,ref}(k+s) \right]^2 + \sum_{p=0}^{N_u-1} r_{i,p+1} u_i^2(k+p|k) \right\},$$

Sous contraintes

$$\begin{aligned} & u_{i,\min} \le u_i(k+p) \le u_{i,\max}; p = 0,\ldots,N_u - 1, \\ & \Delta u_{i,\min} \le \Delta u_i(k+p) \le \Delta u_{i,\max}; p = 0,\ldots,N_u - 1, \\ & y_{i,\min} \le \hat{y}_i(k+s|k) \le y_{i,\max}; s = 1,\ldots,N_p. \end{aligned} \tag{20}$$

Pour cela, un algorithme itératif est mis en œuvre, décrit par l'organigramme de la Fig. 11.

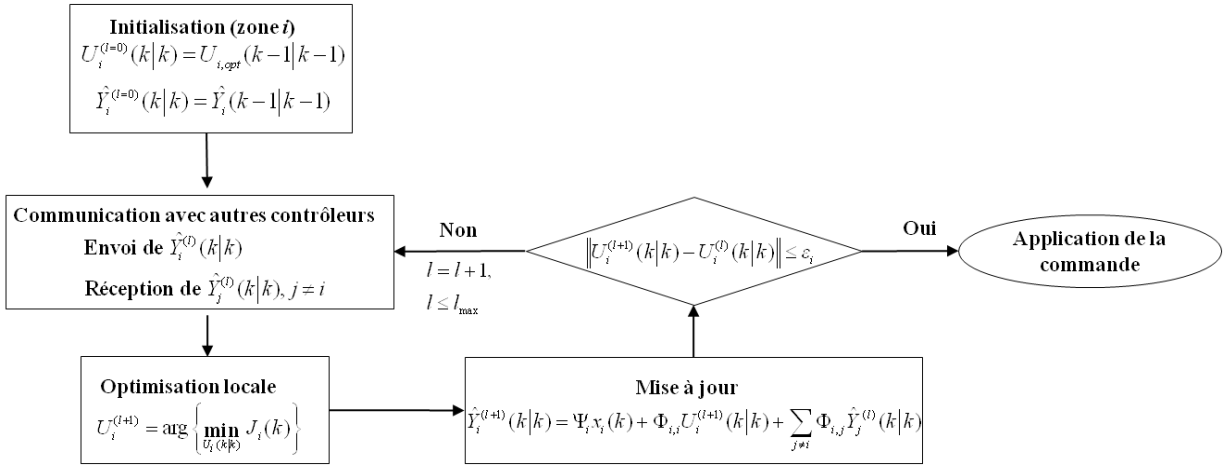

Fig. 11 : Algorithme non-coopératif

La communication entre régulateurs locaux est décrite Fig. 12. A chaque instant *k*, chaque régulateur communique les prédictions de la température de la zone correspondante et reçoit celles envoyées par les autres régulateurs. A partir des informations reçues, chaque régulateur MPC résout son problème d'optimisation locale, et ensuite met à jour les prédictions de température. La première valeur de la séquence de commande après convergence est ensuite appliquée à la zone correspondante.

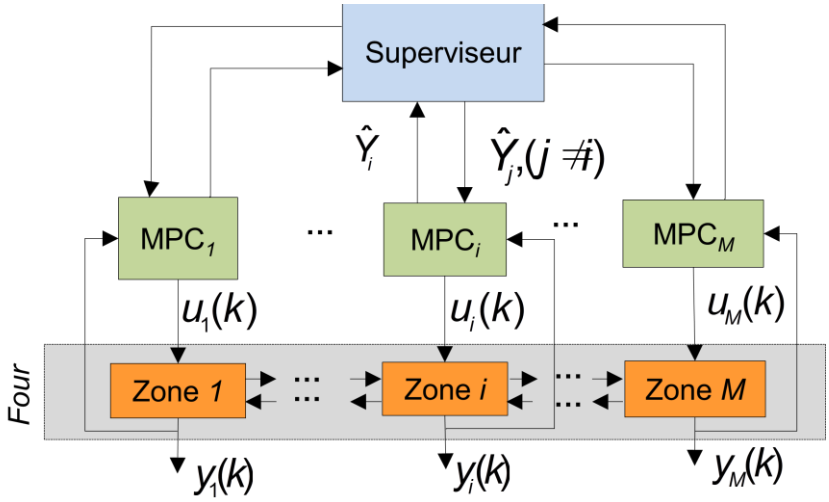

Fig. 12: Communication au sein de la structure MPC distribuée non-coopérative

Dans une approche coopérative, le régulateur de la zone *i* résout le problème de minimisation suivant :

$$\min_{U_i(k|k)} \left\{ J_{i\pm}(k) = \sum_{j=i-1}^{i+1} \left\| \hat{Y}_j(k|k) - Y_{j,ref}(k) \right\|_{Q_j}^2 + \left\| U_i(k|k) \right\|_{R_i}^2 \right\},$$

Sous contraintes

$$\begin{aligned} & u_{i,\min} \leq u_i(k+p) \leq u_{i,\max}; p = 0,\ldots,N_u - 1, \\ & \Delta u_{i,\min} \leq \Delta u_i(k+p) \leq \Delta u_{i,\max}; p = 0,\ldots,N_u - 1, \\ & y_{j,\min} \leq \hat{y}_j(k+s|k) \leq y_{j,\max}; s = 1,\ldots,N_p,\ j = i, i \pm 1. \end{aligned} \tag{21}$$

Avec $\hat{Y}_i(k|k) = \left[\hat{y}_i'(k+1|k),\ldots,\hat{y}_i'(k+N_p|k) \right]'$, $U_i(k|k) = \left[u_i'(k|k),\ldots,u_i'(k+N_u-1|k) \right]'$. (22)

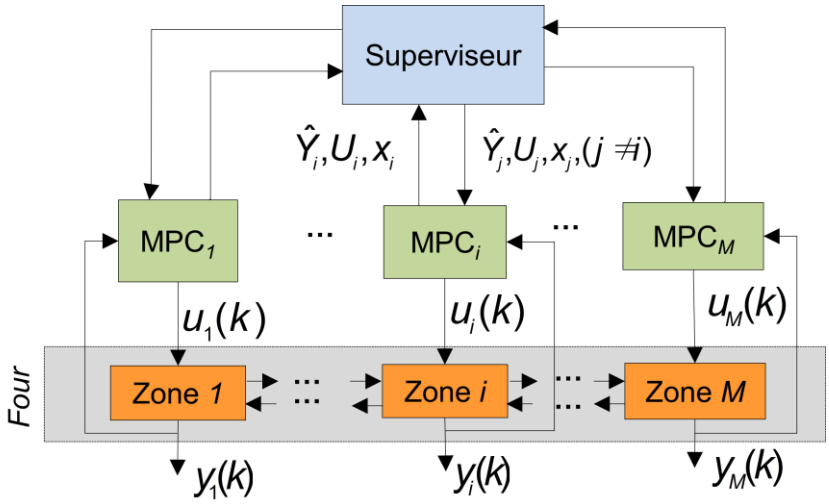

Fig. 13 ; Communication au sein d'un MPC distribué coopératif

Le régulateur de la zone *i* calcule ainsi la commande $U_i(k|k)$ en optimisant non seulement la performance de la zone *i* mais aussi celle des zones voisines *i*+1 et *i*-1. Pour cela, les régulateurs locaux ont besoin de plus de communication entre eux. Un régulateur communique à la fois ses prédictions de température, sa séquence de commande et l'état de sa zone, et reçoit les informations similaires émises par les autres régulateurs (Fig. 13).

Les deux approches précédentes ont été testées sur un système couplé défini par les relations :

$$x_1(k+1) = \begin{pmatrix} 0.15 & 0 & 1 \\ 0 & 0.23 & 0.6 \\ 0 & 0 & 0.4 \end{pmatrix} x_1(k) + \begin{pmatrix} 0.41 \\ 0.53 \\ 0.62 \end{pmatrix} u_1(k) + \begin{pmatrix} 0.03 & 0.01 \\ 0.1 & 0.03 \\ 0.05 & 0.2 \end{pmatrix} \begin{bmatrix} y_2(k) \\ y_3(k) \end{bmatrix}$$

$$y_1(k) = \begin{pmatrix} 0.2 & 0.16 & 0.25 \end{pmatrix} x_1(k)$$

$$x_2(k+1) = \begin{pmatrix} 0.4 & 0.16 \\ 0 & 0.25 \end{pmatrix} x_2(k) + \begin{pmatrix} 0.6 \\ 0.2 \end{pmatrix} u_2(k) + \begin{pmatrix} 0.4 & 0.4 \\ 0.1 & 0.3 \end{pmatrix} \begin{bmatrix} y_1(k) \\ y_3(k) \end{bmatrix}$$

$$y_2(k) = \begin{pmatrix} 0.08 & 0.09 \end{pmatrix} x_2(k)$$

$$x_3(k+1) = \begin{pmatrix} 0.7 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.3 \end{pmatrix} x_3(k) + \begin{pmatrix} 0.16 \\ 0.27 \\ 0.13 \\ 0.31 \end{pmatrix} u_3(k) + \begin{pmatrix} 0.1 & 0.02 \\ 0.3 & 0.09 \\ 0.2 & 0.3 \\ 0.4 & 0.5 \end{pmatrix} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix}$$

$$y_3(k) = \begin{pmatrix} 0.1 & 0.2 & 0.1 & 0.3 \end{pmatrix} x_3(k)$$

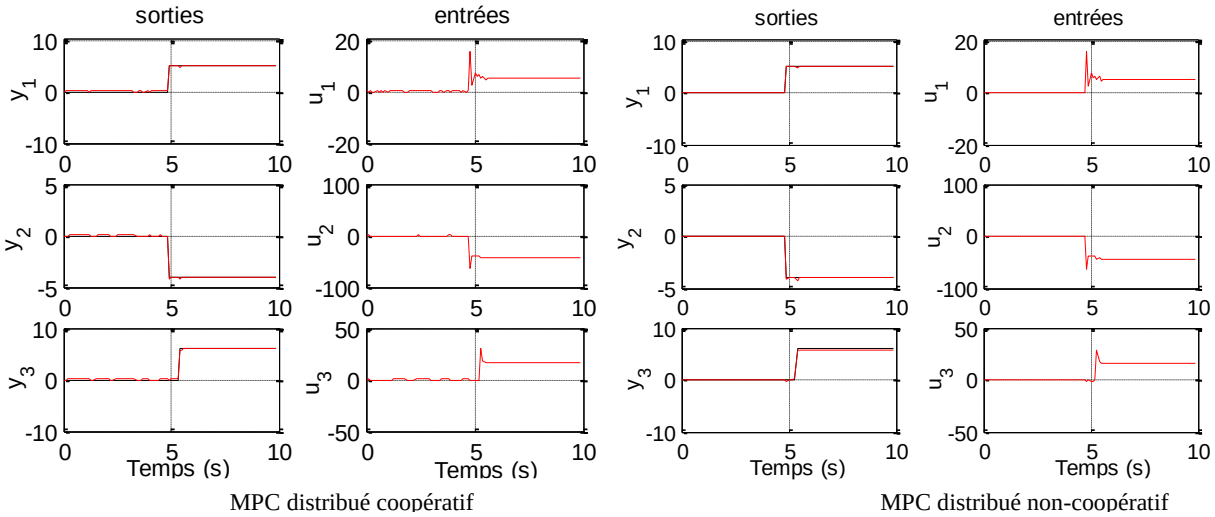

Fig. 14 : Résultats issus de la commande MPC distribuée

Les résultats obtenus par les deux approches MPC distribuées (coopérative et non-coopérative) sont donnés Fig. 14. Nous constatons que les deux approches procurent de bons résultats, en particulier l'algorithme non-coopératif même en présence de couplage. Du fait du couplage peu important entre les zones de réchauffage, la stratégie MPC distribuée non-coopérative a été privilégiée pour la comparaison avec la structure PID et l'implantation sur site. Le train à chaud du site de Florange d'ArcelorMittal possède deux fours de réchauffage ayant la même technologie et chauffant souvent les mêmes types de produits. A titre de comparaison, le MPC distribué non-coopératif a été implanté au sein du niveau 1 du four *A* tandis que les régulateurs de type PID étaient conservés dans le four *B*. La similarité de fonctionnement des deux fours permet de comparer efficacement les comportements obtenus avec le MPC distribué non-coopératif et les régulateurs PID, comme le montre la Fig. 15. On constate alors clairement que la structure MPC procure moins de fluctuations de température et de dépassements de consignes.

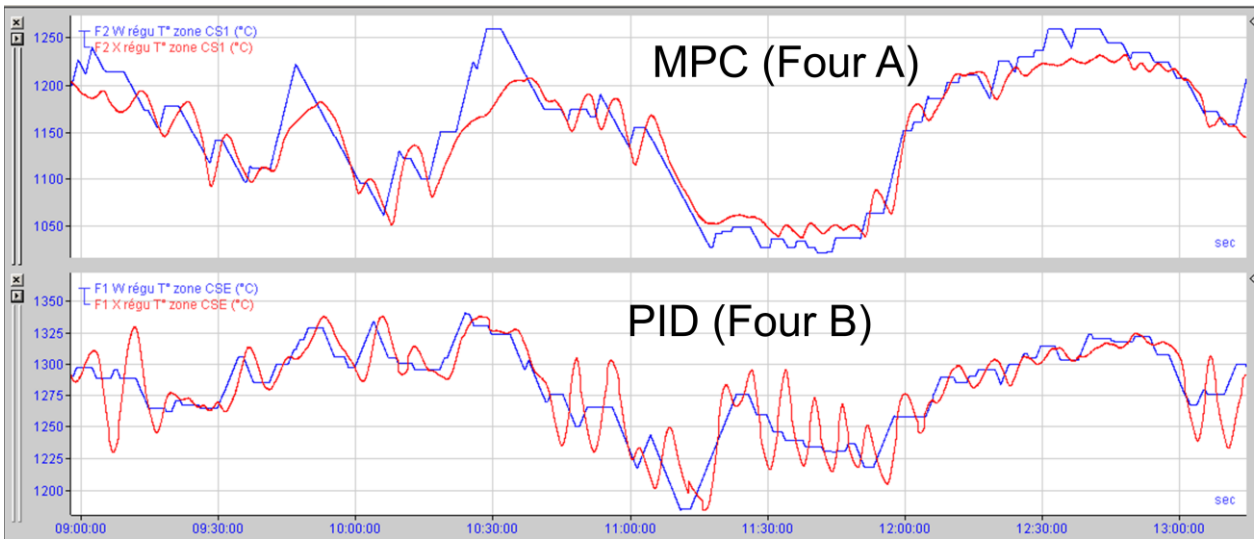

Fig. 15 : Comparaison des comportements MPC distribué et régulateurs PID

Sur l'histogramme de la consommation énergétique des fours montré Fig. 16, nous constatons que la commande MPC distribuée a permis de réduire la consommation du premier semestre de 2014 par rapport à celle du premier semestre de 2013 avec la commande classique PID.

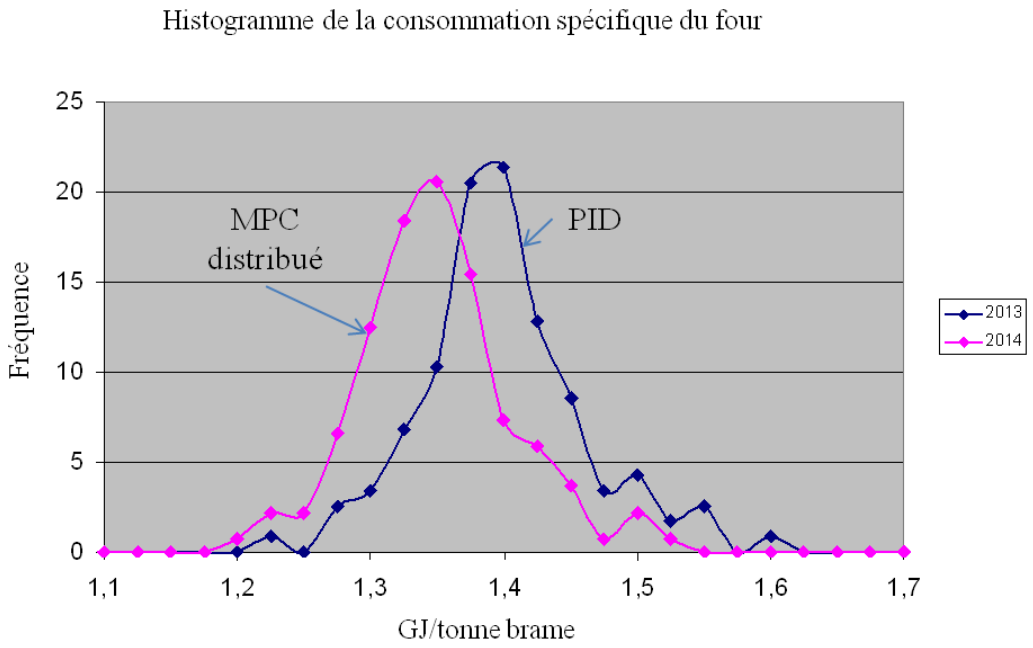

Fig. 16 : Consommation énergétique MPC et PID

D'autres mesures du four sont comparées Fig. 17 et Fig. 18. Le ratio d'oxygène issu de la commande MPC est plus faible et surtout en général moins fluctuant que dans le cas PID. Cela peut permettre de réduire la formation de calamine sur les brames. En termes d'utilisation du mode automatique, les opérateurs du four ont moins besoin d'intervenir en manuel avec la commande MPC, particulièrement dans la zone d'égalisation.

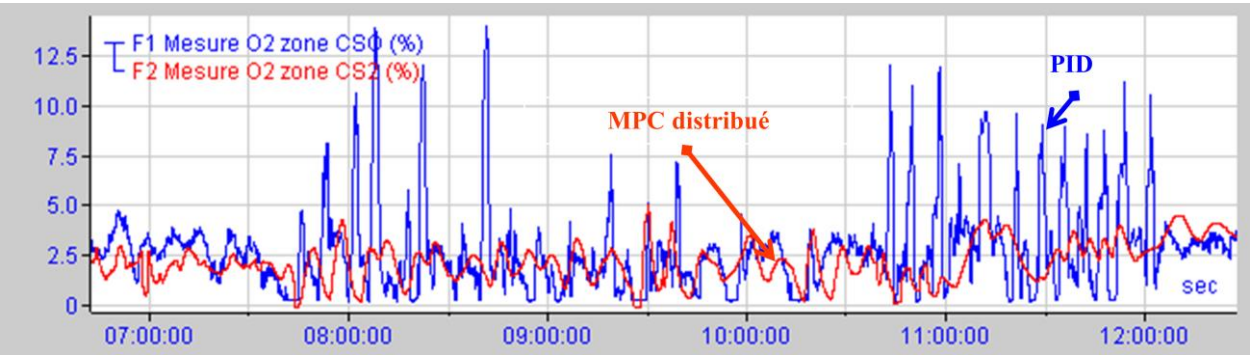

Fig. 17: Ratio d'oxygène dans les fours

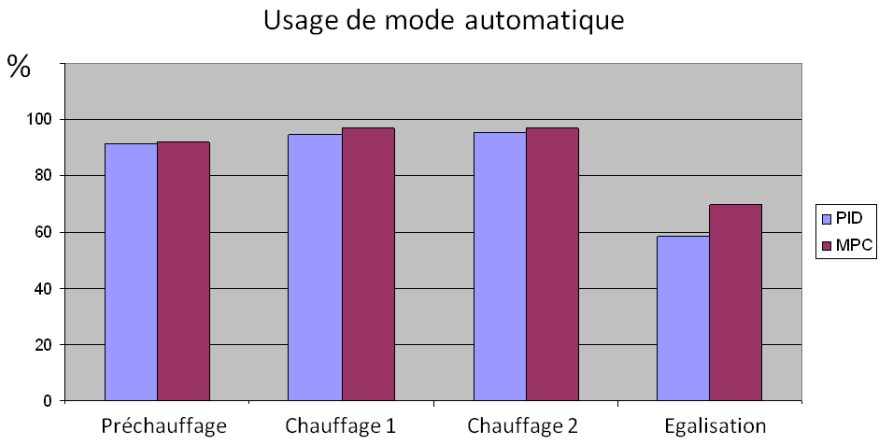

Fig. 18: Taux d'utilisation du mode automatique

3.2. Régulation de la température de brame

Cette partie est dédiée à la commande de niveau 2 dans le cas de cadence prévisionnelle fixe. Les données de cadencement des brames sont envoyées par le niveau 3, cela signifie que le temps de séjour de chaque brame et le contenu du four à un instant donné sont connus à l'avance.

La boucle fermée cascade de niveau 1 et niveau 2 est illustrée Fig. 19. La température visée de la brame *j*, notée $\tilde{T}_{j,end}$, est reçue depuis le niveau 3. La température réalisée de la brame *j*, $T_{j,end}$ est estimée via le modèle de transfert thermique transitoire. Ce modèle utilise les mesures de puissance P_i et celles de température des zones $T_{z,i}$ afin de calculer la température de brames. Le niveau 2 optimise les consignes de température des zones $\tilde{\tilde{T}}_{z,i}$ via un modèle simplifié du four et la stratégie MPC. Ce modèle simplifié est construit sur le même principe que le modèle transitoire, la différence concerne le pas de discrétisation spatiale et échantillonnage temporel. Le MPC de niveau 1 vu précédemment assure le suivi de consigne de température de zone en optimisant la puissance de zones.

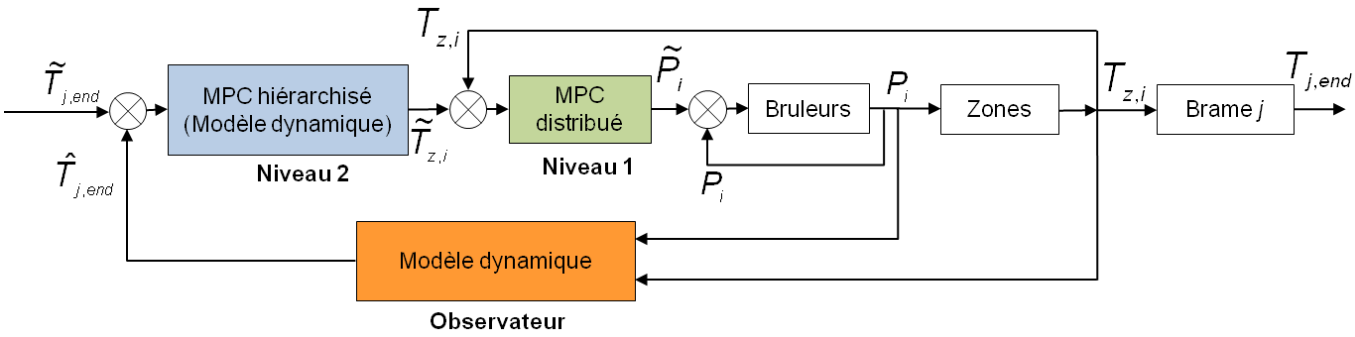

Fig. 19 : Boucle fermée cascade du système de commande niveaux 1 et 2 avec stratégie MPC

Les objectifs de la commande de niveau 2 sont : erreur de température de sortie de brames minimale, inhomogénéité de la température à l'intérieur de la brame minimale, perte de matière issue de la formation de calamine minimale, et décarburation ayant une influence directe sur la qualité du produit minimale. Les contraintes de la commande sont liées aux limites des températures de zones, aux puissances maximales de zones et à la température des brames. Les objectifs et contraintes sur la température des brames sont illustrés Fig. 20, Fig. 21 et Fig. 22.

La Fig. 20 montre que la température de brames est contrainte par $T_{j,max}$ pour éviter les pertes d'énergie et de matière. La température de défournement des brames est contrainte dans un intervalle $[T_{j,end,min}, T_{j,end,max}]$, voir Fig. 21. En effet, cette marge est imposée pour prendre en compte l'erreur issue de la régulation de température. Plus la commande est précise, plus cette marge peut être réduite. Le gradient de température est aussi objet de contraintes comme l'illustre la Fig. 22. La limite supérieure du gradient $\Delta T_{j,max}$ est imposée pour éviter la déformation de la brame. A la sortie du four, les brames doivent avoir une bonne homogénéité définie par $\Delta T_{j,end,max}$ pour être acceptées dans les laminoirs.

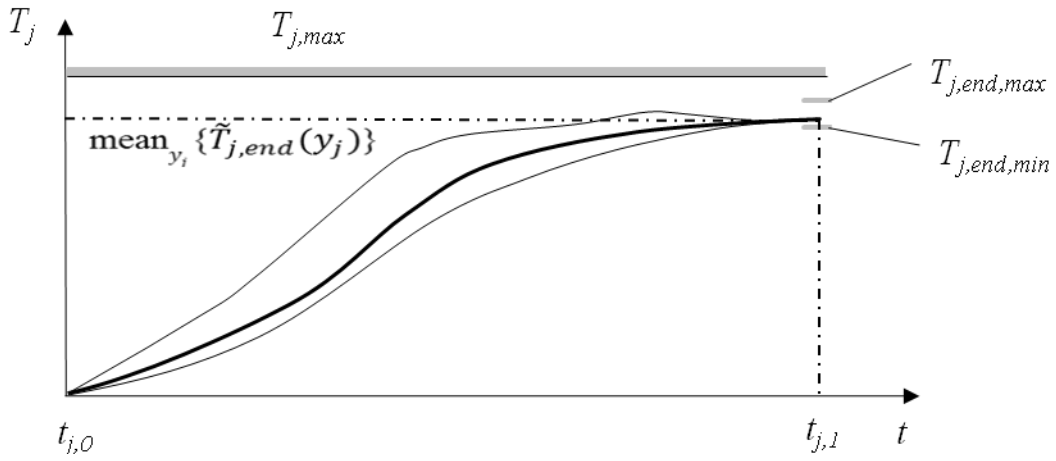

Fig. 20 : Contraintes sur la température de brames

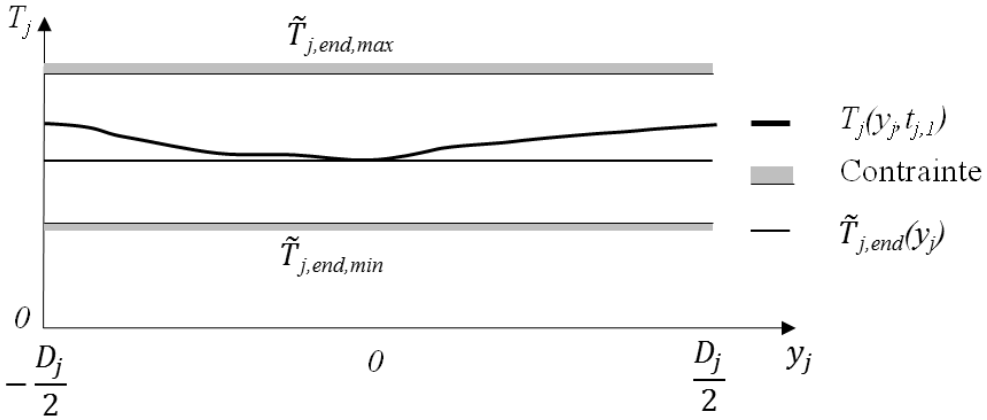

Fig. 21 : Contraintes sur le profil de température finale des brames

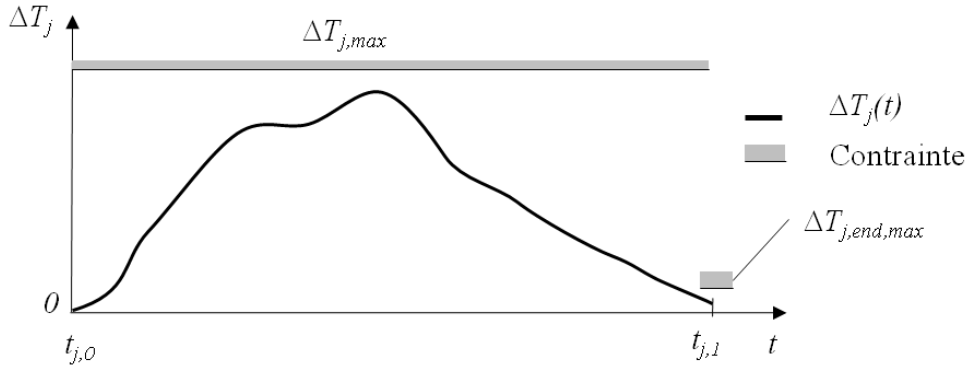

Fig. 22 : Contraintes sur le gradient de température des brames

Comme indiqué, les variables de commande sont les consignes de température :

$$u(t) = \left[u_1(t), ..., u_M(t)\right]^T = \left[\tilde{T}_{z,1}(t), ..., \tilde{T}_{z,M}(t)\right]^T \tag{23}$$

Ces variables et leurs vitesses d'évolution sont limités par des bornes supérieures et inférieures qui varient potentiellement dans le temps. En revanche, dans l'étude, ces bornes sont constantes et données dans le tableau Tab. 1.

$$\begin{aligned}\tilde{T}_{z,i,\min}(t) \le \tilde{T}_{z,i}(t) \le \tilde{T}_{z,i,\max}(t)\\ \dot{\tilde{T}}_{z,i,\min}(t) \le \dot{\tilde{T}}_{z,i}(t) \le \dot{\tilde{T}}_{z,i,\max}(t)\end{aligned} \quad (24)$$

Variable	Valeur nominale
$\tilde{T}_{z,\min}(t)$	[830, 1085, 1200, 1210] (°C)
$\tilde{T}_{z,\max}(t)$	[1100, 1250, 1355, 1300] (°C)
$\dot{\tilde{T}}_{z,\min}(t)$	-4.5 (°C/min)
$\dot{\tilde{T}}_{z,\max}(t)$	9 (°C/min)

Tab. 1 : Contraintes portant sur les consignes de température de zones

Les contraintes sur la température de brame sont exprimées comme suit :

$$T_j(y_j,t) \le T_{j,\max}, \ \ \forall y_j \in \left[-D_j/2, D_j/2\right], \ \forall t \in \left[t_{j,0}, t_{j,1}\right]. \quad (25)$$

$$T_{j,end,\min} \le T_j(y_j,t_{j,1}) \le T_{j,end,\max}, \ \ \forall y_j \in \left[-D_j/2, D_j/2\right] \quad (26)$$

$$\begin{aligned}&\Delta T_j(t) \le \Delta T_{j,\max} \ \ \forall t \in \left[t_{j,0}, t_{j,1}\right],\\ &\Delta T_j(t_{j,1}) \le \Delta T_{j,end,\max}.\end{aligned} \quad (27)$$

où $t_{j,0}$ et $t_{j,1}$ sont les temps d'entrée et de sortie de la brame j ; le gradient de température est déterminé par :

$$\Delta T_j(t) = \max_{y_j}\{T_j(y_j,t)\} - \min_{y_j}\{T_j(y_j,t)\}, \ \ \forall y_j \in \left[-D_j/2, D_j/2\right], \ \forall t \in \left[t_{j,0}, t_{j,1}\right]. \quad (28)$$

Ces contraintes de température de brames sont récapitulées dans le Tab. 2 ci-dessous.

Variables	**Description**	**Valeurs nominales**
$T_{j,\max}$	Borne supérieure de température de défournement de brame	1350 °C
$\tilde{T}_{j,end,\max} - \tilde{T}_{j,end}$	Marge supérieure de précision d'erreur de température finale de brame	20 °C
$\tilde{T}_{j,end,\max} - \tilde{T}_{j,end}$	Marge inférieure de précision d'erreur de température finale de brame	20 °C
$\Delta T_{j,\max}$	Inhomogénéité maximale de température de brame	350 °C
$\Delta T_{j,end,\max}$	Inhomogénéité maximale de température de défournement de brame	100 °C

Tab. 2 : Contraintes portant sur la température de brames

Le problème de la régulation de la température de brames peut être formalisé comme le problème d'asservissement de la température de défournement moyenne (et de son gradient) définie par :

$$T_s(t)=\begin{cases}\underset{y_j}{moyen}\left\{T_j(y_j,t_{j,1})\right\},\ \text{Si } \exists\, j\in S, t=t_{j,1}\\ 0,\ \text{autrement}\end{cases}$$
$$g_s(t)=\begin{cases}\Delta T_j(t_{j,1}),\ \text{Si } \exists\, j\in S, t=t_{j,1}\\ 0,\ \text{autrement}\end{cases} \tag{29}$$

où S est l'ensemble des brames à chauffer par le four. Les consignes de température à suivre et contraintes associées sont alors définies par :

$$\tilde{T}_s(t)=\begin{cases}\tilde{T}_{j,end},\ \text{Si } \exists\, j\in S, t=t_{j,1}\\ 0,\ \text{autrement}\end{cases}$$
$$\tilde{T}_{s,\min}(t)=\begin{cases}\tilde{T}_{j,end,\min},\ \text{Si } \exists\, j\in S, t=t_{j,1}\\ 0,\ \text{autrement}\end{cases} \tag{30}$$
$$\tilde{T}_{s,\max}(t)=\begin{cases}\tilde{T}_{j,end,\max},\ \text{Si } \exists\, j\in S, t=t_{j,1}\\ 0,\ \text{autrement}\end{cases}$$

$$g_{\max}(t)=\begin{cases}\Delta T_{j,end,\max},\ \text{Si } \exists\, j\in S, t=t_{j,1}\\ 0,\ \text{autrement}\end{cases} \tag{31}$$

Pour les contraintes de température de brames, celles-ci s'expriment en fonction des variables d'intérêt x_1, x_2 :

$$x_1(t)=\left[\max_{y_{j+1}}\left\{T_{j+1}(y_{j+1},t)\right\},\max_{y_{j+2}}\left\{T_{j+2}(y_{j+2},t)\right\},...,\max_{y_{j+N_{in}}}\left\{T_{j+N_{in}}(y_{j+N_{in}},t)\right\}\right]^T$$
$$x_2(t)=\left[\Delta T_{j+1}(y_{j+1},t),\Delta T_{j+2}(y_{j+2},t),...,\Delta T_{j+N_{in}}(y_{j+N_{in}},t)\right]^T,\ \ y_j\in\left[-D_j/2,D_j/2\right], \tag{32}$$

Où $[j+1,\ j+2,\ldots,\ j+N_{in}]$ est l'ensemble des indices des brames dans le four à l'instant t. Le problème d'optimisation MPC sur l'horizon de prédiction Δt_P, l'horizon de contrôle Δt_C est défini par la minimisation du critère suivant :

$$\min_{U(\tau)}\left\{J_0\left(U(\tau)\right)=\alpha_1 J_1\left(U(\tau)\right)+\alpha_2 J_2\left(U(\tau)\right)+\alpha_3 J_3\left(U(\tau)\right)+\alpha_4 J_4\left(U(\tau)\right)+\alpha_5 J_5\left(U(\tau)\right)\right\} \tag{33}$$

sous les contraintes des températures de zones, des température de sortie, des gradients de température de sortie définies comme suit :

$$\begin{aligned}&\tilde{T}_{z,i,\min}(t)\le u_i(t)\le\tilde{T}_{z,i,\max}(t),\forall i=1,...,M,t\in\left[\tau,\tau+\Delta t_p\right]\\ &\dot{\tilde{T}}_{z,i,\min}(t)\le u_i(t)\le\dot{\tilde{T}}_{z,i,\max}(t),\forall i=1,...,M,t\in\left[\tau,\tau+\Delta t_p\right]\\ &T_{j,end,\min}\le T_s(t)\le T_{j,end,\max},\ \text{Si } t=t_{j,1},t\in\left[\tau,\tau+\Delta t_p\right]\\ &g(t)\le\Delta T_{j,end,\max}\ \text{Si } t=t_{j,1},t\in\left[\tau,\tau+\Delta t_p\right]\\ &x_1(t)\le T_{\max}(t),\forall t\in\left[\tau,\tau+\Delta t_p\right]\\ &x_2(t)\le\Delta T_{\max}(t),\forall t\in\left[\tau,\tau+\Delta t_p\right]\end{aligned} \tag{34}$$

où S_1 représente l'ensemble des brames défournées pendant la période $[\tau, \tau + \Delta t_P]$. Les termes J_1, J_2 de l'équation (33) représentent les objectifs sur la température de défournement et son homogénéité. Ces termes sont exprimés par les relations :

$$J_1\left(U(\tau)\right) = \sum_{j \in S(\tau)} w_1\left(z_j(\tau)\right) \times \left| T_s(t_{j,1}) - \tilde{T}_s(t_{j,1}) \right|, \;\; t_{j,1} \in \left[\tau, \tau + \Delta t_p \right] \tag{35}$$

$$J_2\left(U(\tau)\right) = \sum_{j \in S(\tau)} w_2\left(z_j(\tau)\right) \times g(t_{j,1}), \;\; t_{j,1} \in \left[\tau, \tau + \Delta t_p \right] \tag{36}$$

où $w_1(z_j)$ et $w_2(z_j)$ sont des coefficients de pondération de la brame j, dépendant de la position z_j de la brame. De même les autres termes du critère sont donnés par les relations :

$$J_3\left(U(\tau)\right) = \int_{\tau}^{\tau + \Delta t_p} w_3(t) \sum_{i=1}^{M} P_i(t) dt \tag{37}$$

$$J_4\left(U(\tau)\right) = \int_{\tau}^{\tau + \Delta t_p} w_4(t) \sum_{i=1}^{M} u_i(t) dt \tag{38}$$

$$J_5\left(U(\tau)\right) = \int_{\tau}^{\tau + \Delta t_p} \left[x_1^T(t) \times w_5(t) \times x_1(t) \right] dt \tag{39}$$

Les contraintes (34) peuvent être exprimées sous forme de pénalités :

$$J_{c,1}(\tau) = \int_{\tau}^{\tau + \Delta t_c} \sum_{i=1}^{M} \left[\max\left\{0, u_i(t) - \tilde{T}_{z,i,\max}(t)\right\} + \left| \min\left\{0, u_i(t) - \tilde{T}_{z,i,\min}(t)\right\} \right| \right] dt \tag{40}$$

$$J_{c,2}(\tau) = \int_{\tau}^{\tau + \Delta t_c} \sum_{i=1}^{M} \left[\max\left\{0, \dot{u}_i(t) - \dot{\tilde{T}}_{z,i,\max}(t)\right\} + \left| \min\left\{0, \dot{u}_i(t) - \dot{\tilde{T}}_{z,i,\min}(t)\right\} \right| \right] dt \tag{41}$$

$$J_{c,3}(\tau) = \int_{\tau}^{\tau + \Delta t_p} \max\left\{0, g(t) - g_{\max}(t)\right\} dt = \sum_{j \in S(\tau)} \max\left\{0, g(t_{j,1}) - \Delta T_{j,end,\max}\right\} \tag{42}$$

$$\begin{aligned} J_{c,4}(\tau) &= \int_{\tau}^{\tau + \Delta t_p} \left[\max\left\{0, T_s(t) - \tilde{T}_{s,\max}(t)\right\} + \left| \min\left\{0, T_s(t) - \tilde{T}_{s,\min}(t)\right\} \right| \right] dt \\ &= \sum_{j \in S(\tau)} \left[\max\left\{0, T_s(t_{j,1}) - \tilde{T}_{s,\max}(t_{j,1})\right\} + \left| \min\left\{0, T_s(t_{j,1}) - \tilde{T}_{s,\min}(t_{j,1})\right\} \right| \right] \end{aligned} \tag{43}$$

$$\begin{aligned} J_{c,5}(\tau) &= \int_{\tau}^{\tau + \Delta t_p} \max\left\{0, \left[x_1(t) - T_{\max}(t)\right]^T \left[x_1(t) - T_{\max}(t)\right]\right\} dt \\ J_{c,6}(\tau) &= \int_{\tau}^{\tau + \Delta t_p} \max\left\{0, \left[x_2(t) - \Delta T_{\max}(t)\right]^T \left[x_2(t) - \Delta T_{\max}(t)\right]\right\} dt \end{aligned} \tag{44}$$

Par conséquent, la minimisation de (33) sous les contraintes (34) devient une minimisation sans contrainte :

$$\min_{U(\tau)} \left\{ J_0\left(U(\tau)\right) = \sum_{i=1}^{5} \alpha_i \times J_i\left(U(\tau)\right) + \mu \sum_{i=1}^{6} J_{c,i}\left(U(\tau)\right) \right\} \tag{45}$$

La discrétisation temporelle est alors réalisée par substitution aux variables continues des variables discrètes récapitulées dans le Tab. 3.

Nom de paramètre	Continu	Discret
Instant de temps d'enfournement	$t_{j,0}$	$k_{j,0}$
Instant de temps de défournement	$t_{j,1}$	$k_{j,1}$
Instant de temps	τ	k
Période d'échantillonnage	$\Delta\tau_s$	Δt_s
Horizon de prédiction	Δt_p	N_p
Horizon de commande	Δt_c	N_c

Tab. 3 : Variables discrètes et continues correspondantes

On en déduit alors la forme discrète des termes de la fonction de coût comme suit :

$$J_1\left(U(k)\right)=\sum_{j\in S(k)} w_1\left(z_j(k)\right)\times\left|T_s(k_{j,1})-\tilde{T}_s(k_{j,1})\right|,\ \ \mathrm{k}_{j,1}\in\left[k,k+N_p\right] \tag{46}$$

$$J_2\left(U(k)\right)=\sum_{j\in S(k)} w_2\left(z_j(k)\right)\times g(k_{j,1}),\ \ k_{j,1}\in\left[k,k+N_p\right] \tag{47}$$

$$J_3\left(U(k)\right)=\sum_{j=1}^{N_p} w_3(k+j)\sum_{i=1}^{M} P_i(k+j) \tag{48}$$

$$J_4\left(U(k)\right)=\sum_{j=1}^{N_c} w_4(k+j)\sum_{i=1}^{M} u_i(k+j) \tag{49}$$

$$J_5\left(U(k)\right)=\sum_{i=1}^{N_p} x_1^T(k+i)\times w_5(k+i)\times x_1(k+i) \tag{50}$$

$$J_{c,1}(k)=\sum_{i=1}^{N_c}\sum_{j=1}^{M}\left[\max\left\{0,u_j(k+i)-\tilde{T}_{z,j,\max}(k+i)\right\}+\left|\min\left\{0,u_j(k+i)-\tilde{T}_{z,j,\min}(k+i)\right\}\right|\right] \tag{51}$$

$$J_{c,2}(k)=\sum_{i=1}^{N_c}\sum_{j=1}^{M}\left[\max\left\{0,u_j(k+i)-\dot{\tilde{T}}_{z,j,\max}(k+i)\right\}+\left|\min\left\{0,u_j(k+i)-\dot{\tilde{T}}_{z,j,\min}(k+i)\right\}\right|\right] \tag{52}$$

$$J_{c,3}(k)=\sum_{j\in S(k)}\max\left\{0,g(k_{j,1})-\Delta T_{j,end,\max}\right\} \tag{53}$$

$$J_{c,4}(k)=\sum_{j\in S(k)}\left[\max\left\{0,T_s(k_{j,1})-\tilde{T}_{s,\max}(k_{j,1})\right\}+\left|\min\left\{0,T_s(k_{j,1})-\tilde{T}_{s,\min}(k_{j,1})\right\}\right|\right] \tag{54}$$

$$\begin{aligned} J_{c,5}(k)&=\sum_{i=1}^{N_p}\max\left\{0,\left[x_1(k+i)-T_{\max}(k+i)\right]^T\left[x_1(k+i)-T_{\max}(k+i)\right]\right\}\\ J_{c,6}(\tau)&=\sum_{i=1}^{N_p}\max\left\{0,\left[x_2(k+i)-\Delta T_{\max}(k+i)\right]^T\left[x_2(k+i)-\Delta T_{\max}(k+i)\right]\right\}\end{aligned} \tag{55}$$

De sorte que la solution recherchée sera issue de la minimisation du critère suivant sous forme discrète :

$$\min_{U(k)}\left\{J_0\left(U(k)\right)=\sum_{i=1}^{5}\alpha_i\times J_i\left(U(k)\right)+\mu\sum_{i=1}^{6}J_{c,i}\left(U(k)\right)\right\} \tag{56}$$

Le coefficient associé aux termes de pénalités, μ, est significativement plus grand que les autres coefficients. Les coefficients α_1, α_2 associés aux objectifs sur la température de défournement des brames sont plus forts (car objectifs plus prioritaires) que ceux liés aux objectifs de consommation. Nous avons donc la relation entre les coefficients :

$$\mu \gg \alpha_1, \alpha_2 > \alpha_3, \alpha_4, \alpha_5 \tag{57}$$

L'horizon de prédiction doit être choisi de sorte que l'optimisation prenne en compte toutes les brames situées dans les zones actives du four. Le temps de séjour d'une brame dans les zones actives représente environ 2/3 du temps de séjour total de la brame dans le four, par conséquent nous avons la contrainte sur l'horizon de prédiction sur la sortie :

$$\Delta t_p \geq \frac{2}{3}\left(t_{j,1} - t_{j,0}\right) \tag{58}$$

Le temps d'échantillonnage du niveau 2 correspond à l'intervalle temporel de défournement des brames qui est de l'ordre de 5 min.

La résolution de problème de minimisation sans contrainte (54) est fondée sur la simulation du four comme le montre la Fig. 16. La simulation de four est réalisée via un programme numérique complexe, par conséquent ne permettant pas d'obtenir une forme analytique de la fonction de cout (54) ni de ses dérivées. Il est alors intéressant d'utiliser l'algorithme d'optimisation du simplex Nelder-Mead, qui a uniquement besoins de la valeur de la fonction de coût. Cet algorithme est connu pour sa faible charge de calcul tout en fournissant une solution satisfaisante dans les applications industrielles.

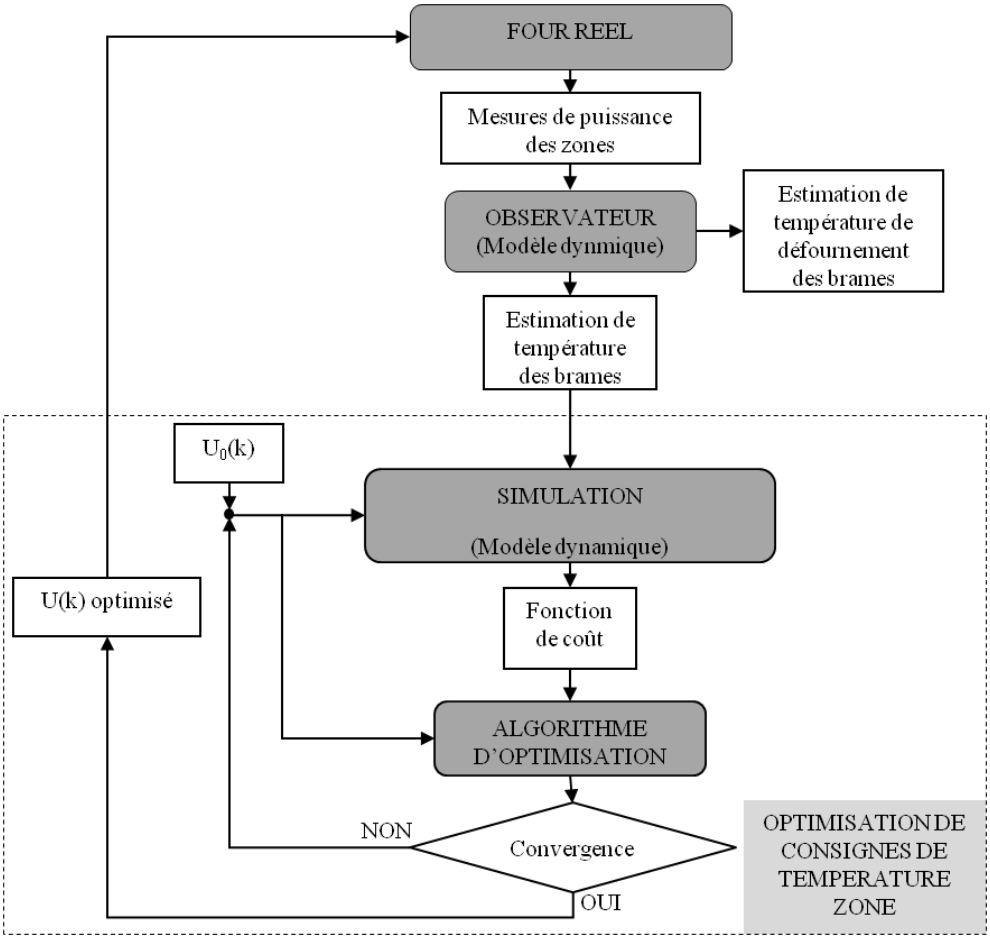

Fig. 23 : Optimisation de la stratégie MPC fondée sur la simulation du four

La simulation de tout cet ensemble pour la validation de la commande (Fig. 24) est effectuée en Fortran, utilisant le simulateur du four fondé sur le modèle transitoire de four. Dans le scénario de test, des données de trois jours du four sont utilisées ; environ 900 brames sont chauffées durant cette période de trois jours. Les consignes de température réelles appliquées dans les fours sont réutilisées en parallèle avec celles données par le régulateur MPC. Les résultats sont ensuite comparés en termes de température finale de brames, de gradient de température, et de consommation du four.

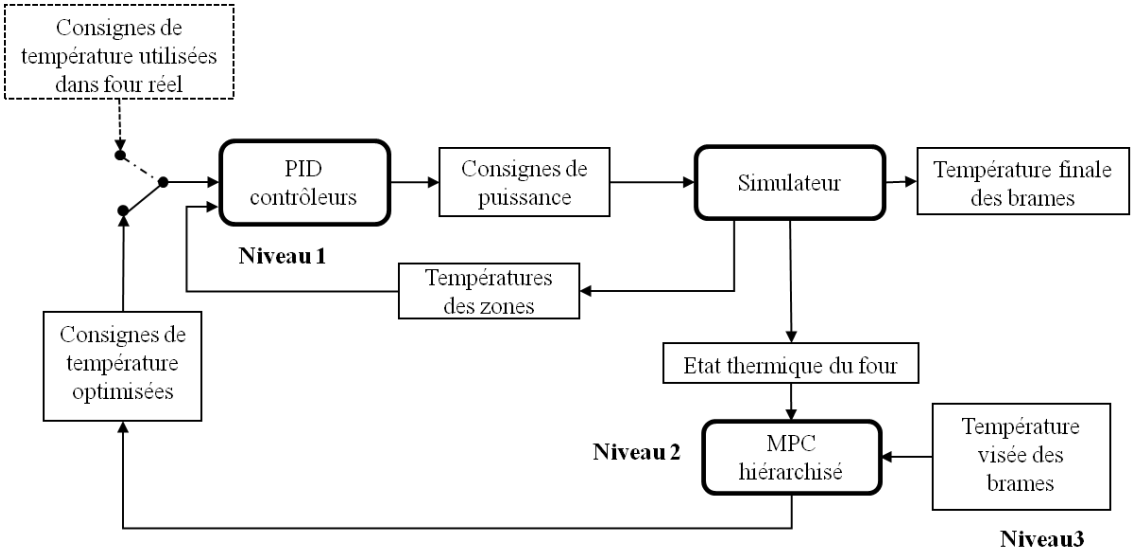

Fig. 24 : Schéma structurel de la simulation (regroupant les trois niveaux hiérarchisés)

Les Fig. 25 et Fig. 26 illustrent les épaisseurs, les largeurs et les longueurs des brames sur ces trois jours. On constate que l'épaisseur des brames ne varie pas significativement (229 mm à 235 mm), le four étant conçu spécialement pour des brames avec une certaine épaisseur. En revanche, la longueur et la largeur des brames changent de façon non négligeable d'une brame à l'autre, par exemple, de 1 m jusqu'à 2 m en largeur, et de 6 m jusqu'à 10 m en longueur.

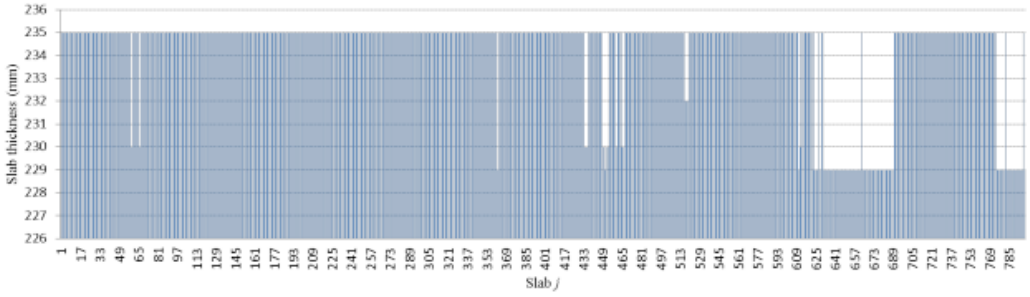

Fig. 25 : Epaisseur de brames

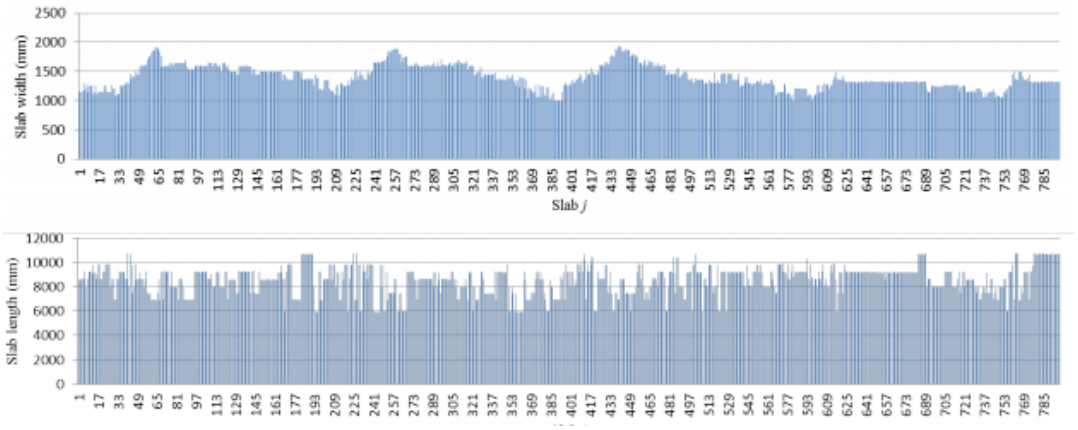

Fig. 26 : Largeur et longueur de brames

Le cheminement des brames dans le four est illustré Fig. 27 et Fig. 28. On constate que pour les brames 596 à 606 le temps de séjour est prolongé à cause d'un arrêt de 2 heures. De fortes variations de température visée des brames peuvent être constatées Fig. 29.

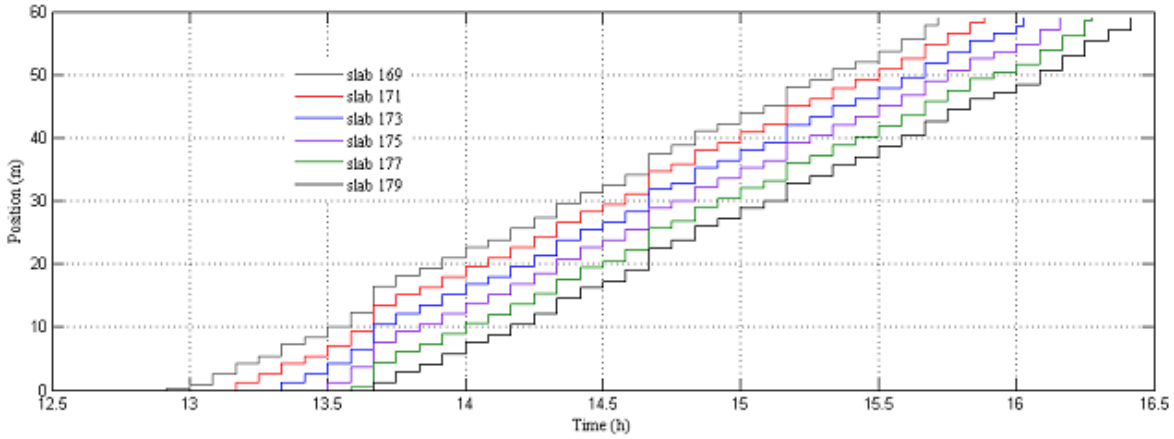

Fig. 27 : Diagramme position-temps des brames (169-179)

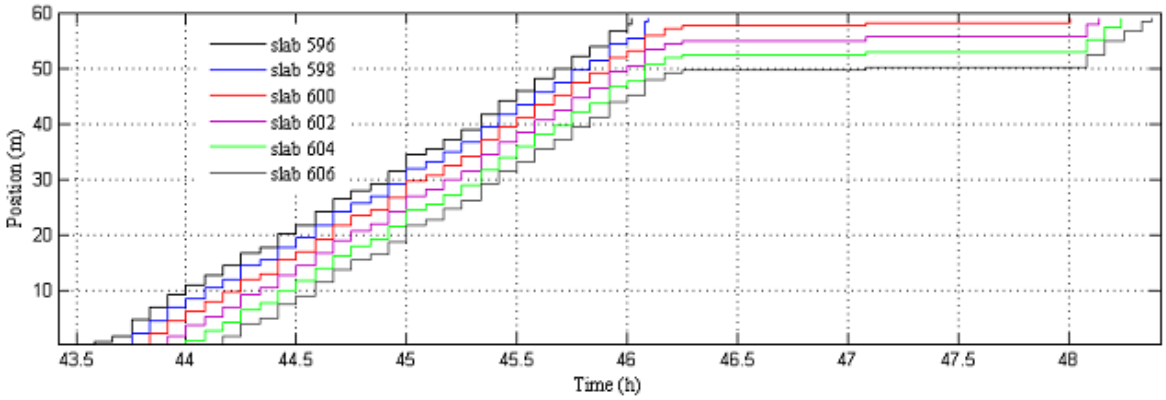

Fig. 28: Diagramme position-temps des brames (596-606)

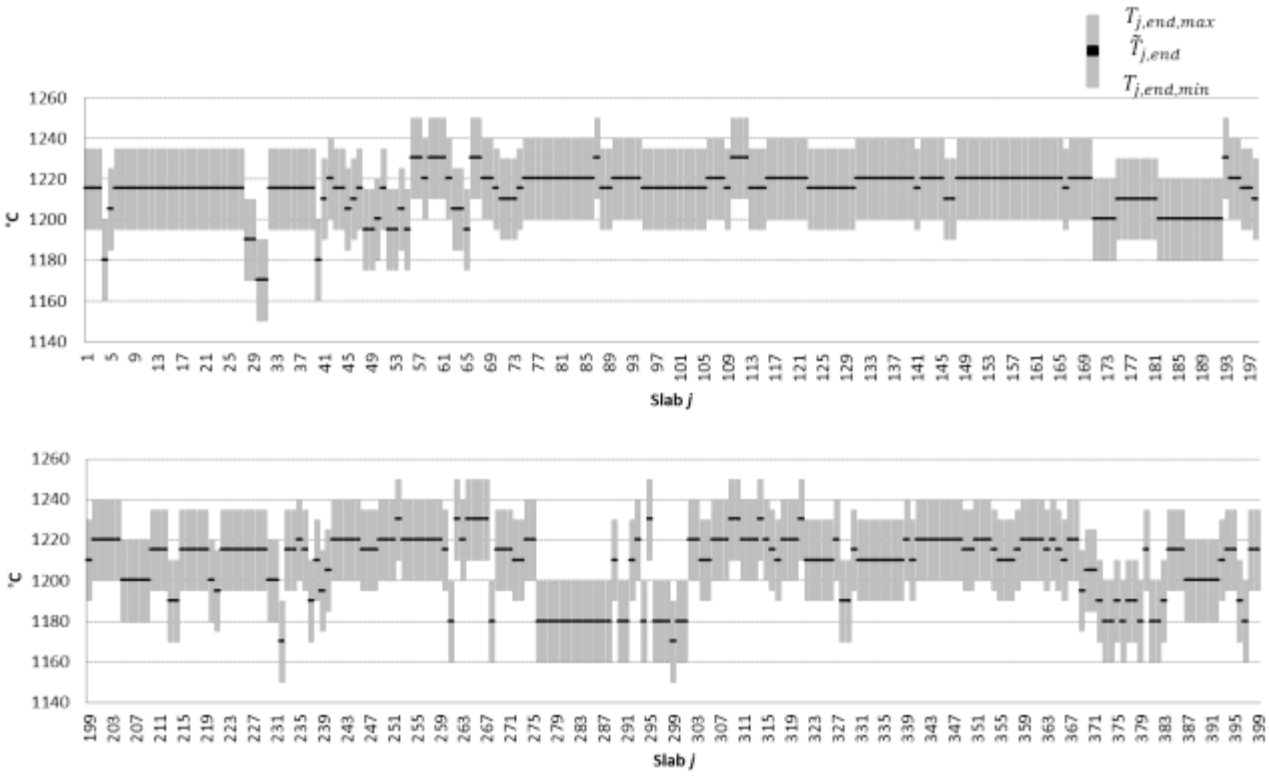

Fig. 29 : Température visée des brames (1-399)

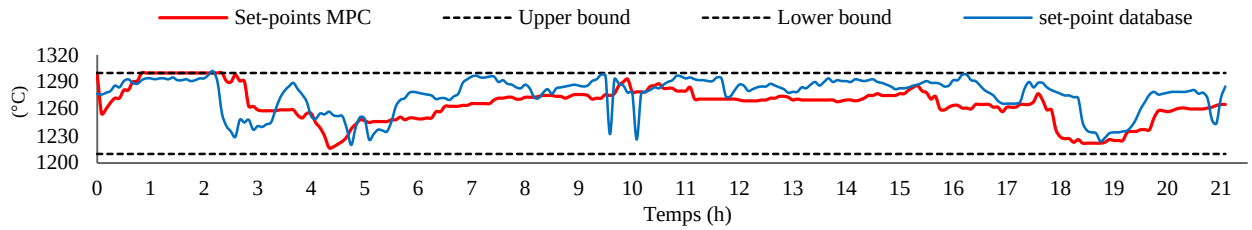

Fig. 30 : Consignes de température de la zone d'égalisation

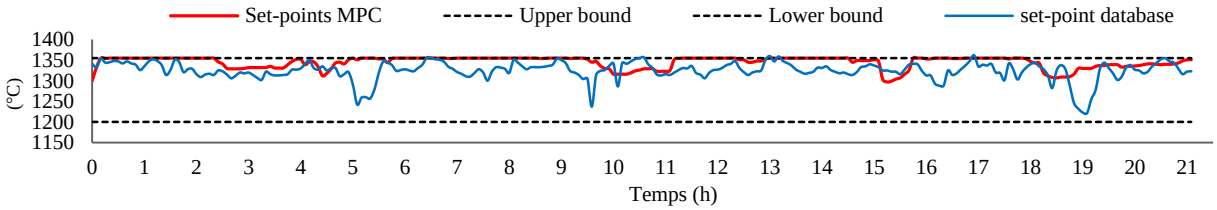

Fig. 31 : Consignes de température de la zone de chauffe 2

Les consignes de température de la zone d'égalisation sont données sur la Fig. 30. On constate que les consignes calculées par la structure MPC sont inférieures à celles fournies par le régulateur existant. En revanche, concernant les consignes des zones de chauffe 2 et de chauffe 1 des Fig. 31 et Fig. 32, on remarque que les consignes données par la loi MPC sont en général supérieures à celles données par la commande existante. En zone de préchauffage, les consignes de température données par la loi MPC Fig. 33 sont nettement inférieures à celles données par la commande actuelle. Les valeurs moyennes des consignes sont récapitulées dans le Tab. 4.

	Moyenne des consignes de température des zones (°C)			
	Préchauffage	**Chauffage 1**	**Chauffage 2**	**Egalisation**
Contrôleur MPC	851	1196	1340	1262
Contrôleur existant	1028	1211	1321	1270

Tab. 4 : Moyennes des consignes de température de zone

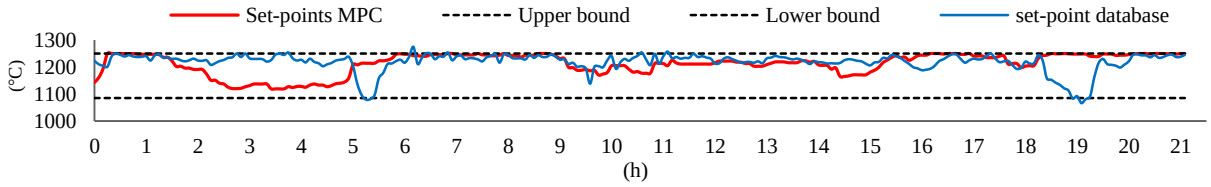

Fig. 32 : Consignes de température de la zone de chauffe 1

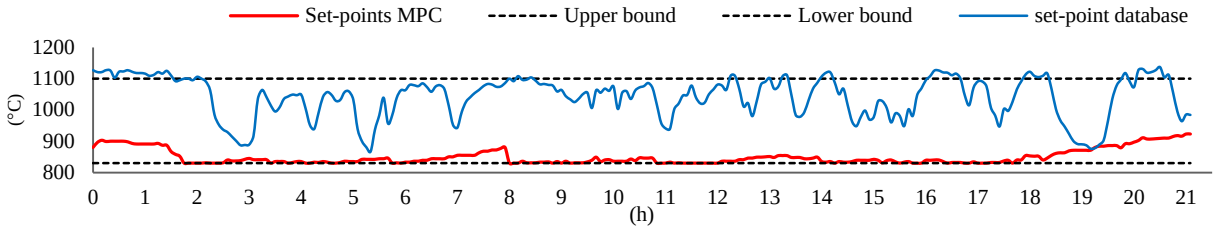

Fig. 33 : Consignes de température de zone préchauffe

Les températures de défournement des brames sont comparées Fig. 34. En général, la structure MPC donne des températures de brame plus précises. Ceci est confirmé par le Tab. 5, pour lequel la loi MPC, obtient pour 92% des brames la température dans la marge des limites, tandis que le régulateur existant n'en obtient que 75%.

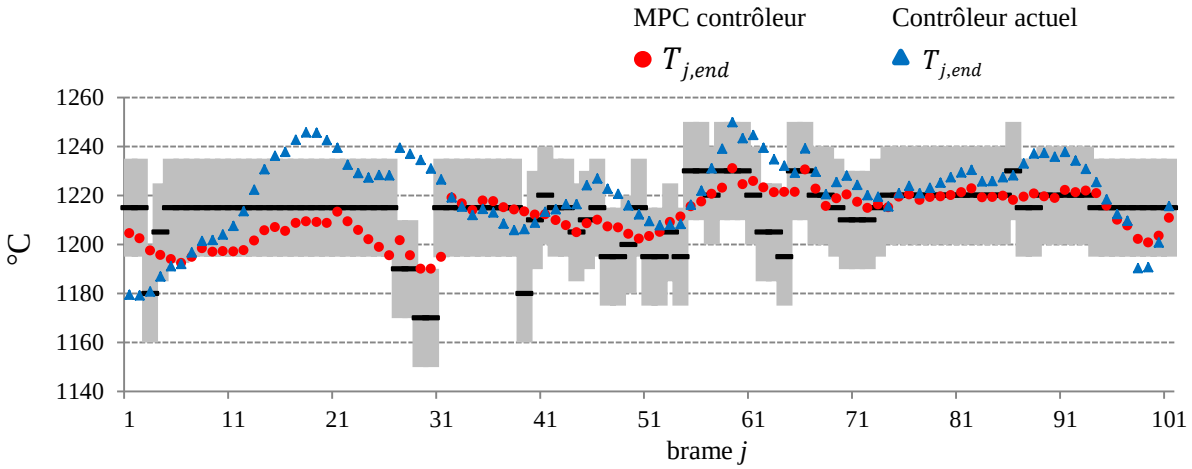

Fig. 34 : Température de défournement des brames

	Moyenne (°C)	Ecart type (°C)	Marges de précision respectées (%)
MPC	7.4	8	92
Existant	13.7	10	75

Tab. 5 : Comparaison des températures de défournement des brames

La Fig. 35 fournit l'histogramme de l'erreur de la température de défournement des brames par rapport à la température visée. Le nombre de brames défournées ayant une erreur de moins de 5°C a augmenté de 10%. La surchauffe est réduite, ce qui peut avoir un effet positif sur la consommation du four.

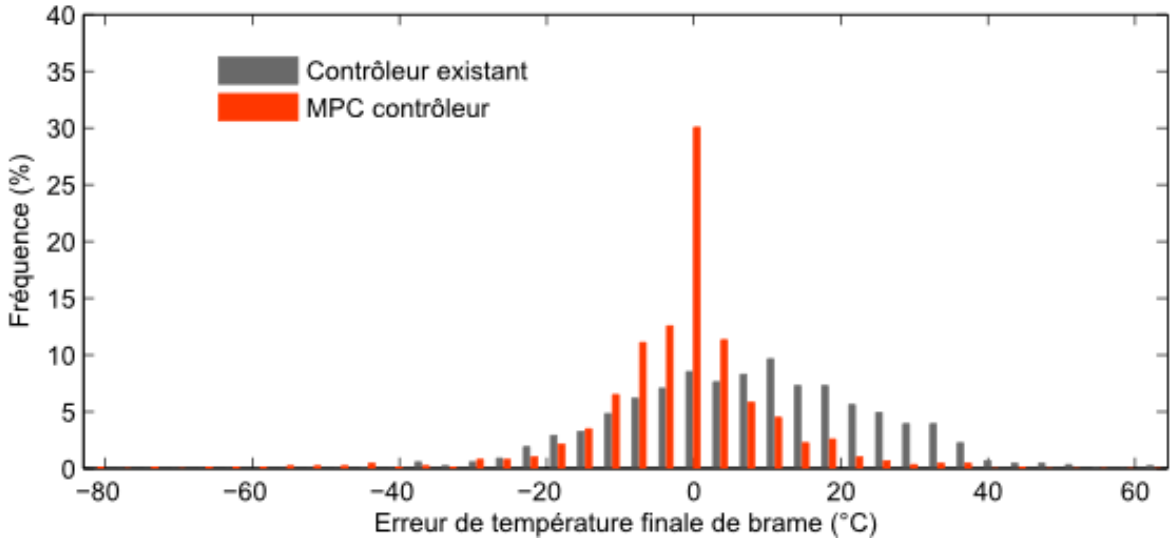

Fig. 35 : Histogramme de l'erreur de la température de défournement des brames

En termes d'homogénéité de la température de défournement des brames, nous pouvons constater sur la Fig. 36 que la structure MPC induit des gradients plus faibles. En moyenne, les gradients de température des brames sont réduits de 5°C comme le montre le Tab. 6.

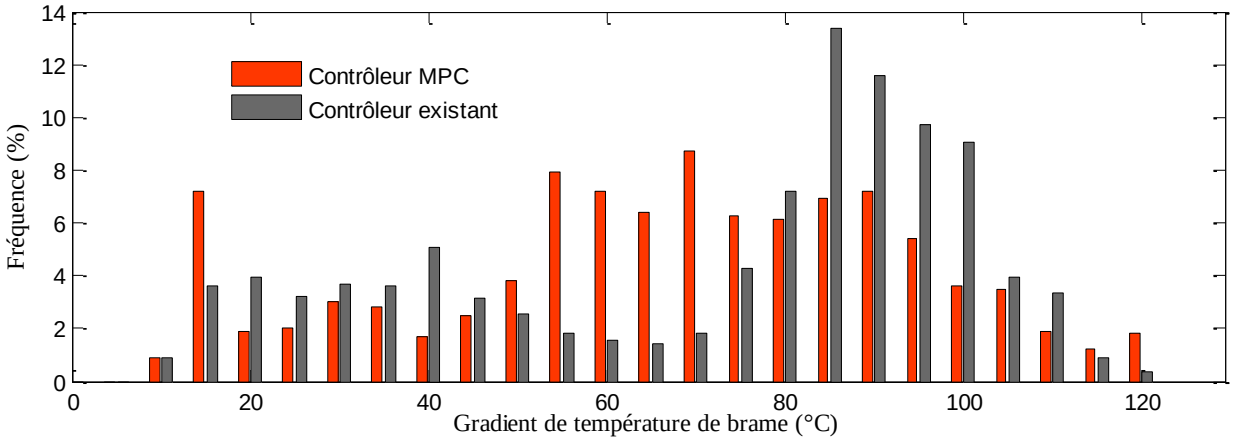

Fig. 36 : Histogramme du gradient de température de défournement des brames

	Moyenne (°C)	**Ecart type (°C)**	**< 100 °C (%)**
MPC	68	26	93
Existant	73	26	90

Tab. 6 : Comparaison des gradients de température de défournement des brames

Les Fig. 37 et Fig. 38 reproduisent les trajectoires des températures de brames. Le régulateur MPC donne en général des trajectoires qui se situent en dessous de celles données par la commande actuelle, particulièrement pour la brame 400. La brame 600 est affectée par un arrêt du four. Avec la commande actuelle, la température défournée de cette brame est bien plus chaude que la valeur visée. Le régulateur arrive à réduire l'effet de l'arrêt et donne la bonne température défournée de la brame.

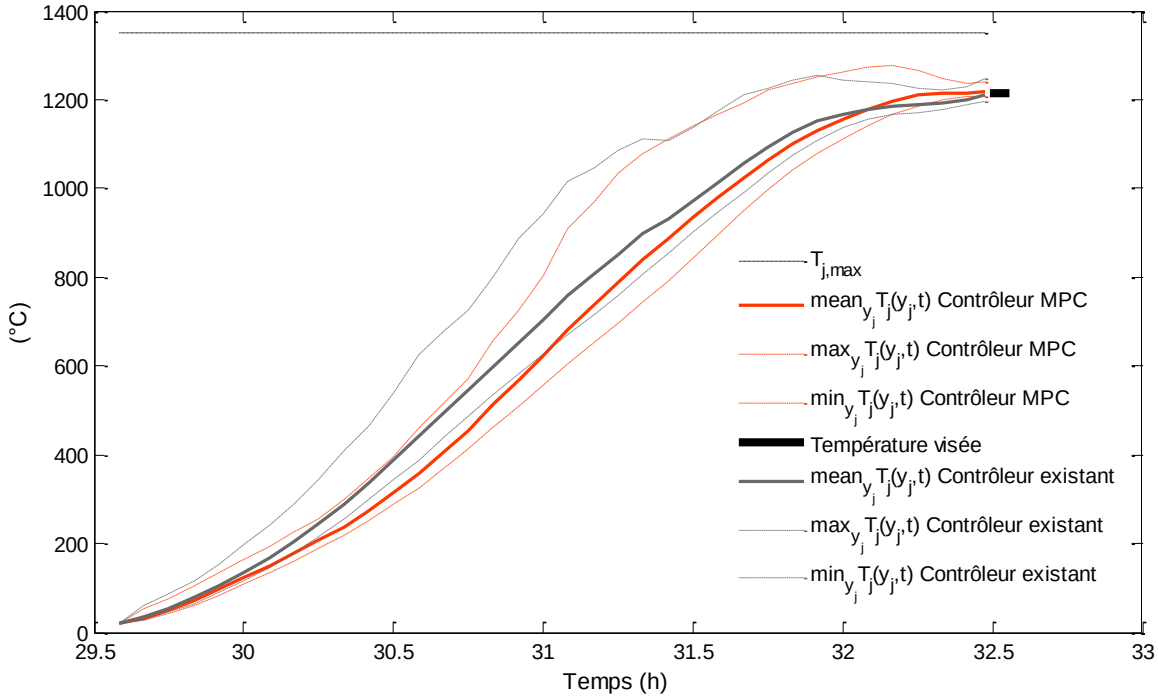

Fig.37 : Trajectoire de la température de brame *j*=400

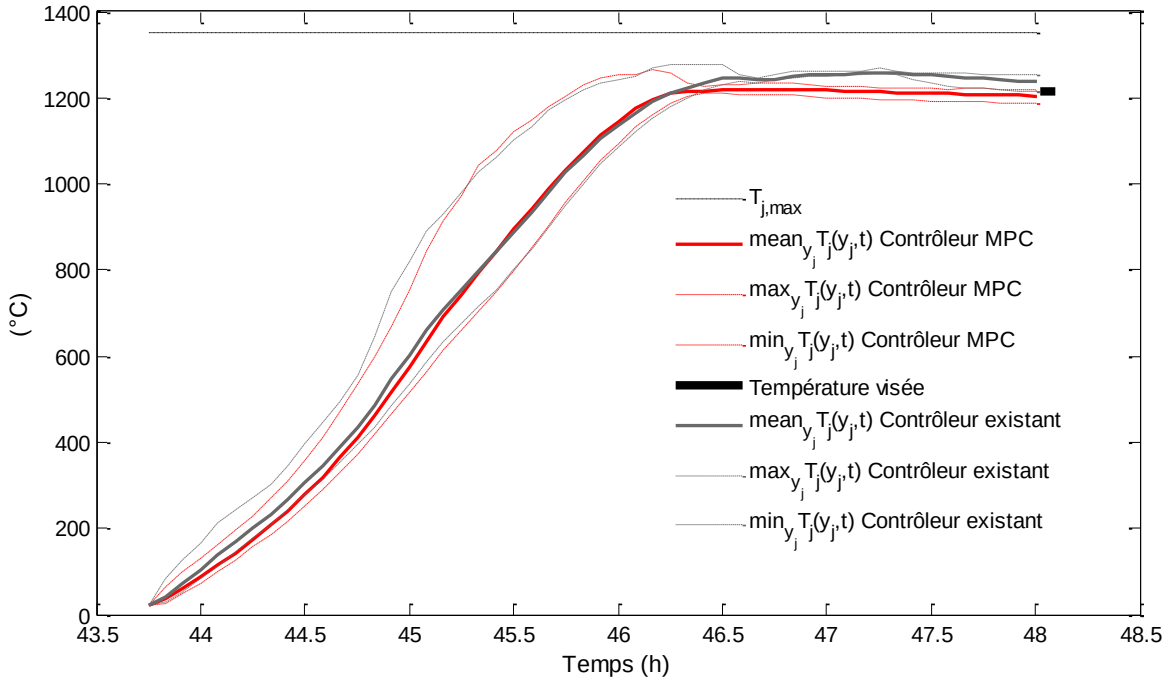

Fig. 38 : Trajectoire de la température de brame *j*=600

Les Fig. 39 et Fig. 40 montrent la trajectoire des gradients des brames. Les gradients respectent leurs bornes supérieures. On remarque le gradient monte moins vite avec le régulateur MPC qu'avec la commande existante. Cela peut avoir une influence positive sur la consommation du four puisque la brame est alors chauffée de façon plus importante uniquement vers la sortie du four.

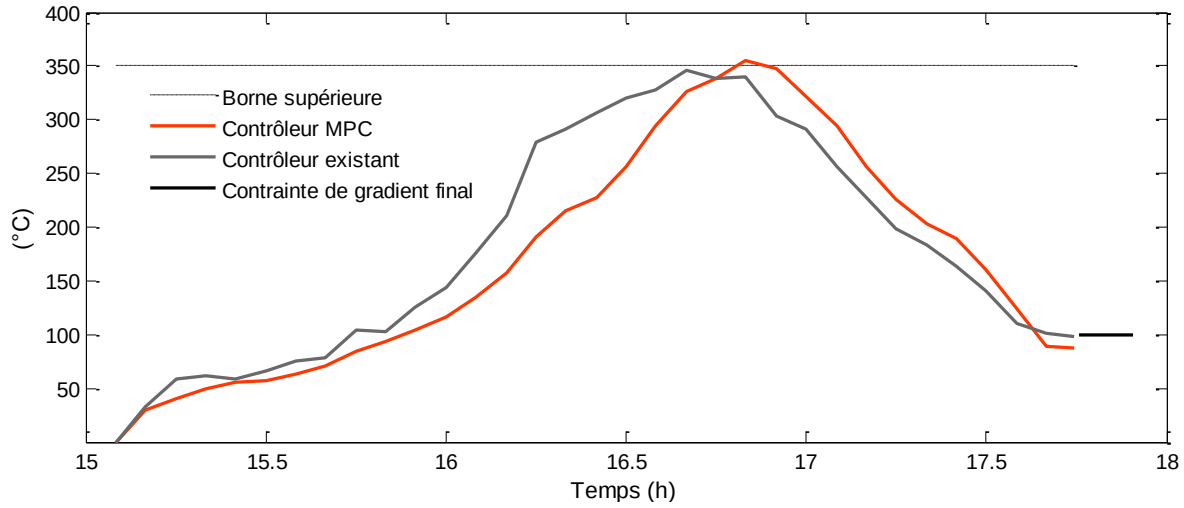

Fig. 39 : Gradient de température de la brame *j=400*

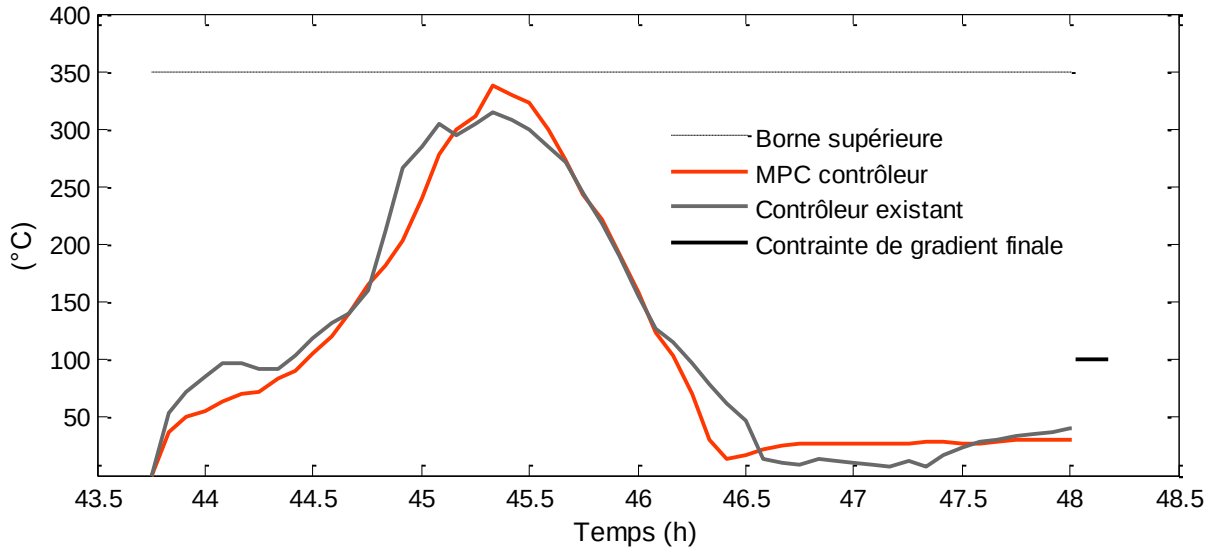

Fig. 40 : Gradient de température de la brame *j=600*

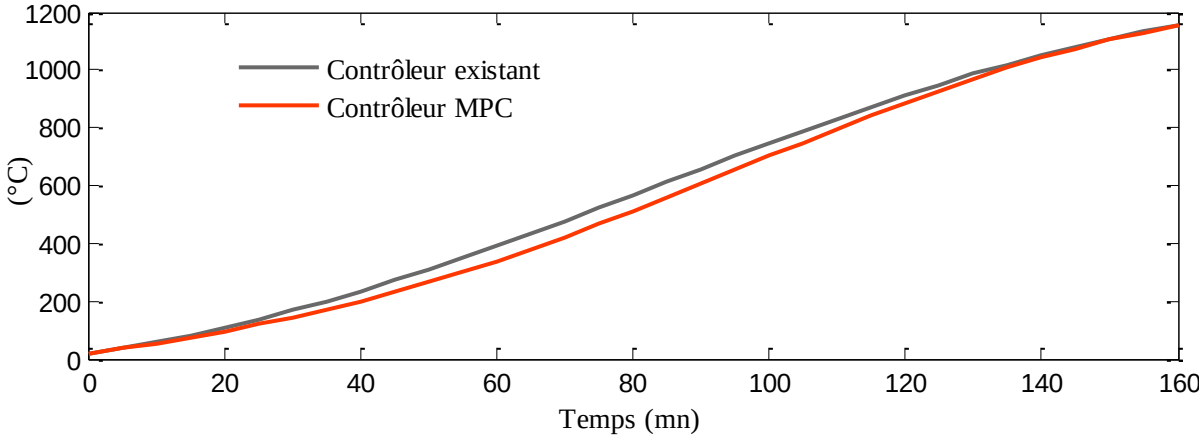

Fig. 41 : Trajectoire moyenne de la température de brames

Les trajectoires moyennes de toutes les brames chauffées sont montrées sur les Fig. 41 et Fig. 42. On constate bien de nouveau que le régulateur MPC a tendance à chauffer un peu plus tard que la commande actuelle. De ce fait, la consommation devrait être réduite.

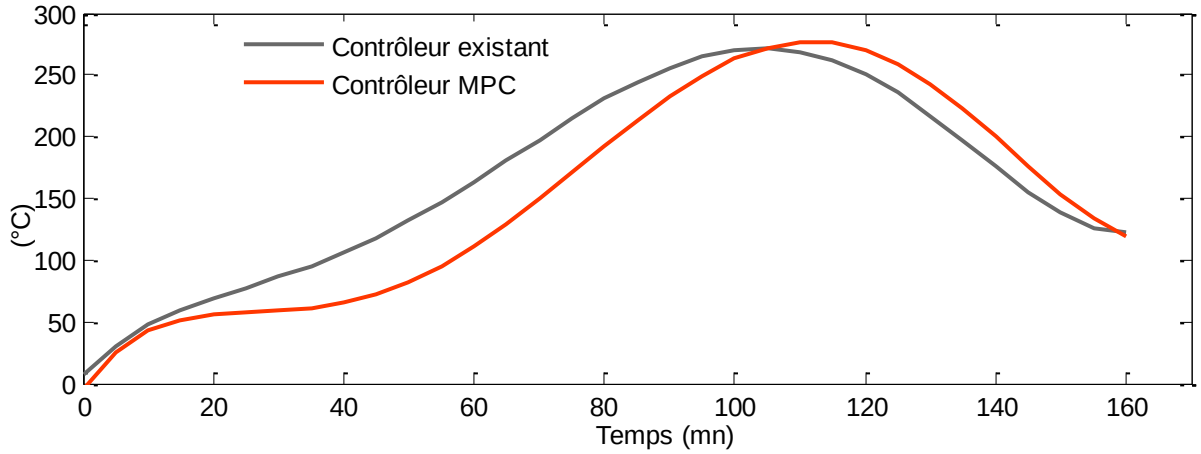

Fig. 42 : Trajectoire moyenne du gradient de température de brames

En termes de consommation d'énergie, nous pouvons constater sur la Fig. 43 que la consommation d'énergie des zones de préchauffage et d'égalisation est diminuée alors que celle de deux zones de chauffage principales est augmentée. La consommation totale du four est montrée Fig. 44. Nous voyons que la structure MPC consomme en général moins que la commande actuelle. Le Tab. 7. récapitule les réductions obtenues via le MPC par rapport à la commande existante. La structure MPC induit en moyenne une diminution de 5% de l'énergie consommée.

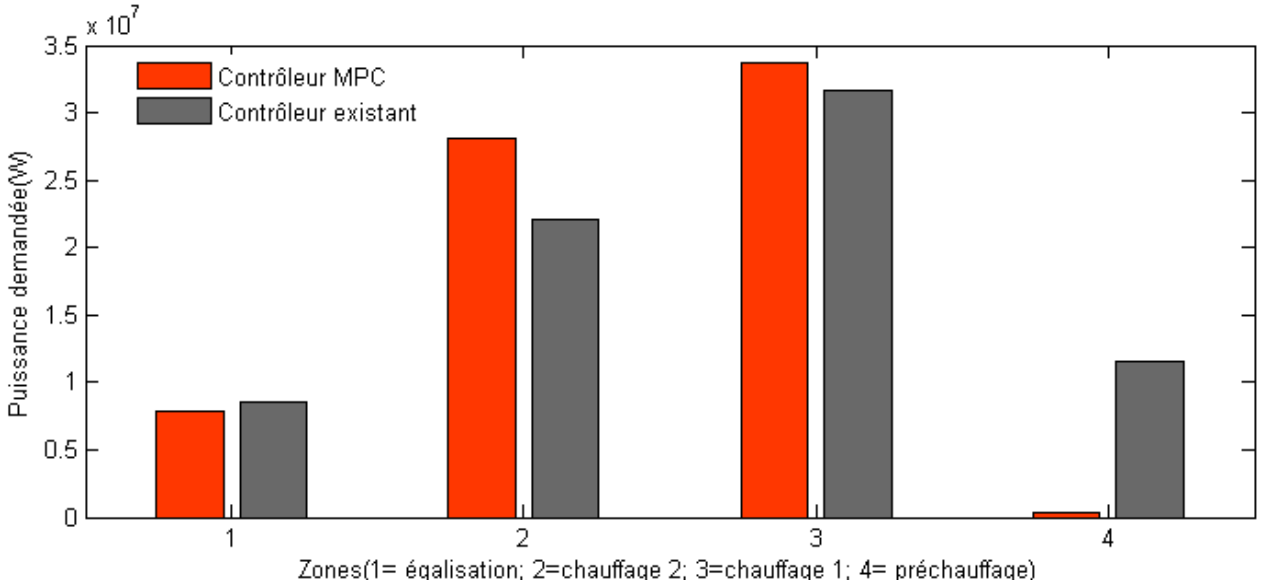

Fig. 43 : Comparaison de la consommation d'énergie des zones

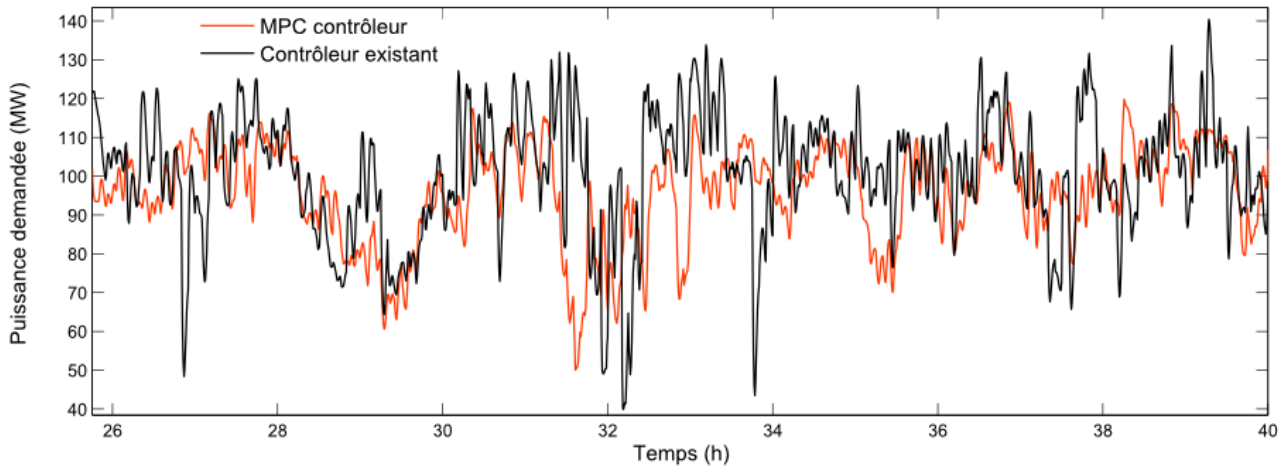

Fig. 44 : Consommation d'énergie du four

	MPC	**Contrôleur existant**	**Réduction (%)**
Jour 1	0.9852	1.0444	5.5
Jour 2	1.0148	1.0667	4.5
Jour 3	0.9111	0.963	5.6
Moyenne	0.9704	1.0222	5.2

Tab. 7 : Réduction d'énergie induite par la stratégie MPC

3.3. Augmentation de la productivité

La régulation de la température de la brame peut être effectuée par la régulation de la température proprement dite ou par la régulation du cadencement des brames, ou par un mixte des deux solutions. Dans la section précédente, nous avons mis l'accent sur le problème de l'optimisation de la consigne de température de zone du niveau 2 et de la régulation de la température de ces zones au niveau 1, avec un cadencement fixe des brames. Or le cadencement des brames est connu à l'avance, via une table fournissant l'intervalle de temps entre les défournements consécutifs des brames comme le montre la Fig. 45.

Identifiant de brame	**Intervalle temporel entre défournements (min)**	**Intervalle temporel entre enfournements (min)**	**Température initiale (°C)**	**Température visée (°C)**
14P144	5	4	20	1215
14P145	4	2	20	1215
14P146	4	4	20	1215

Fig. 45 : Cadencement fixe des brames

Par exemple, considérons deux brames consécutives labellisées *14P144* et *14P145*. Le cadencement fixe impose que la brame *14P145* soit chargée dans le four 2 minutes après la brame *14P144*. De même, la brame *14P145* sera évacuée du four 4 minutes après la brame *14P144*. Dans la pratique, lorsque les laminoirs à chaud ne constituent pas un bouchon de la chaîne de production, la priorité du four de réchauffage est de maximiser sa productivité, et de toute évidence, dans le même temps, d'assurer l'exigence de qualité de chauffage. Par conséquent, nous allons reprendre la solution précédemment construite pour l'optimisation des consignes de température et l'adapter au problème de maximisation de la productivité.

Les variables de commande sont maintenant l'intervalle moyen de défournement et les consignes de températures de zones :

$$u(k) = \left[\delta t_{sch}(k), T_{z,1}(k), T_{z,2}(k), ..., T_{z,M}(k) \right] \tag{57}$$

Le problème de minimisation est réécrit comme suit :

$$\min_{U(k)}\left\{J_0\left(U(k)\right)=\sum_{i=1}^{6}\alpha_i\times J_i\left(U(k)\right)+\mu\sum_{i=1}^{7}J_{c,i}\left(U(k)\right)\right\} \tag{58}$$

où les termes $J_1,\ldots,J_5$ et $J_{c,1},\ldots,J_{c,6}$ sont ceux définis dans la section précédente. J_6 représente l'objectif visant à réduire l'intervalle de temps entre défournements et est déterminé par :

$$J_6\left(U(k)\right)=\sum_{i=1}^{N_c}w_6(k+i)\times u_1(k+i) \tag{59}$$

Le terme de pénalité $J_{c,7}$ représente la contrainte de limite inférieure de l'intervalle de temps $\delta t_{sch,min}$ qui définit la capacité physique du laminoir à chaud :

$$J_{c,7}\left(U(k)\right)=\sum_{i=1}^{N_c}\left|\min\left\{0,\delta t_{sch}(k+i)-\delta t_{sch,\min}\right\}\right| \tag{60}$$

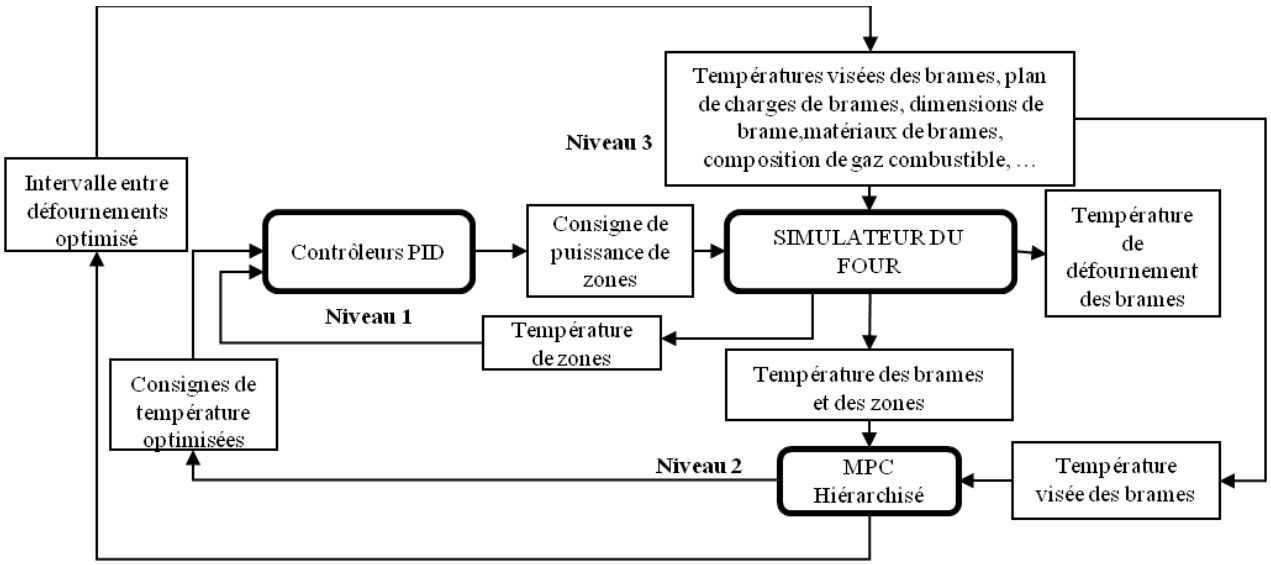

Fig. 46 : Simulation globale avec optimisation de la cadence

La Fig. 46 montre le schéma de la simulation incluant la structure MPC hiérarchisée et l'optimisation de la cadence du four. Le régulateur MPC niveau 2 optimise non seulement les consignes de température mais aussi l'intervalle de temps entre les défournements des brames. Cette dernière variable est ensuite envoyée au simulateur du four pour changer le cadencement des brames.

La Fig. 47 montre que le régulateur MPC donne alors généralement des intervalles de temps entre défournements plus petits, ce qui correspond à une augmentation de la productivité du four comme le montre la Fig. 48.

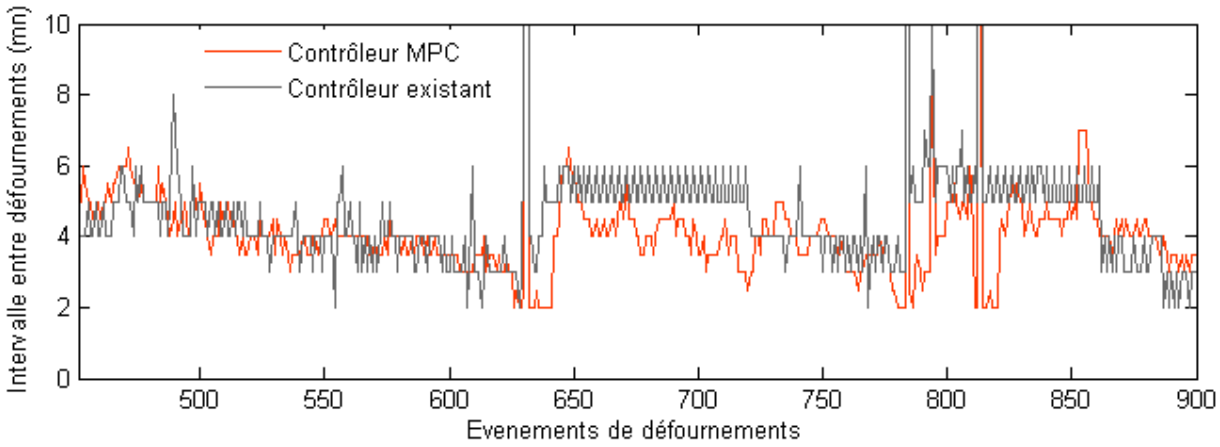

Fig. 47 : Intervalle de défournement des brames

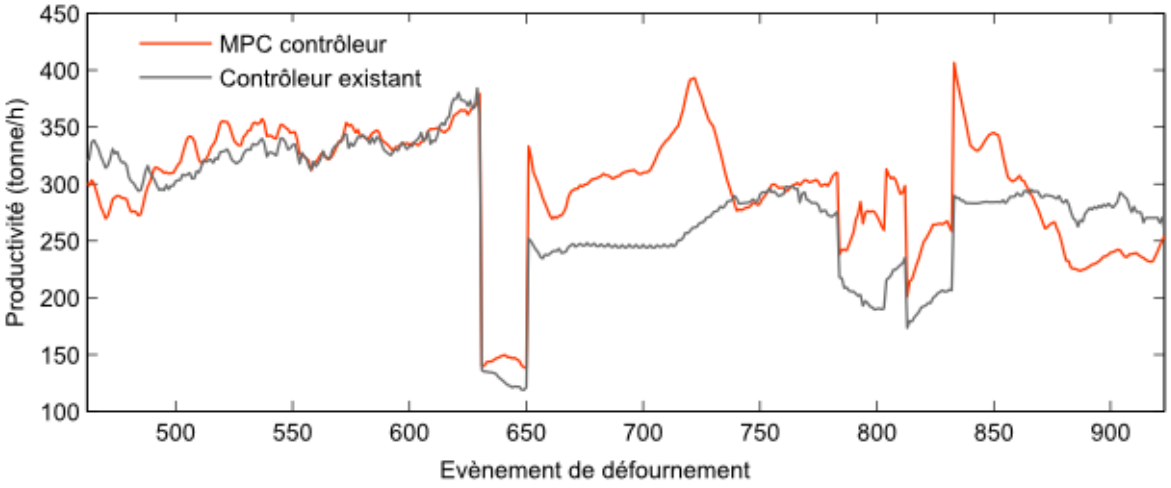

Fig. 48 : Productivité du four

Les histogrammes du gradient et de l'erreur de température défournée des brames sont donnés Fig. 49 et Fig. 50. On constate que la stratégie MPC fournit une meilleure précision de température défournée, et en général un gradient plus faible. On remarque enfin Tab. 8 que la productivité moyenne du four obtenue par la loi MPC est augmentée de 15 tonnes/h par rapport à la commande actuelle avec un surcoût d'énergie de 0,02 Gj/tonne.

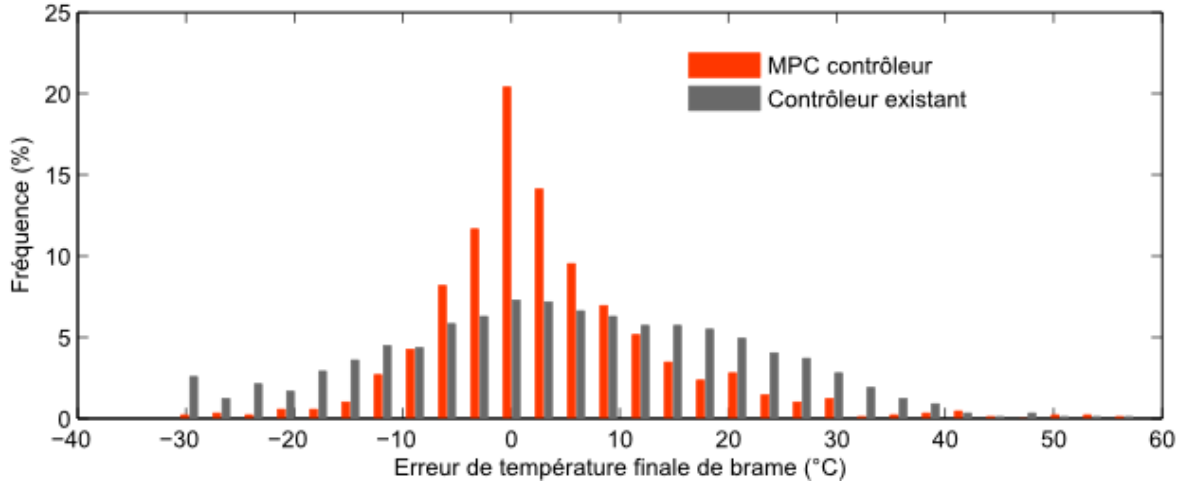

Fig. 49 : Histogramme de l'erreur de température défournée des brames

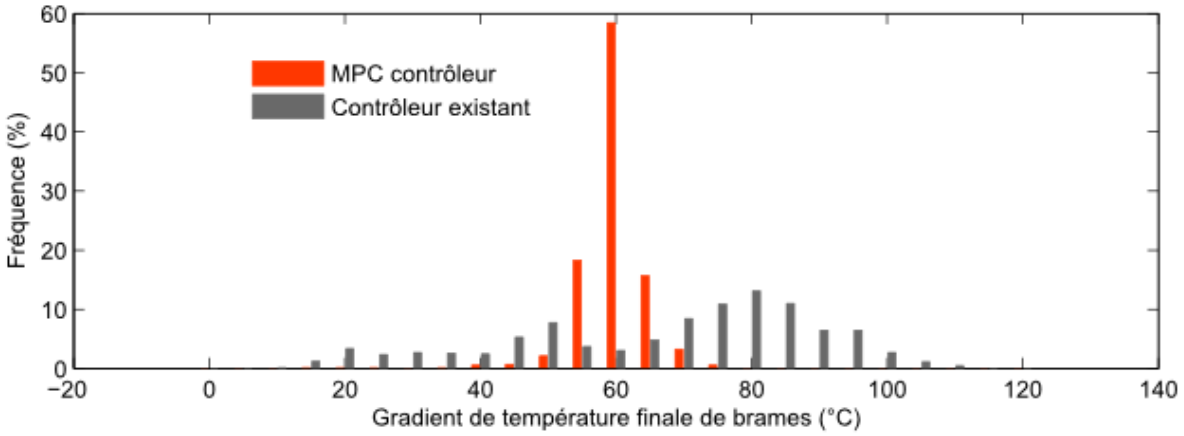

Fig. 50 : Histogramme du gradient de température défournée des brames

Variable	Contrôleur MPC	Contrôleur existant
Productivité moyenne (t/h)	296	281
Temps de séjour moyen (min)	169	179
Durée de réchauffage totale (h)	68	72
Consommation spécifique (Gj/t)	1.27	1.25

Tab. 8 : Comparaison entre MPC et commande actuelle

4. Conclusions et perspectives

4.1. Conclusions

Le but de ce travail était de développer une stratégie de commande prédictive MPC d'un four de réchauffage de brames au sein d'une architecture distribuée hiérarchisée. La fonction première d'un four de réchauffage de brames est de ramener la température des brames d'acier, partant d'une température initiale donnée, jusqu'à environ 1200°C avec une bonne homogénéité avant que les brames n'entrent dans les laminoirs à chaud. La modélisation du four s'avère indispensable pour l'observation des variables telles que la température des brames, la température des parois, etc., et pour la régulation de la température du four. Un modèle permettant de reproduire le comportement transitoire du four a dès lors été élaboré, venant remplacer le modèle stationnaire actuel du four. Ce nouveau modèle permettra en outre de prendre en compte la diversification croissante des brames en termes de température visée, de dimensions, de propriétés des matériaux, etc. Le modèle transitoire a été validé sur des données industrielles et est ensuite utilisé dans la conception de la commande de four.

La tâche de commande des fours de réchauffage concerne principalement deux niveaux. Le niveau 1 a pour objectif la commande des températures de zone du four en agissant sur les débits de gaz des zones (boucle interne). Le niveau 2 quant à lui doit assurer la maitrise de la température des brames en faisant varier les consignes de température de zones (boucle externe).

Le niveau 1 doit permettre de suivre avec précision les consignes de température de zones du four, tout en tenant compte de l'objectif de minimisation de la consommation d'énergie. Pour résoudre ce problème de commande, une stratégie MPC distribuée a été élaborée en se fondant sur un modèle de zones du four identifié et tenant compte de l'interaction entre les zones, déduit expérimentalement sous la forme de fonctions de transfert du premier ordre avec retard pour une entrée en puissances et des sorties en température. Deux approches ont été mises en œuvre pour capturer cet effet de couplage entre les zones du four. La première approche considère que la température d'une zone est une fonction non seulement de l'entrée de cette zone, mais aussi de celles de ses zones voisines. Cela conduit à ce que l'on appelle un modèle de couplage par les entrées. La seconde approche est plus directe; considérant que la température d'une zone est fonction de la

puissance de cette zone et de la température de ses zones voisines. Cela conduit à ce que l'on appelle un modèle de couplage par les sorties. Les résultats des validations ayant montré que le modèle de couplage par les sorties s'avérait meilleur que le couplage par les entrées, ce modèle de couplage par les sorties a été retenu pour la conception de la structure MPC distribuée.

En simulation dans l'environnement Matlab dans un premier temps, les algorithmes MPC distribués proposés ont été testés et validés à partir d'un scénario type issu du fonctionnement réel du four. Les résultats avec un modèle validé de zones de four illustrent un meilleur suivi et une consommation d'énergie plus faible avec la loi MPC distribuée par rapport aux régulateurs PID actuels. La méthode proposée a enfin été mise en œuvre expérimentalement au sein d'un niveau 1 d'un four réel. Corroborant les observations en simulation, les résultats obtenus montrent que la commande MPC distribuée permet d'économiser jusqu'à 3% de la consommation d'énergie tout en assurant un bon suivi des consignes de température de zone. En outre, le taux d'oxygène dans le four est réduit, ce qui conduit à moins de formation de calamine et de décarburation, et donc moins de pertes de matière. La fluctuation de demande globale de gaz est réduite, ce qui induit une pression plus stable de la station d'alimentation en gaz. Par ailleurs, l'utilisation du mode automatique est augmentée de 10% dans la zone d'égalisation, ce qui signifie que l'opérateur du four a moins d'intervention manuelle à réaliser pendant le fonctionnement du four.

Au niveau 2 du four, les consignes de température de zones sont calculées de sorte que les brames puissent atteindre leurs températures visées. La satisfaction de cette spécification est cruciale pour le succès de la tâche de chauffage ainsi que pour l'amélioration de l'efficacité énergétique du four. Cette tâche est par ailleurs soumise à des contraintes de température de zones, de température de brames, et de puissance de zones. La stratégie MPC de ce niveau 2 utilise le modèle de transfert thermique transitoire de façon à prédire la température des brames ainsi que la consommation du four. Les objectifs de chauffage sont formulés via des critères qu'il convient de minimiser, et toutes les contraintes mentionnées sont prises en compte comme des termes de pénalité incorporés dans le critère d'optimisation. Par conséquent, le problème de commande de niveau 2 se ramène à un problème d'optimisation sans contrainte.

Une attention particulière doit être portée à la formulation des fonctions de coût. Les brames doivent être prises en compte avec des coefficients de pondération en fonction de leur priorité. En effet, en général, les brames sont comptabilisées par leur ordre de défournement, ou en d'autres termes en fonction de leur position à l'intérieur du four, plus une brame est proche de la sortie du four, plus elle est importante, car moins il reste de temps pour la chauffer à sa température visée. De plus, une brame ayant une température visée plus importante est plus prioritaire qu'une autre brame voisine, le but étant de la chauffer suffisamment pour éviter le sous-chauffage et le rejet de la brame au laminoir à chaud. La zone d'égalisation est particulièrement dédiée à l'homogénéisation de la température de brames, par conséquent, dans la formulation du critère de minimisation relatif au gradient de température, des brames situées dans la zone d'égalisation sont plus prioritaires que celles des autres zones.

En termes d'algorithme d'optimisation, des algorithmes de recherche directs ont été étudiés, et afin d'éviter d'avoir recours à une forme analytique du gradient de la fonction de coût ou de son estimation, l'algorithme de Nelder-Mead a été privilégié et mis en œuvre en raison de ses avantages, notamment, en plus du précédent, une mise en œuvre simple et une décroissance rapide de la fonction de coût.

En simulation dans un environnement Fortran, la stratégie et l'optimisation MPC proposées pour le niveau 2 ont été testées à partir d'un scénario réel de four. Dans ce scénario, 900 brames sont traitées sur trois jours avec un temps de séjour moyen de 180 min. La production est de 281 tonnes/h. La simulation réalisée avec le simulateur validé du four permet une comparaison de la stratégie MPC et de la stratégie existante de niveau 2. Les résultats obtenus montrent que l'utilisation de la loi MPC peut économiser 5% de la consommation d'énergie du four tout en assurant la température visée et l'homogénéité de la température de brames. De surcroit, malgré un modèle transitoire coûteux utilisé lors de l'optimisation, l'algorithme du simplexe Nelder-Mead est en mesure de fournir les consignes de température dans la période

d'échantillonnage imposée, ce qui signifie que le régulateur MPC proposé pour le niveau 2 peut être mis en œuvre en temps réel.

Une extension de la démarche précédente visant la maximisation de la productivité du four a également été effectuée. Dans cette approche, le cadencement des brames et l'optimisation de la température de brame sont considérés simultanément. En plus de la fonction de coût utilisée précédemment lors de l'optimisation des consignes de température de zones à cadence fixe, l'intervalle de temps moyen entre les défournements est ajouté cette fois comme variable de contrôle. En tant que nouvelle entrée de commande à côté des consignes de température de zones, l'intervalle de temps moyen entre les défournements est soumis à une limite inférieure en raison des contraintes des laminoirs à chaud.

Une originalité dans la conception du MPC visant l'augmentation de la capacité du four est liée au fait que le nombre de brames prises en compte à chaque exécution de l'optimisation est constant alors que l'horizon temporel de prédiction est variable d'une simulation du four à une autre.

La simulation dans l'environnement Fortran montre alors que la production du four peut être augmentée avec un surcoût très modéré de la consommation d'énergie. En outre, la précision de la température de sortie de brames et son homogénéité peut être significativement améliorée. Les résultats obtenus montrent que la productivité de four peut être augmentée jusqu'à 15 tonnes/h. Dans la pratique, ces résultats prouvent que le MPC peut être utilisé pour informer l'opérateur du four, lui permettant d'ajuster ainsi la productivité instantanée du four. Cette information aidera l'opérateur du four pour ajuster le cadencement des brames afin que le four puisse fonctionner à sa productivité maximale.

4.2. Perspectives

La stratégie MPC distribuée proposée a été mise en place et fonctionne actuellement à temps plein dans les fours de l'usine de laminage à chaud de l'usine ArcelorMittal de Florange. Les résultats du MPC niveau 2 obtenus en simulation à partir des données du système réel sont convaincants, de sorte que cette stratégie MPC sera implantée à court terme sur le four pour sa validation expérimentale.

Sur le même laminoir à chaud se trouve un autre four de réchauffage de conception similaire mais avec un dimensionnement des zones différent. Une adaptation à court terme des stratégies développées est donc à prévoir pour ce nouveau four, qui ne concernera en fait que la modification des paramètres du modèle du four.

Dans le développement du MPC pour le niveau 2, nous avons utilisé des bornes constantes de températures de zone lors de la formulation des contraintes du problème d'optimisation. Dans la mise en œuvre expérimentale sur four réel, ces limites seront amenées à varier dans le temps. En effet, à chaque instant, les limites de températures de zone dépendent non seulement de la puissance disponible dans les zones du four, mais aussi de la charge de brames située dans chaque zone. Pour résoudre ce problème, le modèle de four pourra effectivement donner des estimations de ces limites qui pourraient ensuite être utilisées pour l'optimisation.

Enfin, la grande perspective de ce travail est à plus long terme la mise en œuvre du MPC niveau 2 pour tous les fours de réchauffage de brames des usines sidérurgiques d'ArcelorMittal, qui devrait avoir des impacts économiques et environnementaux significatifs. Plus largement donc, il conviendra de généraliser la démarche à d'autres types de fours de réchauffage dédiés à différents types de produits tels que les blooms, les billettes, les tubes, afin de leur appliquer cette stratégie MPC distribuée et hiérarchisée.

Contents

List of Figures

List of Tables

Part I

Process description and modeling

Chapter 1

Introduction

In this chapter, a brief introduction to steel industry is firstly given. Next, the context of slab reheating furnaces is introduced. The challenge of control task and its influence on energy performance of the furnaces are then presented. Finally, an overview of the dissertation is given summarizing the flow of the work through the chapters.

1.1 Context and motivation of the work

In steel industry, iron ore (Fe_2O_3) and ferrous scrap (Fe) are transformed into steel products. Iron must undergo several thermal, mechanical, or thermo-mechanical processes before it is suitable for other industrial applications such as construction, automotive, or space industry, etc. Those processes can be grouped into four stages: iron making, steel making, continuous casting, and hot rolling. In iron making, molten metal is produced by melting ferrous scrap or burning the mixture of coke and iron ore at 1450°C in blast furnaces [Tri03]. In steel making, impurities such as nitrogen, silicon and excess carbon are removed from raw iron liquid. Then, alloying elements such as manganese, nickel, chromium and vanadium are added to produce different grades of steel. In the step of continuous casting, the obtained steel liquid in steel making is cooled down and formed into solid blocks. These steel blocks are semi-finished products that have form of slabs with dimensions of 0.25m x1.5m x10m for example. These slabs are then stocked in a slab yard or transported directly to hot rolling process when they are needed to fill client orders. The primary function of hot rolling process is to heat steel slabs nearly to their melting point, then roll them thinner and longer through successive rolling mill stands driven by motors and finally coil up the lengthened steel sheet for transport to the next process, see Figure 1.1.

Reheating furnaces are positioned just before the hot rolling mills, utilized for reheating, and also for chemical and metallurgical treatment of steel slabs. The role of these furnaces is to provide sufficient hot slabs for rolling mills. The furnace design is based on capacity of hot rolling mill. Steel slabs are usually heated from ambient temperature to 1250°C; this case is called cold charging. In an integrated plant which has both continuous casting and hot rolling processes, slabs might be loaded into reheating furnaces at 800°C [Mår92]; this case is called

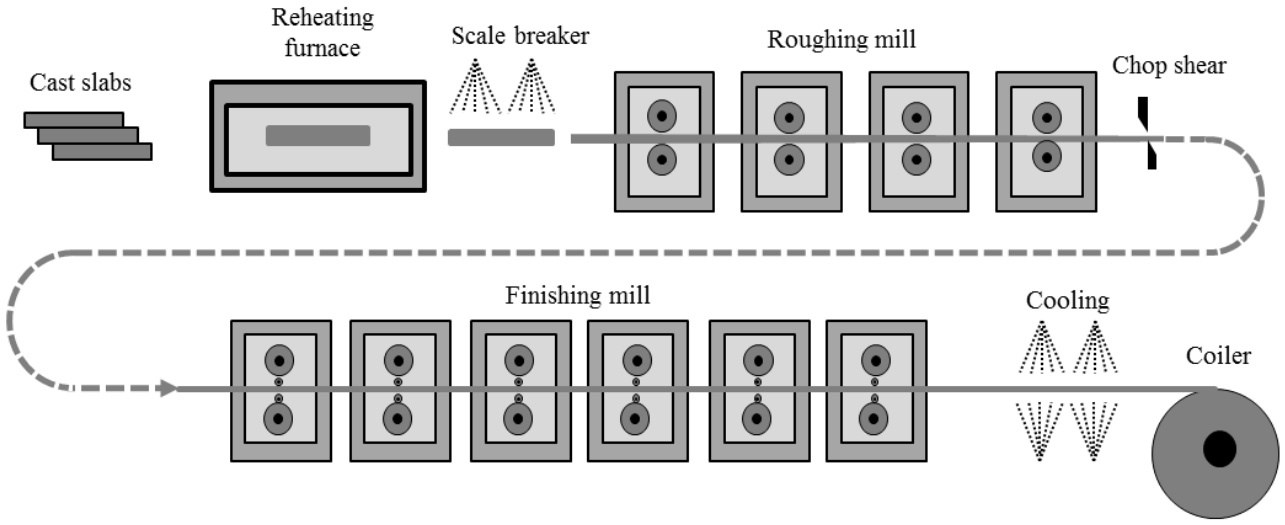

Figure 1.1: Hot rolling of steel slabs

hot charging. The performance of a reheating furnace is evaluated by the precision of discharge temperature, internal temperature gradient of slabs and its energy efficiency. Some phenomenons such as scale formation, decarburization, surface melting or geometrical distortion like bending of slabs are to be reduced as much as possible.

To realize the task, reheating furnaces are constructed with several zones which can be thermally isolated or coupled one from others. Transportation mechanism is a system of mobile and fixed skids. In order to move the slabs forward, the mobile skids are firstly lifted up, next translated, then brought down to the same level of the fixed skids, and finally brought back to the initial position. The presented mechanism is used in walking-beam reheating furnace type which differs from pusher-type reheating furnace where slab rows are pushed by machine at the entry of furnace and slabs slide on the support skids. For instance, the considered furnace in our study is a walking-beam type furnace, it has dimensions of 6m x12m x60m and a production rate that can come up to 400 tons of steel per hour.

In terms of energy consumption, about 20 percent of steel production cost is from energy use [LCoIE85, STY99]. In a steel plant, the largest energy consumers are blast furnaces in ironmaking process with 44 percent of total energy use and reheating furnaces of hot rolling process are second energy consumers with 17 percent of total energy use [STY99]. Particularly, annual energy consumption of the considered slab reheating furnace is equivalent to the consumption of 100,000 houses [PD10]. Therefore, the energy efficiency of reheating furnaces influences significantly the energy performance of the whole production chain. The aftereffects of energy savings are the lowering of greenhouse gas (CO_2, N_2O, etc.) emissions and the reduction of environmental footprint which are subjects to ever more restrictive regulations.

Many design factors affect the energy efficiency of reheating furnaces like furnace geometry, number of zones, wall and roof insulation, and design of transport mechanism [Mår92]. Charging temperature of slabs may vary

from ambient temperature to 800°C as mentioned before, which is a significant condition affecting energy efficiency of the furnace. These are prefixed conditions of a furnace and can not be intervened in temperature control system of furnace.

Furthermore, operational practices are also crucial factors for the energy efficiency. Operational conditions may change from mill to mill but also for the normal operation of each furnace in the same mill. Different disturbances and deviations occur in a furnace operation like delays in the rolling mill, diversity of slab dimensions, fuel composition, charging temperature. For these reasons, optimal manual control is in general an overwhelming task. Even skilled operators generally tend to run the furnace with large safety margins to assure the required discharge temperature of slabs. This practice often leads to excess use of energy and to quality problem due to overheating of slabs.

To increase energy efficiency, more efficient control of the furnace is required. More accurate control will improve accuracy of slab temperature and combustion process. Therefore, safety margins on slab temperatures will be reduced and the overall temperature of the furnace will be lowered.

The control of a slab reheating furnace is an intricate task. From theoretical point of view, reheating furnaces are non-linear systems with dynamic interaction of multiple subsystems which refer to furnace zones. The thermal behavior of the system is rather slow due to high thermal inertia of slabs and furnace enclosure. Efficient operation of the furnace essentially needs knowledge of certain physical quantities such as slab temperature profile, furnace gas and wall temperatures, fuel and combustion air flows, oxygen content of exhaust gas, position of each slab inside the furnace [Mår92]. Unfortunately, these measurements may be difficult, unreliable, or even in-feasible as a consequence of harsh environmental conditions of the furnace like high temperature, dirt, dust, electrical noise, wetness [Ste11]. Since slab temperature profile can not be directly measured; it is usually estimated by a heat transfer model. In addition, homogeneity of slab temperature profile is a critical quantity that needs to meet requirement to prevent rejection of slabs in rolling mills. Process variables such as refractory walls and roof temperature, fuel and combustion air flow rates are also subject to constraints. Furthermore, the diversity of product is continuously increasing in terms of dimensions, steel grade, desired discharge temperature. Changes of product types in reheating furnace are more and more frequent. Therefore, heating process is usually a non-steady state operation.

The purpose of temperature control of reheating furnaces can be simplified as regulation of thermal energy injection and distribution along the furnace length to heat slabs to correct temperature. Control variables available for temperature control are burner fuel and air flows or more indirectly zone temperature set-points for burner regulators. In some cases, it is also possible to adjust the throughput rate of furnace with condition that it does not exceed the rolling mill capacity. Dealing with complexity of the system, the current control structure of the furnace exploits the hierarchical cascade control concept, as shown in Figure 1.2. At level 2, desired discharge temperatures of slabs are tracked via a heat transfer model of furnace. The control variables are temperature of furnace zones. Level 2 then communicates with level 1 the required zone set-point temperature to obtain the target temperatures of slabs. At level 1, necessary heat power to achieve set-point temperatures of each zone is calculated and then the corresponding fuel and combustion air flow rates are derived.

In the considered furnace, current level 2 of control system is designed based on a steady-state model. This model can estimate slab temperature profiles using radiations between slabs, furnace enclosure and gas. The estimations of temperatures are used to decide the temperature set points of each zone. For this purpose, a performance-based choice is used, and no energy efficiency criterion is considered. At level 1, traditional PID controllers are used to control zone temperatures. In fact, PID controllers can not consider the interaction of furnace zones which is an important phenomenon inside furnace. Therefore, fuel combustion process is not optimized, which leads to low energy efficiency.

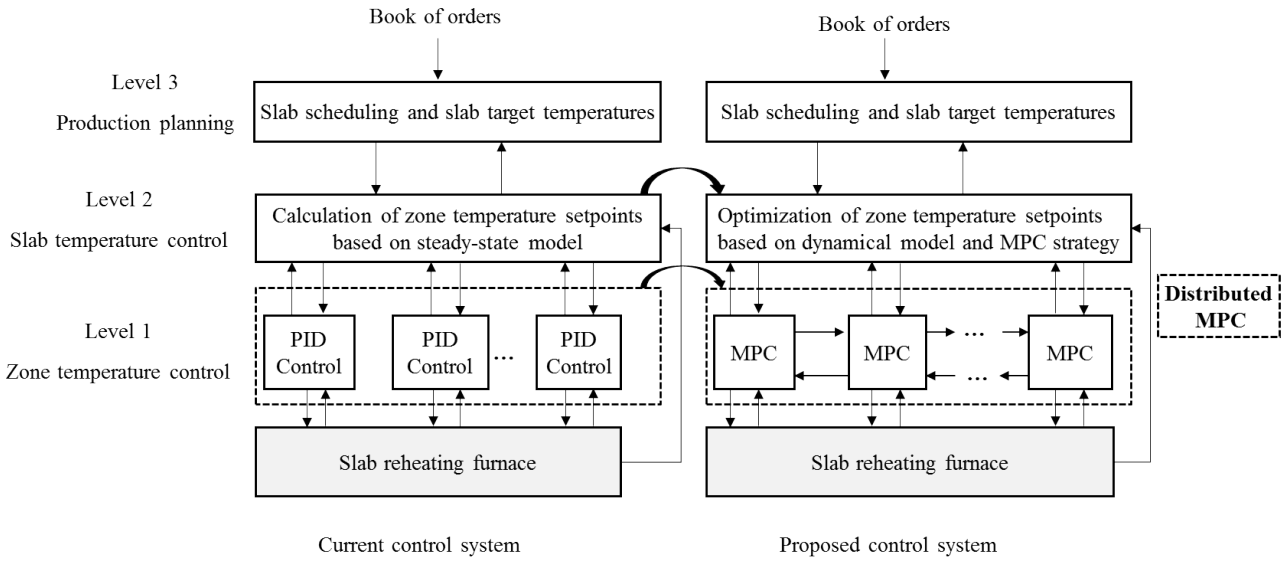

Figure 1.2: Hierarchical cascade structure of control system

In order to improve heating quality and reduce energy consumption, this work proposes a new control structure as shown in Figure 1.2. The PID controllers of level 1 are replaced by a distributed model predictive control law allowing the consideration of the interaction between furnace zones. Furthermore, in level 2, the steady-state model is replaced by a non-linear dynamical model, this model can predict transient behavior of the furnace, permitting a better prediction and estimation of slab temperature profiles.

The work mainly aims for heating quality and energy efficiency improvements. However, the reduction of scale formation, decarburization of slabs are also expected to come with those improvements. In addition, flexibility of production rate is studied as well to boost furnace productivity.

1.2 Structure of this dissertation

After the *Abstract* and the *Introduction*, *Chapter 2* presents in details the important operational aspects of walking-beam slab reheating furnace such as combustion system, mechanical system, energy flows inside the furnace, etc.

In *Chapter 3*, a brief literature review on reheating furnace modeling is given. The steady-state mathematical model that gives the basis for current control system of furnace is described. Afterwards, a dynamical non-steady-state numerical model is presented. The advantages, disadvantages and utilization of two models are discussed. At the end of the chapter, measurement data from the considered walking-beam slab reheating furnace is used to validate the numerical model.

Chapter 4 will give the overview of significant works on control system of reheating furnaces. The existing control strategy of considered furnace is described. And the proposed new solution is then presented.

In *Chapter 5*, an algorithm of distributed MPC is developed for level 1. First, modeling and identification of furnace zones are addressed. Second, formulation of optimization and iterative algorithm of distributed MPC is presented. Finally, simulation results and industrial validation are discussed at the end of the chapter.

Chapter 6 is dedicated to the control at level 2 where set points of zone temperature are optimized. Firstly, the optimization is formulated. Then, potential dynamic optimization methods are discussed. Finally, simulation results based on industrial scenario are presented and discussed.

Chapter 7 studies the improvement of production flexibility and boosting furnace production rate. Literature review on important studies is given. And the optimization problem is formulated. Simulation results are discussed at the end of the chapter showing possibility for an increase of furnace throughput.

In *Chapter 8*, an effort of generalization of control solution is given for different types of furnaces with considerations of their distinctive characteristics.

Conclusions and perspectives of future work are given in *Chapter 9*.

Chapter 2

Slab walking-beam reheating furnace

The context and motivation of this work have been clarified in the *Introduction*. This chapter describes more in detail the considered walking-beam slab reheating furnace by firstly distinguishing it with other types of reheating furnace. Second, the physical structure and transportation mechanism of the furnace is detailed. The combustion, insulation and cooling systems are then presented. Flows of energy inside the furnace are also discussed.

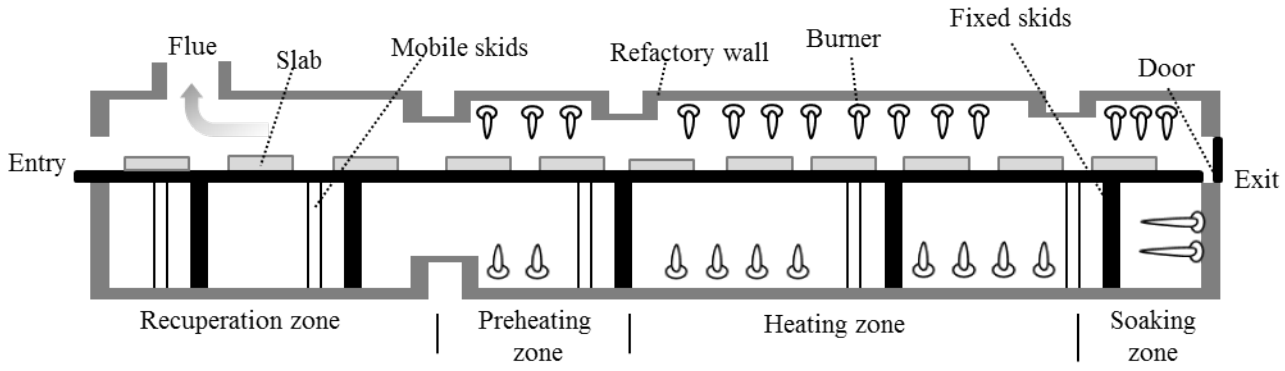

Figure 2.1: Lateral view of a walking beam slab reheating furnace

2.1 Transportation of slabs

Steel slabs are conveyed from slab yard to the entry of the furnace. An automated mechanism is programmed to take the slab standing at the furnace door when there is enough space and predetermined loading event of the slab happens. Between slabs, some distances are imposed to facilitate gas circulation and therefore heat exchange inside the furnace, see Figure 2.1.

Inside the furnace, slabs are placed on skid beam support system. These beams are subject to heat corrosion because the operating temperature are usually around 1250°C. Nowadays, skid beams may consist of water-cooled steel members and topped with refractories which is the only parts exposed directly to the heat of the furnace.

The movement of slabs can be executed in the two directions of furnace length, which can help modifying the slab sequencing. In the furnace, two type of skid beams are utilized, one is mobile also called walking beam and the other is stationary. To move slabs forward, the mobile skid beams are lifted up above the stationary ones, which also brings up the slabs. The whole set mobile skids and slabs is then translated forward through a given step length. Next, mobile skids are pushed down so that slabs return to the level of fixed skids. Finally, mobile skids come back to the initial position ready for a new cycle of movement. Slabs are now in new positions inside the furnace.

Slab movement is a discontinuous process. At one position, a slab will stay for an amount of time so that it can absorb enough heat before being discharged from the furnace, see Figure 2.3. The scheduling of slab movements is programmed in advance. The loading and discharging events of a slab are referenced relatively to those of the previous slab. For instance, a slab could be loaded 5 minutes and discharged 4 minutes after the previous slab. Therefore, the scheduling task can be simplified into imposing interval time of loading events and discharging events of consecutive slabs.

In general, a walking beam movement happens following four configurations. The first one is when a new slab enters, and a slab is discharged from the furnace. Second, no slabs are removed from furnace but a new slab is loaded into it. In the third case, a slab is discharged but no slabs enter the furnace. And in final case, all slabs are moved forward but no slabs are removed from or loaded into the furnace.

The contact between slabs and fixed skids causes marques on the lower surface of slabs. These marques indicate lower temperature of steel compared to other regions of slab surface. Therefore, as shown in Figure 2.2, fixed skids of soaking zone are interspersed between those of other zones to reduce the apparition of marques.

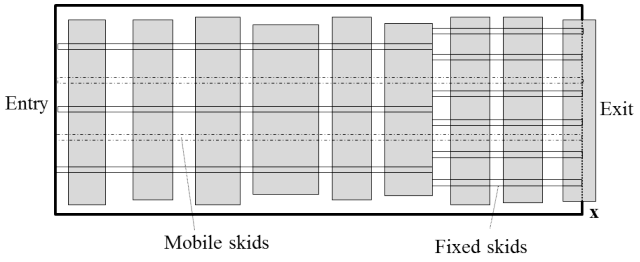

Figure 2.2: Top view of a walking-beam slab reheating furnace

It happens usually in furnace operations that products of a same thickness or material are gathered together to be heated consecutively. With different thicknesses or materials of heated slabs, the required force applied on steel is different, therefore, rolling mills are regularly changed with different sizes and types of cylinders. For this reason, short stops of furnace are usually scheduled. On the other hand, periodical maintenance or operational

incident cause complete stops of the furnace. The furnace production rate is defined as the quantity of steel

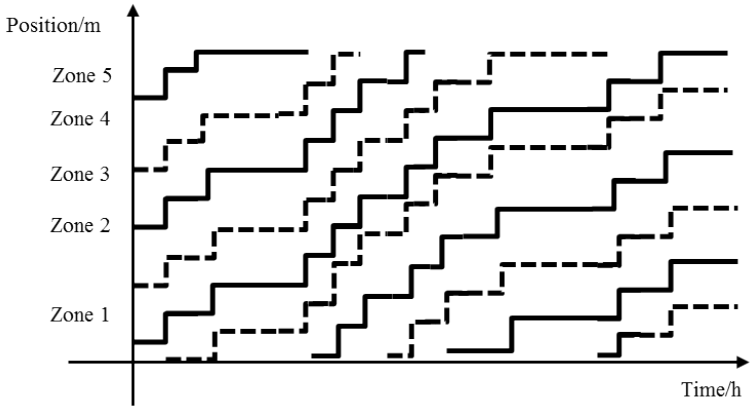

Figure 2.3: Example of path-time diagram of slabs

heated to the desired temperature per unit time. If a slab reheating furnace constitutes a bottleneck in the process chain, the control strategy should make the full use of the furnace capacity. If not, other objectives like energy efficiency, reduced scale formation and decarburization may be prioritized.

2.2 Combustion and heating of the furnace

Thermal energy is supplied into furnace by gas burners which are installed in the refractory wall of the furnace. Each zone of the furnace is equipped with a certain amount of burners, and its input power is limited. In addition, between upper and lower sub-zones of the same zone, burners are coupled with a certain ratio so that heat flux absorbed into upper and lower surface are as similar as possible.

In terms of combustion process, gas and combustion air must be appropriately mixed. With a required heat power for a zone, the gas flow rate of each burner is determined based on the gas heating capacity which defines the thermal energy released as the gas undergoes complete combustion. The flow rate of combustion air is then computed accordingly to gas flow rate. In practice, air/gas ratio is controlled to avoid excessive rich burning which results in complete combustion with partially burned or unburned gas escaping from the furnace without releasing heat where it can be used effectively [Tri03]. On the other hand, air/fuel ratio control also prevents excessive lean burning, in which extra unused air passes through the furnace, absorbs heat, and carries that heat out of the furnace, therefore unabsorbed by the loads [Tri03].

Heating process of a slab starts by the exchange of energy with the exhaust gas in the recuperation zone. This zone is not provided with burners. At preheating and heating zones, main heat inputs are released to bring the average temperature of slabs to the desired ones. However, internal temperature gradient of slabs are still high at the end of heating zone. In fact, slab surface temperature can rise rapidly but due to thermal inertia of slab the core slab temperature takes more time to increase. At the end of heating zone, slabs are generally not ready for rolling. Therefore, soaking zone is dedicated for equalization of internal temperature of slabs. In general,

soaking zone is isolated from other zones to preserve heat of slabs. Surface temperatures of slabs are maintained at desired discharge temperature so that homogenization will take place during residence period inside the zone. Temperature evolution of a slab inside the furnace is shown in Figure 2.4.

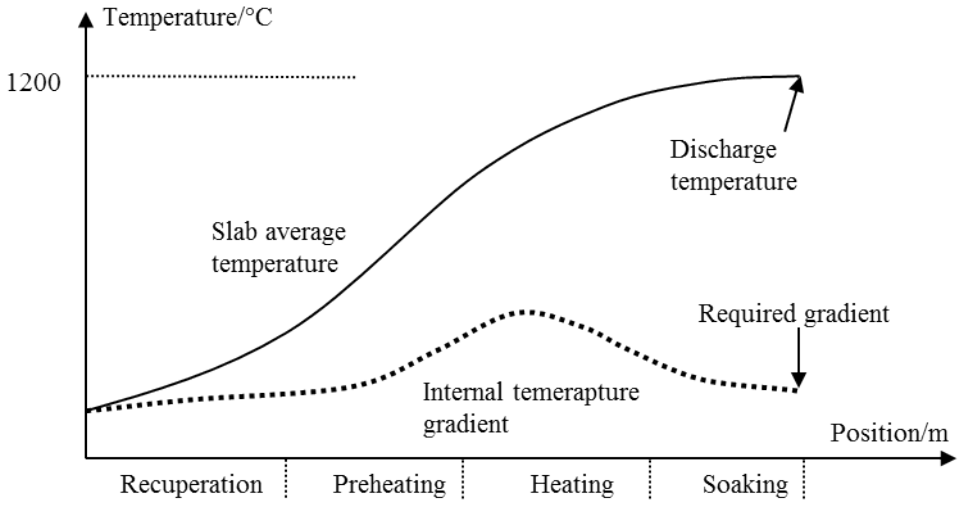

Figure 2.4: Heating curve of a slab with respect to its position

A phenomenon that may significantly affect heating process is the scale formation. The slabs inside furnace are exposed to hot-oxidizing combustion gas that contains H_20, O_2, $C0_2$ and N_2. These hot gases transfer also heat into slabs. However, reactions between slab surfaces and furnace gas happen as slab temperatures increase, which causes the formation of an iron oxide film usually called scale. The thickness of the scale depends on residence time in the furnace of slabs, their surface temperature, temperature and composition of the furnace gas. The scale formation causes material losses of the slabs. Moreover, because thermal conductivity of the scale is much smaller than that of steel, the existence of the scale can have high effect on heat transfer behavior of slabs.

On the other hand, decarburization is a chemical process that affects quality of slabs. Decarburization is the change of structure and content of steel slabs in which some carbon in the slab surfaces is lost due to the reaction with O_2. Since carbon content determines the physical performance of steel, the decarburization phenomenon of the furnace has to be reduced.

2.3 Temperature measurement

Temperature control can not have better precision than the sensors upon which it relies [Tri03]. For temperature control of walking-beam slab reheating furnace, it is necessary to know the temperature distribution in the slabs, in the refractory furnace walls and in the atmosphere of the furnace. In practice, the use of thermal sensors is a challenging and costly task due to high temperature and harsh environment of the furnace.

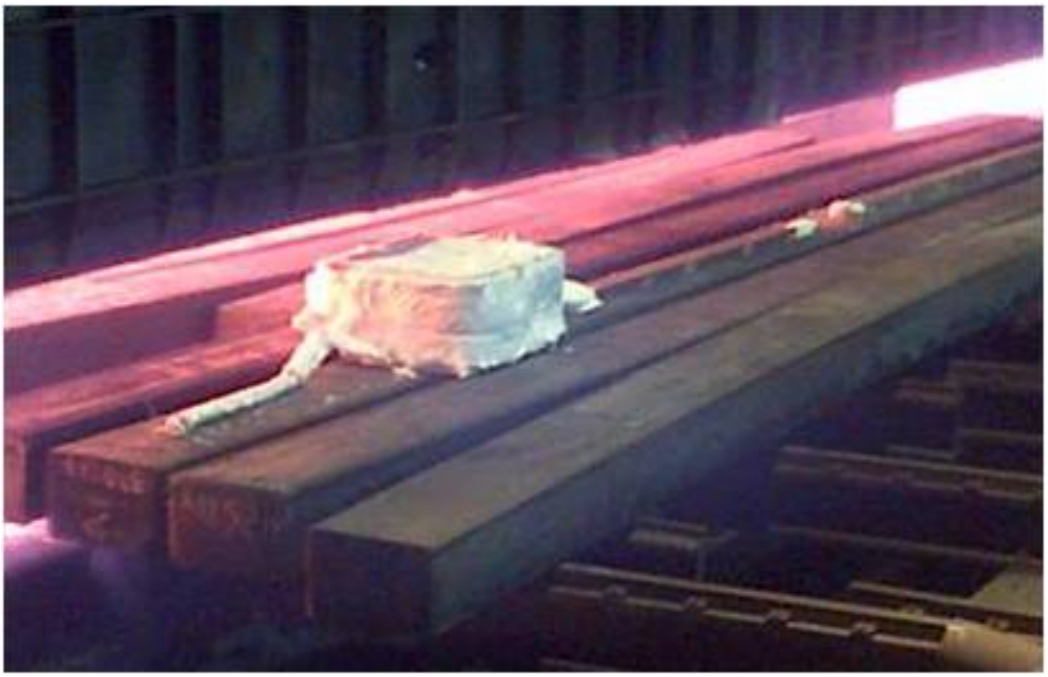

Figure 2.5: Instrumented bloom with thermal box before charging into the furnace

During normal furnace operation, measurement of slab temperature profile is infeasible. Therefore, it is usually estimated by a heat transfer model. Sometimes, the surface temperatures of the slabs can be monitored by *radiation pyrometry*, but the temperature profile still has to be estimated by a conduction heat transfer model inside slabs.

Thermocouples are frequently utilized for measuring wall temperatures and gas temperature in reheating furnaces. For instance, several thermocouples measure gas temperatures at different positions of a zone, an average temperature is then used for zone temperature regulation. The method is relatively robust, and sufficiently accurate for most control purposes. In contrast to pyrometers, themocouples require thermal contact, may give significant time delays, and can only make local measurements.

On the other hand, thermocouples can be implemented in holes drilled at several locations in in a product. During the heating time, the thermocouples signals were recorded on an acquisition system travelling with the product inside the furnace. Because of the extreme thermal conditions in the furnace, the acquisition system is placed inside a water box to keep its temperature lower than 100 °C. The thermal box is also covered with additional insulation as it shown in Figure 2.5 to protect it from radiation. In the example, the dimensions of the instrumented bloom were 160 mm in width, 128 mm in thickness and 10 m in length. The cooling chamber was place at as shown in Figure 2.5. This acquisition system is utilized only for the validation of furnace model which needed exact measurement of slab temperature through the length of the furnace.

2.4 Energy flows of reheating furnace

As previously mentioned, in a hot rolling mill, approximately 80 percent of energy consumption is due to reheating furnaces [Ste11]. Therefore, energy efficiency is a key performance value that directly affects the operational costs. The most significant energy flows of a walking-beam reheating furnace are shown in the Figure 2.6. The heat released by gas combustion is distributed to all furnace interior. A considerable portion of the total energy is stored in hot exhaust gas that leaves the furnace. Energy losses are minimized by a recuperation unit that reuses hot exhaust gas for preheating the combustion air and gas. Effectively, the recuperation process cannot totally recirculate the heat back into furnace, which means that some thermal energy will always be lost. In some cases, the heat losses to the flue is observed as 30% of the total supplied energy [CCL05, ZXWS01]. Without the recuperation unit, heat losses due to flue gas may increase to 50% [Ste11].

The other losses of energy are caused by door openings or expansion of walking beam joints. In addition, refractory walls and roof conduct heat from furnace interior to the outer surface. Heat is subsequently emitted to the external environment by radiation and convection. Energy that heats walking beams or skids may account for more than 10% of the total energy losses [CCL05]. Cooling system equipped for skids or walking beams is connected to the heat recovery system improving the overall energy efficiency of furnace.

The heat absorbed by the slabs is considered as helpful energy. Additionally, its absolute value may be predetermined by the material properties, the initial temperatures, and the desired discharge temperatures of slabs. According to [CCL05], from 40% to 60% energy of the heat supplied to a reheating furnace is effectively absorbed by the slabs.

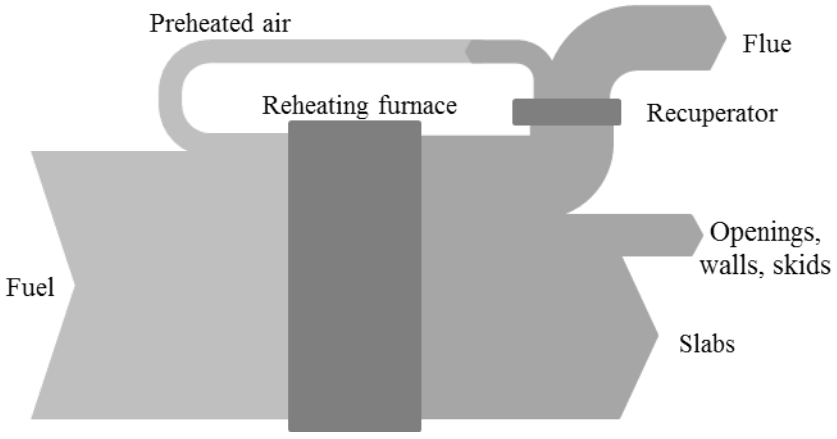

Figure 2.6: Energy flows in a walking-beam reheating furnace

Energy losses can be reduced by insulation of the furnace, recuperation of flue gas, heat recovery from cooling water or steam. On the other hand, energy savings can be obtained by reducing consumption of raw material while volume of heated products remains the same. This method is associated with improved heating quality, e.g., reduced material loss through scale formation, decarburization, reduced steel melting, and reduced need for reworking.

A better precision of final slab temperature may lead to energy savings. If the slabs temperature can be controlled more accurately, the desired final temperatures can be reduced because the safety margins between the controlled final slab temperature and the temperature at least required for rolling mill can be smaller.

Hot charging may constitute also a method for energy savings. Instead of storing slabs after continuous casting process, which cools slabs to ambient temperature, they may be immediately transported to the reheating furnace. This method can save energy and operational costs. It reduces residence time of slabs, gas combustion and therefore flue gas. Higher furnace throughput, and increase furnace capacity is also obtained with hot charging compared to cold charging. However, integrated conveyance system is required to handle hot slabs from continuous casting stage. In addition, it happens practically that not all the slabs are loaded into reheating furnace at high temperature. Instead, a mixture of hot and cold slabs is fed to the furnace, which may make temperature control more complicated and more considerations will be needed.

After all, it is important to improve the strategy of temperature control, for which a model with high precision is necessary. As a result, the modeling of the furnace will be discussed in the next chapter.

Chapter 3

Modeling of slab walking-beam reheating furnaces

The presentation of the furnace has given an overview of its functional aspects and the main challenges for furnace control task. This chapter deals with furnace modeling. Firstly, a non exhaustive literature review of furnace models that are usually used is considered. Secondly, a steady-state model of the considered furnace describing the heat transfer phenomena inside furnace is presented. Finally, a dynamical model is introduced. This model is developed and tailored for use in transient phase of the furnace. The advantages and disadvantages of these two models are also discussed.

3.1 Literature review

A model of reheating furnace can be used for energy consumption analysis, estimation of physical quantities, simulation of control design, or analysis of furnace design. The specifications are detailed as follows [Ste11]:

- Energy consumption analysis of influence of insulation, oxygen enrichment, excess air coefficient, temperature of supplied combustion air and fuel, slab hot charging, distribution of heat input through burners along the furnace
- Estimation of quantities that cannot be measured such as: slab profile temperatures, walls and roof temperatures, scale formation on slab surfaces, composition, temperature, and flow rates of flue gases.
- Simulation for control design which needs analysis on control strategies, controller gains, fuel consumption, influence of slab characteristics (dimensions, initial temperature, desired final temperature profile, material, etc.), and influence of residence time and time intervals between discharging events.

- Analysis of furnace design like: furnace geometry, shape and cross sectional area of flue gas ducts, shading effects, influence of refractory wall materials, location, size and type of burners, location and type of temperature sensors, modifications of existing furnaces.

Mathematical models are crucial for estimating or predicting slab temperature profiles by an observer algorithm or by a simulator. In order to compute slab temperatures, a model has to consider the heat transfer into surface of the slabs, and then the conduction through the slab thickness. Sometimes, temperature of slab surfaces can be measured by radiation pyrometry but there is still a need for a model that calculates the temperature distribution inside the slabs. In addition, mathematical models of furnaces can provide statistics about the gas flow, pressure inside the furnace, and energy consumption of the furnace [FRLD06a, GSR+98a, HCK10a, HHLW08, KHK00, KKY+00, ZIS97], excess oxygen ratio [KKY+00, ZIS97], scale formation [JLKK10, Led86b, RSCG86, Sch96], and decarburization of the material [Sch96].

In real-time applications, computational constraints demand a reduction of mathematical furnace models to less expensive models while sufficient accuracy is still maintained [GSR+98b, vDSK02, HVS04, KvD94, ESYS97, KKY+00, Led86a]. Mathematical furnace models can be simplified by means of reduction techniques. However, for a highly nonlinear model, the application of reduction techniques is a rather challenging task.

In general, models of reheating furnaces are classified based on the integrated physical theory. White box models are derived and created from fundamental equations of chemistry and physics. Therefore, structure and parameters of white-box models have an explicit connections to the phenomena and quantities observed in the real system. On the contrary, black-box models or grey-box models utilize generic structures of system identification theory which do not allow direct physical interpretations [KvD94, vDSK02]. However, black-box models are able to simulate unknown disturbances which affect physical effects.

Essentiallly, there are three methods for modeling heat transfer and the temperature distribution inside the furnace enclosure [JLKK10] [KHK00]. First, the furnace enclosure is divided into distinct zones which are assumed to be a *stirred vessel* [ESYS97, GSR+98a, HBK09, KMP94, KKY+00, KvD94, PW98, YNY91, ZCX+02]. Second, the furnace volume may be considered as a tube with a 1-*dimensional* plug flow in the longitudinal direction of the furnaces [Ste11]. Then the furnace temperature profile is determined along this direction [BBG08, GSR+98b, HVS04, JLKK10] [JK07]. Third, 2-*dimensional* or 3-*dimensional* temperature profile inside furnace is calculated with techniques of fluid dynamics [HCK10b, HBK09, JKZ05, Kim07].

Temperature profile of slabs is usually assumed as 1-*dimensional* along the thickness direction of the slabs [vDSK02, Led86a, RSCG86, SWKK11, WK99, YNY91, ZCX+02]. In [SWKK10], some conditions that justify this assumption are given. Other models in [CCL05, ESYS97, GSR+98b, JK07, Kim07, Li86] provide 2-*dimensional* temperature distribution in the slabs. These models allow an analysis of the influence of skids on the in-homogeneity of the slab temperature profile. In [HCK10b, JKZ05], a 3-*dimensional* temperature profile is computed.

Frequently, heat conduction equation is solved by means of the finite difference method [CCL05, ESYS97, GSR+98b, HBK09, JLKK10, Led86a, WK99, YL88, ZCX+02]. The finite element method is used in [JK07,

ZIS97]. An alternative approach is adopted in [YUU94], such as the method of weighted residuals with up to 5 polynomial trial functions. Some works account for temperature dependence of material parameters [WK99, SWKK11, SWKK10, Led86a, JLKK10, JK07].

Most of the models reported in the literature are based on the Fourier's law for heat conduction problem inside the slabs. Nevertheless, there are different approaches of reflecting the heat exchange outside the slabs. Three physical principles contribute to the heat exchange between the slabs and the environment:

- Heat conduction from skids and hearths
- Convective heat transfer from flue gas
- Radiative heat transfer from flue gas and furnace walls

Because a furnace operates in relatively high temperature, radiative heat transfer is usually the dominant mode of heat exchange [HCK10a, HHLW08, JK07, Kim07, KKY+00, Mår92, ZIS97, Li86]. The radiative effect of reheating furnace is characterized by *Stefan-Boltzmann* radiation law, which contains 4^{th} power of gas and surface temperatures. In the first category, furnace models do not take into account the radiative interaction between the flue gas and the surrounding surfaces. The low accuracy of these models can be reduced if local temperature of flue gases and furnace walls are averaged. The resulting mean temperatures are sometimes referred to as zone temperatures. Methods that compute the important effect of gas radiation like the *net radiation method* [Inc11] and the *zone method* [Nob75], constitute the second category.

Some models neglect the radiative heat exchange in the longitudinal direction of the furnace [vDSK02, YUU94, WCGS99, ESYS97, HVS04, JK07, KKY+00]. Then, if the gas flow inside the furnace is not taken into account, there is no themodynamic interaction between slabs. This approach leads to particularly simple mathematical formulations as presented in Section 3.2.2.

Linearized dynamic model is also a possibility of furnace modeling, it could be sufficiently accurate for the respective purpose [vDSK02, ESYS97, KKY+00, KvD94, WK99]. However, it shows some limits since the 4^{th} power non-linearity associated with radiative heat transfer is significant. In addition, the temperature of slabs during heating process varies in a wide range, which means that there is no steady operating point for linearization.

A more elaborate furnace model is constructed if a full *energy balance* supplements the computation of radiative heat fluxes. Energy balance method helps evaluate the system in terms of energy consumption and efficiency [ESYS97, GSR+98b, KKY+00, YNY91]. Accurate formulation of energy balance also requires a model of heat losses through the furnace walls and the cooled skids [CCL05, ESYS97, ZIS97, GSR+98b, HHLW08]. The *convective heat transfer* happens between the gas flow and the surfaces inside the furnace as presented in [WK99, HCK10a, KMP94, HHLW08, HVS04]. According to [HCK10a, HHLW08, KHK00, ZIS97], 5% of heat input to slabs comes from convection.

As discussed in [LR05], the parameterization of a model can be a laborious task with numerous measurement series. However, it is a necessity to have accurate estimation of slab temperature, and model parameterizing

is generally rewarded once achieved with the capacity of predicting online tracking quantities that cannot be measured due to consequent costs. This argument especially applies to the slab temperature profiles. The effort put into parameterizing physical models may be reduced if a *stochastic model* is used as in [HVS04].

Black box methods can avoid physical modeling as used in [KKY+00, KvD94]. Step response experiments are carried out on the real furnace system to obtain data for non parametric system identification with a discrete-time autoregressive model with exogenous input enhanced by time-lag behavior.

Furnace models for *steady-state operation* are usually computationally low-cost, which makes them suitable for control design [Li86, WCGS99, Kim07]. Steady-state operation implies constant slab speed, regular discharge events, similarity of product dimensions and material [SGW+11]. In that case, the slab temperature inside furnace is a function of slab position [SGW+11].

Furnace models are utilized as *comprehensive model* and *reduced model*. The comprehensive model accounts for detailed physical effects and therefore captures the system dynamic as accurately as possible. This model is intended for furnace simulation and estimation of physical quantities that cannot be measured in the real system. Inputs to this model can be flow rates of fuel and combustion air. Possible outputs are energy flows within the furnace, temperature profiles of the slabs, of furnace walls, as well as composition, temperature, velocity of flue gases. Comprehensive model might constitute computational burden. Therefore, reduced model may be constructed by neglecting convective heat transfer and simplifying furnace operation conditions. Optimization and control applications usually need reduced model to fit real-time operation requirements.

Nowadays, fast development of computer hardware offers more and more computation power as well as data storage for industrial applications. Therefore, in our case, detailed furnace models can be implemented providing high precision estimation or prediction of slab temperatures and other furnace variables. Furthermore, based on these detailed models, advanced control technologies can be designed for furnace control and optimization to improve furnace performance. In the following section, a steady-state model which is currently implemented in the considered furnace will be described. After that, a dynamical model is presented, this model is utilized in the design of a new furnace controller to increase the furnace performance.

3.2 Steady-state model

For the development of a model, the geometry of the furnace is simplified as described in Figure 3.1. The upper and lower half of the furnace are separated by the slab row. The furnace enclosure is partitioned into N_z zones, this partition serves as a discretization in furnace length. The slab i_{last} is the last loaded slab into furnace, and the slab i_{first} will be the next slab discharged from the furnace.

In the frame reference of the furnace, the center of a slab i has the current *z-position* z_i as indicated in Figure 3.2. In general, slabs are moved in positive *z-direction* which means that z_i is a monotonously non-decreasing

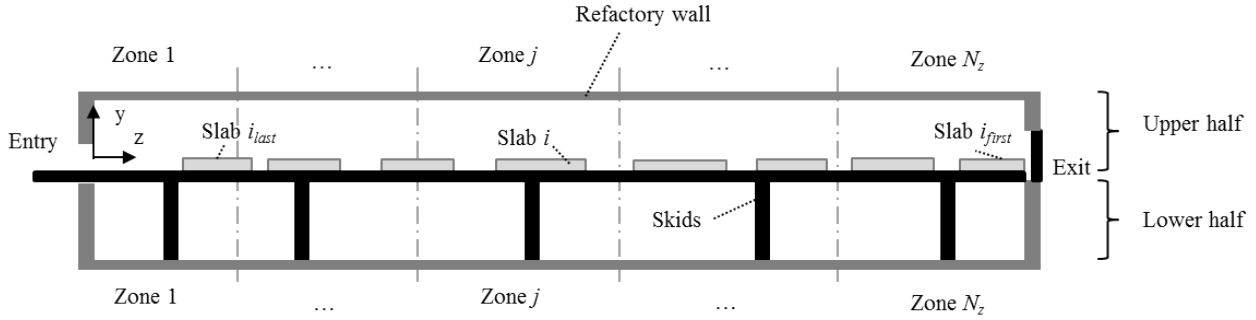

Figure 3.1: Simplified geometry of walking beam reheating furnace

function of time t. In contrast with pusher-type furnaces where slabs can only be moved forward; walking-beam furnaces allow backward motions to create gaps between slabs.

3.2.1 Unsteady one-dimensional heat conduction inside slabs

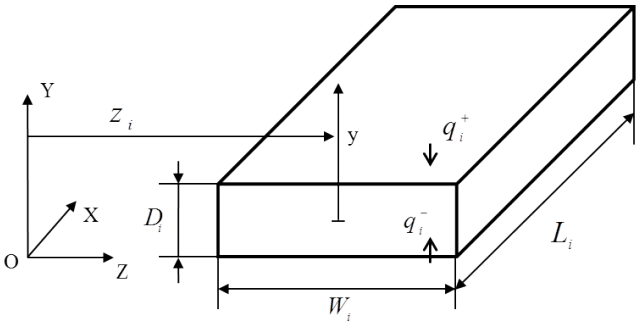

Figure 3.2: Geometry and position of slab *i*

This section addresses the heat conduction problem for a single slab i. Let $[x, y, z]$ be the coordinates of a slab i associated with the spatial reference of the furnace, as shown in Figure 3.1 and Figure 3.2. In addition, $T_i([x, y, z], t) > 0$ is the absolute temperature profile of slab i. The terms $q_i^+(t)$ and $q_i^-(t)$ are heat fluxes received by the upper and lower surface of the furnace, respectively. Heat flux $q_i([x, y, z], t)(W.m^{-2})$ inside the slab depends on specific heat capacity $c_i(J.kg^{-1}.K^{-1})$, thermal conductivity $\lambda_i(W.m^{-1}K^{-1})$, temperature gradient, and boundary conditions at the surface of the slab. The parameters of a steel grade c_i and λ_i are usually temperature-dependent variables.

Heat conduction problem inside slab follows *Fourier's law*. Let $\bigtriangledown$ be the gradient operator with respect to the spatial coordinates $[x, y, z]$. Knowing that heat flux is a vector quantity, the conduction rate equation from *Fourier*'s law is expressed as follows:

$$q_i([x, y, z], t) = -\lambda_i \bigtriangledown T_i \tag{3.1a}$$

$$= -\lambda_i (u_x \frac{\partial T_i}{\partial x} + u_y \frac{\partial T_i}{\partial y} + u_z \frac{\partial T_i}{\partial z}), \tag{3.1b}$$

with u_x, u_y, u_z are unit vectors of three axes OX, OY, OZ, respectively. The thermal conduction inside slab is therefore described by *heat diffusion law* as presented in [Inc11]:

$$\rho_i c_i \frac{\partial T_i}{\partial t} = - \bigtriangledown q_i \tag{3.2a}$$

$$= \bigtriangledown [\lambda_i \bigtriangledown T_i], \tag{3.2b}$$

where ρ_i represents the mass density of the slab material.

Since the slab dimension ratios (cf.Figure 3.2) are frequently characterized by $L_i \gg D_i$ and $W_i \gg D_i$, therefore $T_i([x, y, z], t)$ is reasonably independent from both x and z, and therefore can be approximated by $T_i(y, t)$ with $y \in [-D_i/2, D_i/2]$. Consequently, the heat flux q_i is considered as function of y and time t, and is determined by one-dimensional version of Fourier's law through the *y*-direction, which is expressed as follows:

$$q_i(y, t) = -\lambda_i \frac{\partial T_i(y, t)}{\partial y}. \tag{3.3}$$

In this cas, Equation 3.2 is simplified to:

$$\rho_i c_i \frac{\partial T_i}{\partial t} = \frac{\partial}{\partial y}(\lambda_i \frac{\partial T_i}{\partial y}). \tag{3.4}$$

The initial conditions concerns temperature profile of the slab when it is loaded into the furnace, which is described as:

$$T_i(y, t_{i,0}) = T_{i,0}(y), \tag{3.5}$$
$$y \in [-D_i/2, D_i/2]\,,$$

where $t_{i,0}$ is time when slab *i* is loaded into the furnace and $T_{i,0}(y)$ is temperature at position y inside the slab when it is charged into the furnace.

The boundary conditions of Equation 3.3 is given as:

$$q_i^+(t) = -\lambda_i \frac{\partial T_i}{\partial y} \mid_{y=D_i/2}, \tag{3.6a}$$

$$q_i^-(t) = -\lambda_i \frac{\partial T_i}{\partial y} \mid_{y=-D_i/2}\,. \tag{3.6b}$$

In summary, one-dimensional heat conduction problem inside slab i is now defined by one-dimensional heat diffusion law given in Equation 3.4, initial condition in Equation 3.6, and boundary condition described in Equation 3.6.

3.2.1.1 Kirchhoff transformation

In practice, specific heat capacity c_i is assumed to be independent of temperature. However, the dependence of thermal conductivity λ_i on temperature has to be considered. This leads to non-linearity of Equation 3.4 which can be solved by *Kirchhoff transformation* as described in the following:

We first introduce the Kirchhoff variable

$$\theta(T_i) = \int_0^{T_i} \lambda_i(T) dT. \tag{3.7}$$

It should be noted that if $\lambda_i(T)$ is specified, Equation 3.7 gives the relationship between θ and T_i. To transform Equation 3.4 in terms of the variable θ, $d\theta/dT_i$ is obtained by:

$$\frac{d\theta}{dT_i} = \lambda_i \tag{3.8}$$

And $\partial T_i/\partial t$ and $\partial T_i/\partial y$ are then constructed as:

$$\frac{\partial T_i}{\partial t} = \frac{dT_i}{d\theta}\frac{\partial \theta}{\partial t} = \frac{1}{\lambda_i}\frac{\partial \theta}{\partial t}, \tag{3.9}$$

and

$$\frac{\partial T_i}{\partial y} = \frac{dT_i}{d\theta}\frac{\partial \theta}{\partial y} = \frac{1}{\lambda_i}\frac{\partial \theta}{\partial y}. \tag{3.10}$$

Substituting Equations 3.9 and 3.10 into Equation 3.4, we have

$$\frac{\partial^2 \theta}{\partial y^2} = \frac{1}{\alpha_i}\frac{\partial \theta}{\partial t}, \tag{3.11}$$

where α_i is the thermal diffusivity of slab i, defined as

$$\alpha_i = \frac{\lambda_i}{\rho_i c_i}. \tag{3.12}$$

Since ρ_i and c_i are temperature-dependent, the diffusivity α_i is also a function of temperature. It shall be noted that in steady heat conduction problem, $\partial\theta/\partial t = 0$. Then, Equation 3.11 becomes linear. To continue the development, we introduce the temperature-dependent enthalpy $H(T_i)$ which is defined by $\partial H/\partial t = c_i \partial T_i/\partial t$ into Equation 3.4, then we have

$$\rho_i \frac{\partial H}{\partial t} = \frac{\partial(\lambda_i(\partial T_i/\partial y)}{\partial y}. \tag{3.13}$$

In order complete the Kirrchhoff transformation, the boundary conditions in Equation 3.6 and enthalpy in Equation 3.13 are also expressed in term of θ_i. By substituting Equation 3.9 into Equation 3.6 and Equation 3.10 into

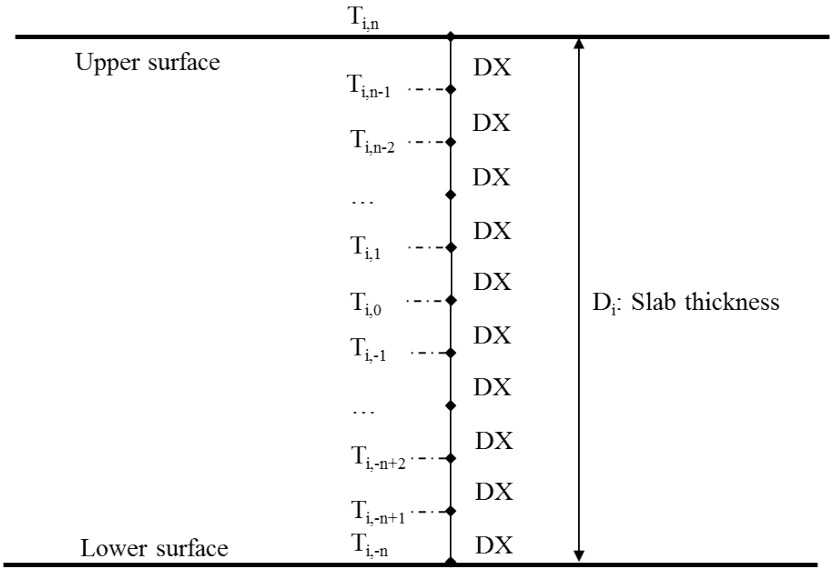

Figure 3.3: Temperature profile of $2n+1$ points inside a slab

Equation 3.13, we have:

$$q_i^+(t) = \frac{\partial \theta}{\partial y} \mid_{y=D_i/2}, \tag{3.14}$$
$$q_i^-(t) = \frac{\partial \theta}{\partial y} \mid_{y=-D_i/2},$$

and

$$\rho_i \frac{\partial H}{\partial t} = \bigtriangledown^2 \theta. \tag{3.15}$$

3.2.1.2 Finite difference method

The Equations 3.14 and 3.15 constitute the problem of one-dimensional heat conduction inside a slab of continuous-time and continuous space domains. However, a discrete representation is required because the model has to be implemented on a computer. The discretization is needed for both time and space domain. For instance, temperatures at $2n+1$ points through slab thickness will be computed. First, the sampling distance is $\triangle\ y = DX = D_i/2n$, and as shown in Figure 3.3, the temperature at position $y_j = j \times \triangle\ y$ inside the slab i will be denoted $T_{i,j}$. Sampling instants $t_k, \forall k \in N$ are discrete time, and $\triangle\ t_k = t_{k+1} - t_k$ defines the corresponding sampling period. The main assumptions are:

- At a fixed position z_i of slab, the heat flow input $q_i^{\pm}(t)$ are constant within the period $t \in [t_k, t_{k+1}[$
- Conductivity of steel is *piesewise* linear between two temperatures $T_{i,j}(t_k)$ and $T_{i,j}(t_{k+1})$
- Kirchhoff variables are constant with $t \in [t_k, t_{k+1}[$

Using first order central difference and Taylor series expansions [Jij09], we have

$$\frac{\partial\theta}{\partial y}\mid_{y_j} = \frac{\theta_{j+1}-\theta_{j-1}}{2\triangle y} + \mathcal{O}(\triangle y), \tag{3.16}$$

where θ_j represents the Kirchhoff variable corresponding to temperature $T_{i,j}$. Similarly, second order difference is expressed as:

$$\frac{\partial^2\theta}{\partial y^2}\mid_{y_j} = \frac{\theta_{j+1}+\theta_{j-1}-2\theta_j}{(\triangle y)^2} + \mathcal{O}((\triangle y)^2), \tag{3.17}$$

when number of points is sufficiently large, i.e, $\triangle y$ is significantly small, $\mathcal{O}((\triangle y)^2)$ can be neglected. Therefore, from Equation 3.17, the problem defined by Equations 3.14 and 3.15 is rewritten as

$$\rho_i \frac{H_j(t_{k+1}) - H_j(t_k)}{\triangle\, t_k} = \frac{(\theta_{j+1}-\theta_j)-(\theta_j-\theta_{j-1})}{(\triangle\, y)^2}, \tag{3.18}$$

where $j \in [-n+1, n-1]$ indicates the position inside slab and H_j represents the enthalpy corresponding to the temperature $T_{i,j}$. Thus, the evolution of enthalpy H_j is defined by as follows:

$$H_j(t_{k+1}) = H_j(t_k) + \frac{\triangle\, t_k\,(\theta_{j+1}+\theta_{j-1}-2\theta_j)}{\rho_i(\triangle\, y)^2}. \tag{3.19}$$

To compute boundary values of H_j with $j = \pm n$, from Equation 3.16, Equation 3.19 is rewritten as

$$H_j(t_{k+1}) = H_j(t_k) + \frac{\triangle t_k\,(2(\theta_{j+1}-\theta_j)-(\theta_{j+1}-\theta_{j-1}))}{\rho_i(\triangle y)^2} \tag{3.20a}$$

$$= H_j(t_k) + \frac{2\triangle t_k\left((\theta_{j+1}-\theta_j) - \triangle y\frac{\partial\theta_i}{\partial y}\mid_{y_j}\right)}{\rho_i(\triangle y)^2}. \tag{3.20b}$$

Substituting Equation 3.14 into Equation 3.20

$$H_{-n}(t_{k+1}) = H_{-n}(t_k) + \frac{2\triangle t_k\left((\theta_{-n+1}-\theta_{-n}) - \triangle y q_i^-(t_k)\right)}{\rho_i(\triangle y)^2}, \tag{3.21a}$$

$$H_n(t_{k+1}) = H_n(t_k) + \frac{2\triangle t_k\left((\theta_{n+1}-\theta_n) - \triangle y q_i^+(t_k)\right)}{\rho_i(\triangle y)^2}. \tag{3.21b}$$

At each time instant t_k, temperature $T_{i,j}(t_k)$, *Kirchhoff* variables $\theta_j(t_k)$ and corresponding enthalpy values $H_j(t_k)$ are computed by help of *Bisra* tables which is given in Appendix A. In a given a sampling time $\triangle t_k$, the evolution of enthalpy $H_j(t_k)$ is defined with Equations 3.19 and 3.21. Then, new temperature profile and *Kirchhoff* variables are inversely calculated by *Bisra* table.

The heat flows received by slab surfaces $q_i^{\pm}(t_k)$ are subsequently determined by steady-state radiative heat transfer model based on position of slabs inside furnace.

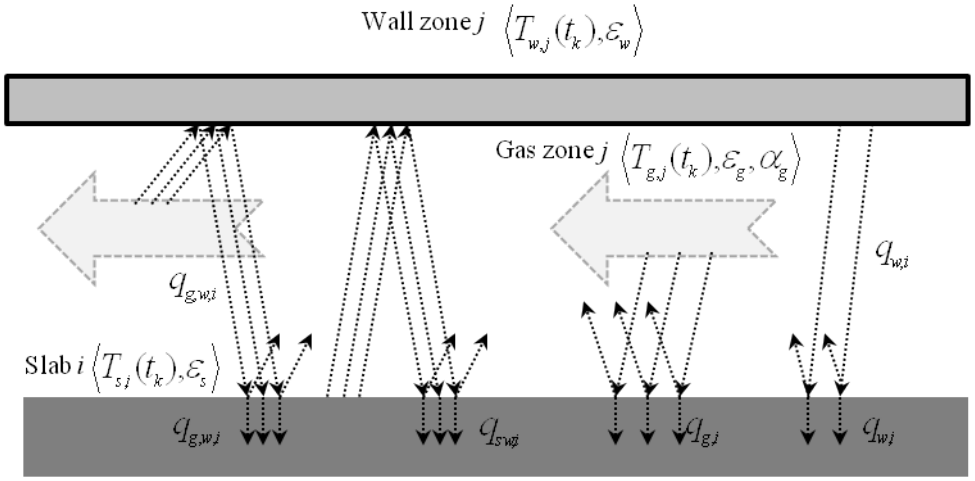

Figure 3.4: Radiative heat exchange between slabs and environment

3.2.2 Steady-state radiative heat transfer in the furnace

The heat fluxes $q_i^{\pm}(t_k)$ received by surfaces of slab i introduced in the previous section as boundary conditions are necessary to compute temperature profile of the slab, and they represent radiative thermal fluxes inside the furnace. This section presents a simple method for calculation of $q_i^{\pm}(t_k)$. In general, radiative heat transfer couples the dynamic of all furnace zones, and all slabs inside the furnace because radiation is, in contrast to heat conduction, not just a local phenomenon. Inside the furnace, the heat exchange through radiation is determined by:

- The areas, shape, and position of the participating surfaces such as: walls, roof, skids, slabs
- Temperature of surfaces that participate to the radiation
- Emissive, absorptive, and reflective properties of the surfaces
- The transmissive properties of the medium residing between the surfaces, which is, for instance, the exhaust gas circulating inside the furnace.

The general problem of radiative heat transfer is described with fundamental concepts. After that, the assumptions for steady-state operation of the furnace will be given. The radiative heat transfer is then computed based on the simplifying assumptions.

3.2.2.1 Assumptions

- Thermal radiation is the only considered mode of heat exchange between the slabs and their surrounding as shown in Figure 3.4. Moreover, in high temperature environment of the furnace, the dominant mode of heat exchange is radiation because radiative heat transfer is proportional to the difference of the 4^{th}

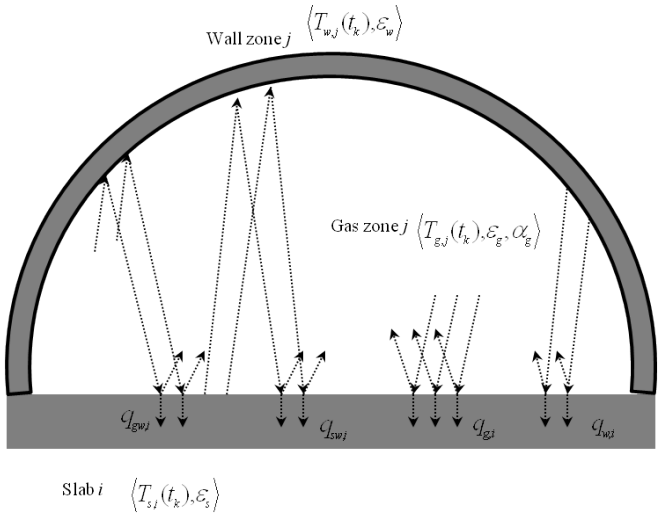

Figure 3.5: Assumption of local radiation for a slab i residing in zone j

powers of the temperatures whereas most other types of heat exchange depend only on the difference or the gradient of the temperatures themselves. Therefore, heat conduction between the slabs and the skids, or convection between the slabs and the flue gas, are considered negligible. However, heat convection will be examined in the next section.

- Only the bottom and the top surface of a slab serve as interfaces for heat exchange between the slab and its environment. Lateral, front, and back faces of the slab do not participate in the radiative heat exchange process. This assumption is inline with one-dimensional heat conduction inside the slabs as considered in Section 3.2.1

- For the computation of radiative heat exchange, the thickness of the slabs is not taken into account, i.e., all top surfaces of the slabs are assumed to be at the same level. The heat fluxes received by two surfaces are assumed to be equal: $q_i^+(t_k) = q_i^-(t_k) = q_i(t_k)$

- At any time, the temperature of a participating surface is homogeneously distributed. This assumption seems adequate for slab surfaces. For furnace wall surface, however, its validity depends largely on the surface size, the distribution of burners, and the gaseous flow inside the furnace.

- All participating surfaces are ideal diffuse reflectors which means that an incoming ray of light is equally reflected in any direction over the hemisphere covering the surface.

- The furnace is divided into zones. It is assumed that the temperature and properties of the gas are homogeneous within each volume zone.

- When considering slab i residing in zone j, furnace wall and slab are combined into a closed space filled with flue gas as shown in Figure 3.5. That is all radiation exchanges are between furnace wall, gas of zone j, and slab i.

3.2.2.2 Computation of heat radiation

Let $T_{g,j}$ with $\forall j \in 1, 2, ..., N_z$ be the local gas temperatures which are usually measured by thermocouples, $T_{s,i}$ and $T_{w,j}$ are respectively the surface temperature of slab i and wall temperature of zone j. These temperatures are homogeneous as indicated with the assumptions. Radiative fluxes absorbed by slabs come mainly from furnace walls, and gas. Between slab surface and walls of the corresponding zone, the radiative heat flows are absorbed by the flue gas, continuously reflected and absorbed by solid surfaces. Radiative properties of walls and slabs are respectively characterized by their emissivities $\varepsilon_w, \varepsilon_s$, reflectivities $\tau_w = 1 - \varepsilon_w, \tau_s = 1 - \varepsilon_s$, absorbtivities $\alpha_w = \varepsilon_w, \alpha_s = \varepsilon_s$. Flue gas radiative characteristics are defined by its emissivity ε_g and its absorbtivity α_g. For instance, four radiative fluxes have to be considered in order to compute radiative heat flux absorbed by the slab as shown in Figure 3.4

- Heat flows received by slab i from wall: $q_{w,i}$
- Heat radiation emitted by the gas to wall, reflected by the wall and finally absorbed by the slab i: $q_{gw,i}$
- Heat flows emitted from the slab to wall and reflected back to the slab i: $q_{sw,i}$
- Heat radiation emitted to slab i from flue gas: $q_{g,i}$

The total heat flux absorbed by surfaces of a slabs is computed as

$$q_i^+ = q_i^- = q_i = q_{g,i} + q_{w,i} + q_{gw,i} + q_{sw,i}. \tag{3.22}$$

Notice that the subscripts w, g, s stand for wall, gas and slab, respectively.

Radiation from furnace walls absorbed by a slab.
The radiation of furnace walls may be described by the *Stefan-Boltzmann law* applied for a grey body

$$q_w = \varepsilon_w \sigma T_{w,j}^4. \tag{3.23}$$

where $\varepsilon_w \in [0, 1]$ is the emissivity of furnace walls as previously mentioned, σ is the *Stefan-Boltzmann constant*. In order to reach slab surface, radiative fluxes from furnace walls have to travel through the gas. Flue gas will absorb a portion of radiative energy, which is characterized by absorbtivity coefficient α_g. Once the radiation has reached slab surface, it is generally reflected back through the gas and then comes back to furnace walls. The radiation reflection continues until its energy is attenuated. The reflectivity τ_s on slab surface is characterized by its emissivity ε_s by the relation $\tau_s + \varepsilon_s = 1$. The heat flux absorbed by slab from the initial radiation (indicated by the subscript 1) is computed as

$$q_{w,i,1} = \varepsilon_s(1 - \alpha_g)q_{w,0} = \varepsilon_s A_w, \tag{3.24}$$

with $A_w = (1 - \alpha_g)q_{w,0}$. Then, the reflection of the radiation to furnace walls through the gas is determined as

$$q_{w,i,1}^* = (1 - \varepsilon_s)A_w. \tag{3.25}$$

The reflective radiation is then absorbed partially by furnace walls and then reflected back to slab surface as

$$q^{**}_{w,i,1} = (1-\varepsilon_w)(1-\alpha_g)^2 q^{*}_{w,i,1} = (1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2 A_w \tag{3.26}$$

The radiative heat absorbed by slab surface from the second radiation is calculated as

$$q_{w,i,2} = \varepsilon_s q^{**}_{w,i,1} = \varepsilon_s (1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2 A_w. \tag{3.27}$$

From Equations 3.24 and 3.27 , we can generalize the heat radiation of wall absorbed by the slab surfaces after n^{th} time reflected back to slab surface as

$$q_{w,i,n} = \varepsilon_s \left[(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2\right]^{n-1} A_w. \tag{3.28}$$

The total radiative heat flux received by slab surface originated from furnace walls is

$$q_{w,i} = \sum_{k=1}^{n} q_{w,i,k} = \varepsilon_s A_w \left(1 + a + a^2 + ... + a^{n-1}\right) = \varepsilon_s A_w \frac{1-a^n}{1-a} \tag{3.29}$$

where $a = (1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2$; and it is obvious that $a < 1$. Therefore, when radiation reflects many time, i.e.,$n \to \infty$, $a^n \to 0$. Then, Equation 3.29 can be simplified as

$$q_{w,i} = \varepsilon_s A_w \frac{1}{1-a} = \frac{\varepsilon_s (1-\alpha_g)\varepsilon_w \sigma T^4_{w,j}}{1-(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2}. \tag{3.30}$$

Direct radiation from gas to slab surface.

The gas is assumed to radiate directly on the slab surface like a normal solid object. Consequently, the initial heat flux coming from gas absorbed by the slab is computed as

$$q_{g,i,1} = \varepsilon_s \varepsilon_g \sigma T^4_{g,j}. \tag{3.31}$$

Following the Equation 3.28, the energy from the consequent reflective radiation on furnace walls after k time is computed as

$$q_{g,i,k} = \varepsilon_s \left[(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2\right]^{k-1} A_g \tag{3.32}$$

where $A_g = \varepsilon_g \sigma T^4_{g,j}$.

Similar to the Equations 3.30, the sum of direct radiation flux from gas to slab surface and its reflections can be calculated as

$$q_{g,i} = \sum_{k=1}^{n} q_{g,i,k} = \varepsilon_s A_g \frac{1}{1-a} = \frac{\varepsilon_s \varepsilon_g \sigma T^4_{g,j}}{1-(1-\varepsilon_w)(1-\varepsilon_s)(1-\alpha_g)^2}. \tag{3.33}$$

Radiation from gas to furnace walls reflected to slab surface.

The direct radiation reflected by furnace walls from gas, and comes to slab surface is computed as

$$A_{gw,i} = (1 - \alpha_g)(1 - \varepsilon_w)\varepsilon_g \sigma T_{g,j}^4. \tag{3.34}$$

Heat flux is then absorbed by the slab as the given formula

$$q_{gw,i,1} = \varepsilon_s A_{g,w,i}. \tag{3.35}$$

Radiative heat flux after k reflections

$$q_{gw,i,k} = \varepsilon_s \left[(1 - \varepsilon_w)(1 - \varepsilon_s)(1 - \alpha_g)^2\right]^{k-1} A_{gw,i}. \tag{3.36}$$

The total radiative thermal flux absorbed by slab surfaces in this case is constructed as

$$q_{gw,i} = \varepsilon_s A_{gw,i} \frac{1}{1 - a} = \frac{\varepsilon_s (1 - \alpha_g)(1 - \varepsilon_w)\varepsilon_g \sigma T_{g,j}^4}{1 - (1 - \varepsilon_w)(1 - \varepsilon_s)(1 - \alpha_g)^2}. \tag{3.37}$$

Radiation issued from the slab, reflected on furnace walls, then back to slab surface.

The radiation flux arriving at furnace walls from slab surface across gas is constructed as

$$q_{i,w} = (1 - \alpha_g)\varepsilon_s \sigma T_{s,i}^4. \tag{3.38}$$

The radiation is then reflected by the walls, and partially absorbed by gas as:

$$q_{sw,i} = (1 - \varepsilon_w)(1 - \alpha_g)^2 \sigma T_{s,i}^4 = A_{sws}. \tag{3.39}$$

Absorption by the slab of this first reflected radiation is calculated by

$$q_{s,i,1} = \varepsilon_s A_{sws}. \tag{3.40}$$

The heat flux absorbed by slab surface after k time reflected on furnace walls is given as

$$q_{s,i,k} = \varepsilon_s \left[(1 - \varepsilon_w)(1 - \varepsilon_s)(1 - \alpha_g)^2\right]^{k-1} A_{s,w,s}. \tag{3.41}$$

Total heat flux absorbed by slab surface issued from its own radiation

$$q_{sw,i} = \varepsilon_s A_{sw,i} \frac{1}{1 - a} = \frac{\varepsilon_s (1 - \varepsilon_w)(1 - \alpha_g)^2 \sigma T_{s,i}^4}{1 - (1 - \varepsilon_w)(1 - \varepsilon_s)(1 - \alpha_g)^2}. \tag{3.42}$$

In summary, solution of radiative heat transfer is determined by Equations 3.30, 3.33, 3.37, 3.42.

3.3 Dynamical model

Nowadays, furnace production has to respond rapidly to the client demand. The rolling mill needs a diversity of products in terms of steel grade, dimensions, desired rolling temperature. The steady-state model cannot provide accurate temperature estimations for furnace control. Consequently, safety margins of slab temperatures are used to avoid under-heated slabs. The strategy is rather conservative, and it leads to general over-heating of slabs, especially during stops of the furnace as shown in Figure 3.6. Energy cost reduction of reheating furnace

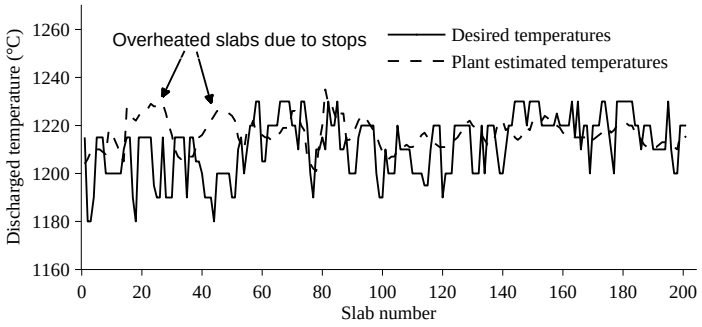

Figure 3.6: Overheating of slab due to stops; waste of energy

significantly enhances the competitivity of the final product. New product with special characteristic requires more and more strict control on heating quality of the furnace. Therefore, the furnace operators would want to have a flexibility on production rate, being able to integrate an optimization of energy consumption and at the same time improve the heating process. For these reasons, furnace model needs to be able to monitor and predict gas, wall, slab temperature, energy consumption, etc., by taking into account available process variables of furnace conditions such as excess air, input power to each zone, or heat loss to skids.

In the following a dynamical model of the furnace is described

3.3.1 Heat conduction

In Section 3.2.1, one-dimensional heat conduction problem has been defined by Equations 3.4 and 3.6. The problem is then solved by introducing enthalpy variable specified by Equation 3.13, utilizing finite difference method and Kirchhoff transformation. In dynamical model of the furnace, finite volume method is used to construct one-dimensional heat conduction model inside slabs. Consider again the unsteady one-dimensional heat conduction equation for a slab i assuming that specific heat capacity c_i is temperature independent, we have

$$\rho_i c_i \frac{\partial T_i}{\partial t} = \frac{\partial}{\partial y}\left(\lambda_i(T_i)\frac{\partial T_i}{\partial y}\right). \tag{3.43}$$

To find discretized equation, we use the grid-point cluster shown in Figure 3.7. We focus on the grid point P, which has grid points E and W as its neighbors. The dashed lines show the faces of a so-called control volume;

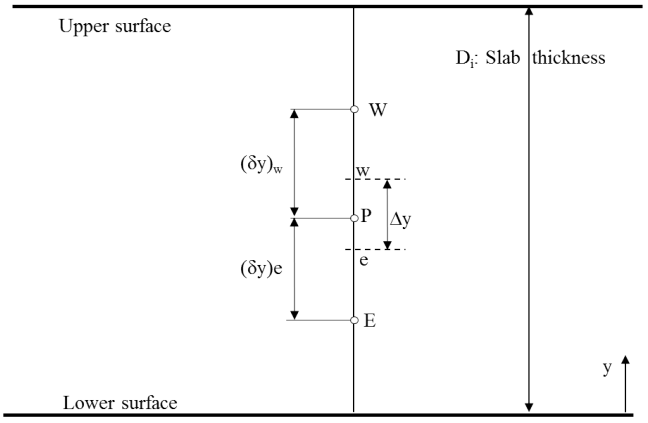

Figure 3.7: Grid-point discretization for one-dimensional heat conduction problem

their exact positions are not of interest for the time being. The letters e and w denotes these faces. The control volume has unit thickness in the z and x directions. Therefore, volume of the control volume shown is $\Delta y \times 1 \times 1$.

Temperature profile of a slab at each time instant is defined by Equation 3.43 and its initial distribution of temperature. Thus, in a time step Δt, the problem to be solved is: Given the grid-point values of T at time t, find the values of T at time $t + \Delta t$. The given values of T at time t are denoted by T_P^0, T_E^0, T_W^0, and the new values at time $t + \Delta t$ by T_P^1, T_E^1, T_W^1. Now, discretization equation is described by integrating Equation 3.43 over the control volume shown in Figure 3.7, and over the interval from t to $t + \Delta t$. Thereby,

$$\rho_i c_i \int_w^e \int_t^{t+\Delta t} \frac{\partial T_i}{\partial t} dt dy = \int_t^{t+\Delta t} \int_w^e \frac{\partial}{\partial y} \left(\lambda_i(T_i) \frac{\partial T_i}{\partial y} \right) dy dt, \tag{3.44}$$

where the order of the integration is chosen according to the nature of the term. Assuming that the whole control volume has the same temperature with the grid point P, then $\partial T/\partial t = dT/dt$. Then,

$$\rho_i c_i \int_w^e \int_t^{t+\Delta t} \frac{\partial T_i}{\partial t} dt dy = \rho_i c_i \Delta y (T_P^1 - T_P^0). \tag{3.45}$$

At each time instant, heat conduction inside slab is considered as in steady-state condition, that is $\partial T/\partial y = dT/dy$; therefore,

$$\rho_i c_i \Delta y (T_P^1 - T_P^0) = \int_t^{t+\Delta t} \left[\frac{\lambda_e (T_E - T_P)}{(\delta y)_e} - \frac{\lambda_w (T_P - T_W)}{(\delta y)_w} \right] dt. \tag{3.46}$$

At this point, we need to make assumption about how T_P, T_E, T_W evolve with time from t to $t + \Delta t$. Many choices are possible, some of them can be given by the following formula

$$\int_t^{t+\Delta t} T_P dt = \left[f T_P^1 + (1 - f) T_P^0 \right] \Delta t, \tag{3.47}$$

where f is a weighting factor in $[0, 1]$. By assuming the same formulas for the integrals of T_E and T_W, Equation 3.46 becomes

$$\rho_i c_i \frac{\Delta y}{\Delta t}(T_P^1 - T_P^0) = f\left[\frac{\lambda_e(T_E^1 - T_P^1)}{(\delta y)_e} - \frac{\lambda_w(T_P^1 - T_W^1)}{(\delta y)_w}\right] + (1-f)\left[\frac{\lambda_e(T_E^0 - T_P^0)}{(\delta y)_e} - \frac{\lambda_w(T_P^0 - T_W^0)}{(\delta y)_w}\right] \quad (3.48)$$

To obtain a more general form, we shall drop the superscript 1, and remember that T_P, T_E, T_W from now on stand for the new values of T at time $t + \Delta t$. The resulting equation is

$$a_P T_P = a_E\left[fT_E + (1-f)T_E^0\right] + a_W\left[fT_W S + (1-f)T_W^0\right] + \left[a_P^0 - (1-f)a_E - (1-f)a_W\right]T_P^0, \quad (3.49)$$

where

$$a_E = \frac{\lambda_e}{(\delta y)_e}, \quad (3.50a)$$

$$a_W = \frac{\lambda_w}{(\delta y)_w}, \quad (3.50b)$$

$$a_P^0 = \frac{\rho_i c_i \Delta y}{\Delta t}, \quad (3.50c)$$

$$a_P = f a_E + f a_W + a_P^0. \quad (3.50d)$$

The Equations 3.49 and 3.50 constitutes the discritization equation of one-dimensional conduction inside a slab i. At each time instant, temperature profile of a slab is determined if the temperature of lower and upper surfaces is known.

Now, an important question remains, that is: how the weighting factor f is chosen so that the discretization equation is physically realistic. The answer is $f = 1$ as well investigated in [Pat80]. As written in the Equation 3.47, f indicates how temperature evolve from T_P^0 at time t to T_P^1 at time $t + \Delta t$. In Figure 3.8, explicit scheme corresponds to $f = 0$, that is the old value T_P^0 endures throughout the entire time, except at time $t + \Delta t$. Crank-Nicolson scheme defined by $f = 0.5$ assumes the linear variation of T_P. Fully implicit scheme determined by $f = 1$ supposes that, at time t, T_P suddenly jumps from T_P^0 to T_P^1 then stays at T_P^1 over the whole time step. As reasoned in [Pat80], most realistic situations will be such that temperature at one point in the grid is influenced by the temperature of neighboring points only through process of conduction and diffusion. Then, decrease of temperature at one point, without changing of other conditions, leads to decrease (and not increase) of temperature at neighboring points. Applying this reasoning to the discretization equation 3.49, if T_P decreases then T_W and T_E also decrease. It implies that the coefficients a_P,a_W, and a_E must have the same sign, and for that only $f = 1$ is satisfactory.

3.3.2 Radiative heat transfer

The furnace enclosure is considered as consisting of n grey and diffusing surface S_i with emissivity ε_i and homogeneous temperature T_i as shown in Figure 3.9. The gas has a uniform temperature T_g. These surfaces

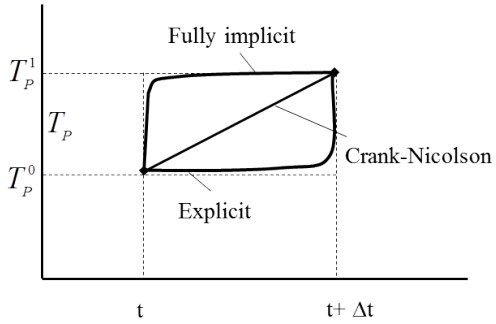

Figure 3.8: Variation of temperature with time for three different schemes

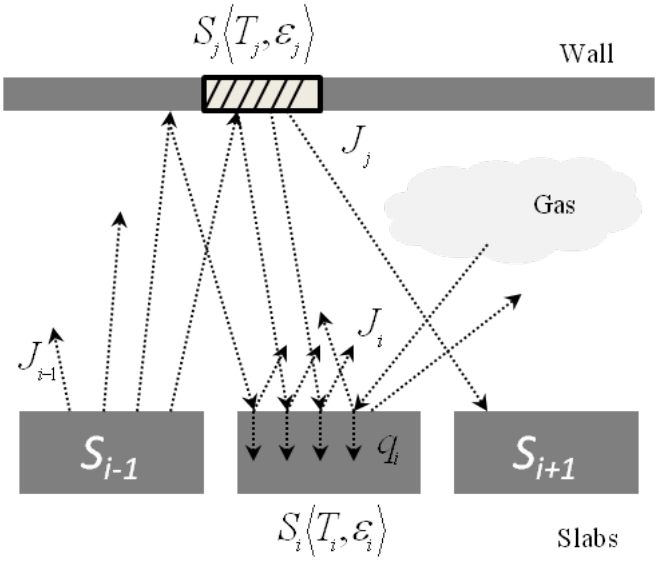

Figure 3.9: Radiation exchange between different surfaces

can be slab surfaces, wall surfaces, roof surfaces, etc,. The radiosity of the surface S_i is the total heat flux coming out from S_i. A part of the radiosity of surface S_i is its own radiative emission $S_i\varepsilon_i\sigma T_i^4$. A second part is the radiative fluxes transmitted through furnace gas, reflected by S_i coming from all other surfaces of furnace enclosure including S_i if it is concave. The final part is the heat flux of furnace gas emitted to S_i and reflected on S_i with a reflectivity τ_i. The total radiative flux leaving S_i is therefore given as follows:

$$S_i J_i = S_i\varepsilon_i\sigma T_i^4 + \tau_i \sum_{j=1}^{n} S_j F_{j,i}\beta_{j,i} J_j + \tau_i S_i \varepsilon_{g,i}\sigma T_g^4. \tag{3.51}$$

In this expression: $\beta_{j,i}$ is the total transmissivity of the gas separating surfaces S_j and S_i, $F_{i,j}$ and $F_{j,i}$ are the view factors between two surfaces, J_i is the radiative flux of surface S_i, ε_g, i is the total emissivity of the gas to S_i.

Knowing that $S_j F_{j,i} = S_i F_{i,j}$, also $\tau_i = 1 - \varepsilon_i$, and dividing both side of Equation 3.51 by S_i we obtain the following equation:

$$J_i = \varepsilon_i\sigma T_i^4 + (1-\varepsilon_i) \sum_{j=1}^{n} F_{i,j}\beta_{j,i} J_j + (1-\varepsilon_i)\varepsilon_{g,i}\sigma T_g^4. \tag{3.52}$$

Using Kronecker operator: $\delta_{i,j} = 1$ if $i = j$; $\delta_{i,j} = 0$ if $i \neq j$, we have the following form of Equation 3.52:

$$\sum_{j=1}^{n}[\delta_{i,j} - (1-\varepsilon_i)F_{i,j}\beta_{j,i}]J_j = \varepsilon_i\sigma T_i^4 + (1-\varepsilon_i)\varepsilon_{g,i}\sigma T_g^4. \tag{3.53}$$

This equation is only for the surfaces with imposed temperatures. For those with imposed flux, we have to establish another equation by using the net flux leaving S_i. This latter net flux is equal to total flux leaving S_i reduced with flux arriving on this surface, as given in the following equation:

$$q_i = J_i - \sum_{j=1}^{n} F_{i,j}\beta_{j,i}J_j + (1-\varepsilon_i)\varepsilon_{g,i}\sigma T_g^4.$$

Substituting with Equation 3.53, we have the following form:

$$\sum_{j=1}^{n}[\delta_{i,j} - F_{i,j}\beta_{j,i}]J_j = q_{net,i} + \varepsilon_{g,i}\sigma T_g^4. \tag{3.54}$$

The radiative heat transfer inside furnace is then defined by the following system of equations:

$\sum_{j=1}^{n} A_{i,j}J_j = B_i$, for $i = 1:n$		
S_i	imposed with temperature T_i	imposed with net flux $q_{net,i}$
$A_{i,j} =$	$\delta_{i,j} - (1-\varepsilon_i)F_{i,j}\beta_{j,i}$	$\delta_{i,j} - F_{i,j}\beta_{j,i}$
$B_i =$	$\varepsilon_i\sigma T_i^4 + (1-\varepsilon_i)\varepsilon_{g,i}\sigma T_g^4$	$q_i + \varepsilon_{g,i}\sigma T_g^4$

Table 3.1: System of equations for radiative heat transfer inside reheating furnace

The resolution of system of equations in Table 3.1 provides the radiosity J_i of a surface S_i. The calculation of the unknown temperature T_i or unknown net flux q_i is defined by the following relation:

$$q_i = \frac{\varepsilon_i}{1-\varepsilon_i}(\sigma T_i^4 - J_i) \tag{3.55}$$

For the furnace, different surface is defined based on division into zones. All slab surfaces participate to the overall heat radiation inside the furnace. In this dynamical model, local assumptions are eliminated.

3.3.3 Convective heat transfer

In continuous reheating furnace, typically less than 5% of the total heat input into the slabs comes from heat convection. This may explain why convection can be neglected in most of the models. However, if a model needs to be refined, convection is surely to be considered.

Fundamental differences between convection and radiation transfer which is dominant mode of heat exchange in a slab reheating furnace are:

- Convection can only occur between the exhaust gas and an adjacent surface of slabs, furnace walls/roof.
- Heat convection is due to the gas motion which may be generated by external source like pumps, fans, suction devices, or only by the density differences in the gas occurring due to temperature gradients.
- Convection is a local phenomenon, as opposed to radiation which transfers heat over large distances in the furnace

Typically, the convective heat flux is modeled with linear dependence on the temperature difference. Thus, the convective heat flux into the surface of slab i which resides currently in zone $j \in 1, ..., N_z$ is computed as

$$q^{\pm}_{c,i}(t) = h^{\pm}_{c,j} \left(T^{\pm}_{g,j}(t) - T_{s,i}(\pm D_j/2, t) \right) \tag{3.56}$$

where $T_{s,i}(\pm D_j/2, t), T^{\pm}_{g,j}(t)$ are slab surface temperatures, and gas temperature, respectively. This expression is often known as *Newton's law of cooling*. The convection heat transfer coefficient $h^{\pm}_{c,j}$ is generally constant for a zone and is proportional to velocity of gas. The convective heat fluxes $q^{\pm}_{c,i}(t)$ are only added to the radiative heat fluxes.

3.4 Discussion

Now, we discuss the differences between steady-state model and dynamical model. About heat conduction problem, steady-state model utilizes the initial derivative form of conduction Equation 3.4. Finite difference method assumes the polynomial form of Kirchhoff variable and uses Taylor series expansion to approximate the temperature at a number of nodes inside slab. As investigated in [BRW00], this method is not mass conservative because the boundary conditions are not included in the governing equation. However, using a large number of nodes for the temperature profile can solve this problem.

The dynamical model, on the other hand, utilizes a integral form of heat conduction problem, which is the basic idea of finite volume method defined by Equation 3.44. The integral form assures the conservation of mass for any number of nodes in temperature profile. Therefore, with a profile consisting of small number of nodes, the finite volume method is superior to finite difference method. In fact, furnace control only needs a representative temperature profile of slabs. Thus, temperature estimation at several points through slab thickness shall be satisfactory. This validates the finite volume method and therefore validates the new approach to conduction problem adopted in the dynamical model.

In steady-state model, it is assumed that radiation in a zone is a local phenomenon. Only the radiative heat exchanges between slabs, exhaust fume, and wall of the zone are considered. The dynamical model considers heat radiation in the furnace as a global phenomenon, therefore heat radiation from zone to zone and slabs to slabs are taken into account.

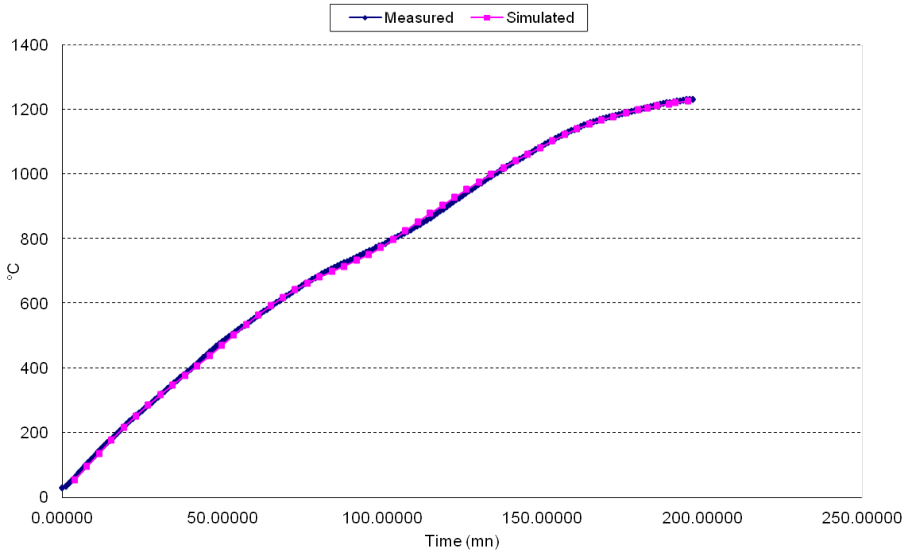

Figure 3.10: Averaged slab temperature from validation experiment of the dynamical furnace model

The developed dynamical model can also calculate the temperature profile of the furnace walls and take into account different material of layer of furnace walls, which is not the case of steady-state model. While the steady-state model can only use measured temperature of furnace zones as inputs, the dynamical model can use also measurement of gas flows to calculate the temperature of furnace walls and then temperature of slabs. On the other hand, when the dynamical model use zone temperatures as inputs to calculate slab temperatures, it is able to estimate the consumption of the furnace while the steady-state model can not. This is crucial because we want to optimize energy consumption of the furnace.

In practice, the developed dynamical model has been validated experimentally. In Figure 3.10, the measurement of slab average temperature trajectory is compared with simulated values. We see that the model is highly precise, which means it can be utilized to replace the measurement.

Part II

Control design

Chapter 4

Control system of walking beam reheating furnace

General structure of reheating furnace was introduced in Chapter 1 and Chapter 2. Chapter 3 has described the currently used heat transfer model of the considered furnace. A dynamical model which is able to deal with unsteady-state furnace operation mode was also presented. This chapter will firstly give an overview of past works on the control structure of reheating furnaces. Then, the existing control strategy of the considered furnace will be presented. Finally, basic ideas of proposed control method and structure will be clarified.

4.1 Literature review

Main performance indicators of reheating furnaces are energy consumption, production rate, accuracy of the controlled temperatures, and operational costs. These indicators are influenced by construction factors, characteristic of charge, operational factors and control system of the furnace. Construction factors can be listed as: furnace geometry, refractory of furnace enclosure, measurement system, e.g., thermocouples, construction of heat sinks and heat sources, e.g., cooling system of skids and fuel burners, composition of fuel and limitation on fuel flow rates. Charge factors concern initial temperatures of slabs, desired final temperatures of slabs, material properties, sequence, geometry, geometric arrangement of slab inside the furnace, and metallurgical constraints on the slab temperature trajectories. Operational and system control factors involve the following aspects:

- Control of furnace zone temperatures and slab temperatures
- Control of temperature gradients inside slabs
- Throughput rate and movements of slabs
- Air and fuel ratio control which assures the proper ratio between combustion air and fuel in the air/fuel mixture

- Scheduled or unscheduled downtime or delays, which may be caused by upstream or downstream processes
- Control of heat recuperation units
- Control of opening and closing of furnace doors
- Control of temperature, pressure, and flow rates of combustion air and fuel

In reheating furnace control, slab temperatures are the most relevant process variables. These variables are controlled by varying furnace zone temperatures. Subsequently, zone temperatures are controlled by varying the fuel flow to the burners of the zones. In a particular case, production rate can be regulated to control slab temperature while furnace zone temperatures are maintained constant [KMP94]. Therefore, it may be distinguished between controllers that regulate only temperatures or just the movement speed of slabs through the furnace [KMP94] or both [BBG08]. In this part of the work, like in most control strategies, only the temperature is controlled, the scheduling and so velocity of the slabs are forecast parameters. It shall be noted that *temperature control* can refer to slab temperature or zone temperature control or both.

Most furnace control strategies reported in the literature belong to one of the following strategies:

- Open-loop control
- State-feedback control
- Two-degree-of-freedom control
- Control strategies based on mathematical optimization

Open-loop control schemes are only suitable when furnace operations are stable or do not have considerable uncertainties. The references [ESYS97, CA08] report on open-loop control of slab temperature by adjusting zone temperatures, i. e., there is no feedback from slab temperatures whether they are measured or estimated. The set-points of zone temperatures are sent to subordinate control device, which regulates the supply of fuel and combustion air. The major drawback of this type of control is the lack of feedback [SWHP87]. In [HZ82], the presented control system directly defines reference signals for fuel flow rates, which are applied to the furnace by open-loop control. The furnace zone temperatures are measured only for safety reasons, e.g., to prevent slabs from overheating. Therefore, open-loop control of the slab temperatures is reasonable as long as the furnace zone temperatures are feedback controlled or at least monitored.

Feedback control strategies control directly slab temperatures by feedback as in [ESYS97, FPSF90, KMP94, Led86a, PW98, WWC04]. The current slab temperatures can be obtained either by measurement, e.g., radiation pyrometers, from estimation of an online simulator or a state observer. It shall be noted that only slab surface temperatures are accessible by means of pyrometers. From slab surface temperatures, the slab temperature profile can be further estimated by heat conduction models.

The control schemes presented in [WYY09] use temperature feedback from radiation pyrometers installed inside the furnace. The radiation pyrometers are also used in feedback control application by measuring slab surface temperatures after discharge from furnace or after the first roughing mill in order to compensate for remaining model uncertainties [She94, Sta04, Wö, WYY09, WWC04]. This strategy of tracking and feeding back slab surface temperatures is particularly suitable for steady-state operation.

For feedback control, reference trajectories of slab temperature profiles may be provided online or preplanned and stored in the database [CXZ+03, Bar, She94, SII+99]. Each slab may be reheated according to its individual reference trajectory depending on parameters like the slab thickness, the desired final temperature, and the scheduled residence time. The challenging task of this strategy is the development of a MIMO controller that transforms the heat input requested by each individual slab into optimal values of control variables, e.g., zone temperatures or fuel flows.

Other feedback control strategies have been developed for *steady-state furnace operation* or based on steady-state furnace models [BBG08, Bar, CA08, DCM+92, ESYS97, FBF+83, HZ82, Led86a, RB92, SII+99, VJKM08, WCGS99, WWC04, Wö]. Obviously, if the furnace has to operate in non-steady-state conditions, the basic assumption is no longer satisfied and therefore furnace performance may significantly decline. If open-loop control structures are utilized in these cases, the problem can be particularly severe because the heating history of the furnace is totally ignored.

In *two-degree-of-freedom control* structure, the output of feedback controllers are added to optimal preplanned zone temperature trajectories [DBS88, Led86a, Pri80, RB92, Sta04, SWK11, VJKM08, WCGS99, WWC04] or optimal preplanned fuel flow rate [DCM+92, Ves86]. Here, the motivation is to compensate for remaining deviations between the real furnace and the mathematical model used in feedforward control. Usually, the feedback law is based on the control error of the slab temperature trajectory and therefore requires the slab temperature to be estimated by some observer. For control of zone temperatures, a nonlinear feedback law can be developed and guarantees asymptotic stability [PW98]. Before looking into model predictive control strategy, we will now discuss on some particular aspects of furnace control.

In case where steady-state furnace models are applied for non-steady-state operating conditions like diversity of slabs, *averaging methods* are utilized. To obtain the control inputs, these methods use the averaged quantities of all considered slabs that are situating currently in the same zone. A feedback control law may be applied either before or after averaging. Thus, quantities that can be averaged are desired final values of slab temperatures, control errors of slab temperatures, heat inputs demanded by individual slabs, furnace temperatures demanded by individual slabs, fuel flow rates demanded by individual slabs, etc. In some cases, only one representative slab may be used, the approach is considered as weighted averaging technique.

Cascaded hierarchical control structure is widely utilized in the majority of published furnace control methods [BBG08, DBS88, CXZ+03, vDSK02, FRLD06b, HZ82, IZS04, Led86a, MPS04, Pri80, She94, SS04, Sta04, VGB05, VJKM08, WYY09, WWC04, ZCX+02, ZXWS01]. This control structure may facilitate the design of linear controllers for nonlinear systems. Particularly, cascaded hierarchical structure is useful for the control of

interconnected subsystems who exhibit significantly different dynamics such as, in case of reheating furnaces, furnace zones, steel slabs, fuel burners [GGS01, Lev96]. For reheating furnace control, the inner loops usually control the furnace zone temperatures, for instance by PI or PID controllers [BDM99, vDSK02, DCM+92, FRLD06b, GSR+98a, HZ82, IZS04, Sta04, VGB05, VJKM08, WYY09, WWC04, ZCX+02]. Furnace zones can be modeled as linear first order systems as argued in [NRAD+12a, VJKM08]. The design approach for inner loops presented in [NRAD+12a] accounts for the dynamic interaction between furnace zones capturing the dynamic relation between fuel inputs and zone temperatures.

In some studies, set-point values of furnace zone temperatures are stored in databases as lookup tables or predefined curves [Bar, CA08, HZ82, Led86a, Pri80, She94, VGB05]. The curves or lookup tables data can be determined off-line using steady-state planning and optimization or empirically from operator's experiences. The averaging method used in these cases accounts for slab geometry, scheduled reheating time, slab material, slab desired final temperature [Bar, FPSF90, HZ82, Pri80, WWC04]. The advantage of this approach is that it can save computer power and memory.

With some penalties on slab reheating quality, the furnace control task can be simplified into individual controls for each furnace zone. Disadvantage of this strategy is that the dynamic interactions between furnace zones are neglected [Bar, CA08, FBF+83, HZ82, PW98, Sta04, SII+99, WWC04]. *Zone-based* feedback control methods impose that the slab should reach predefined (optimal) temperature set-point values at the end of furnace zones [CXZ+03, vDSK02, DCM+92, FBF+83, MPS04]. Therefore, the error used for feedback control is usually formulated in terms of estimated slab temperatures (at the end of furnace zones). The feedback controller either defines set-point values for the zone temperature [CXZ+03, vDSK02, Pri80, She94, Sta04, WYY09, WWC04] or directly for the heat input power to the corresponding zone [DCM+92, FBF+83, Wes88] .

Prior knowledge of the future configuration and performance objectives of the furnace about slab sequencing, desired final temperatures, etc., can be utilized by some control strategies to improve furnace operating efficiency. Among these control strategies are many *optimization methods*. Optimization techniques are good choices for furnace control because they can cope well with situations where the number of control objectives or controlled variables (slab temperature profiles) significantly exceeds the number of controllable inputs (furnace zone temperatures or fuel supplies). Furthermore, these techniques allow an explicit consideration of integral control objectives like minimum total energy consumption or maximum throughput. The following classes of optimization strategies are relevant to slab reheating furnaces:

- Static optimization under steady-state operating conditions
- Heuristic improvements based on results of simulation models
- Dynamic optimization and model predictive control or also called receding horizon control

In furnace control case where steady-state conditions are assumed, *static optimization* is usually efficient and especially less computational demanding than dynamic optimization. Therefore, highly detailed and accurate

nonlinear models can be utilized in this case. The results for different operating scenarios may be stored in databases and commanded to the furnace upon demand. In published studies, optimal slab temperature trajectories can be derived as presented in [ESYS97, Ste11, FBF+83], whereas optimum zone temperature set-points can be calculated as provided in [CA08, RB92, WCGS99, WWC04, Wö, YL88]. Fuel flow rates can also be used as input variables for optimization of steady-state furnace operation [IZS04]. A general method of optimizing the distribution of local fuel inputs and the shape of counterflow reheating furnaces is presented in [KSB10].

An example of *heuristic algorithms* for furnace control can be found in [YL86]. In general, heuristic methods are undemanding in terms of development and computational costs. These criteria are also important for the fuzzy logic control approach discussed in [ZXWS01]. Optimal slab reheating trajectories are empirically designed based on a set of rules obtained from the operators' expertise. Fuzzy logic control is used for defining set-point values of furnace zone temperatures depending on control errors of slab temperatures [KDW97]. Another heuristic furnace control system based on genetic algorithm is proposed in [LGC].

On the contrary, *dynamic optimization* methods usually demand more computational power and can be executed *offline* or *online*. An offline preplanning of furnace fuel flow rates are published in [PJC70]. An optimal time-variant feedback law suitable for online implementation and direct computation of fuel flow rates is presented in [BSSP04]. These optimization methods need estimations of slab temperatures based on mathematical models which take fuel flow rates as inputs. The furnace throughput rate and fuel flow rates can be simultaneously optimized by means of mixed integer linear programming as provided in [FUK+10].

If optimization algorithms are implemented online, the furnace models must be kept at an average level in term of computation demand. Therefore, the complexity of underlying furnace models is a constraint for control problem.

In [MPS04, SII+99, Ste11], a quadratic objective function is minimized to find optimal furnace zone temperature set-points. This objective function contains weighted average of deviations between the realized and the desired heat inputs into the slabs. The heat inputs are calculated based on the desired slab temperatures at the end of each furnace zone.

A notable control strategy that uses dynamic optimization in closed loop is *model predictive control*. This method has been applied to slab reheating furnaces as reported in [BBG08, CXZ+03, IZS04, KKY+00, NvWS, ALRS05, YNY91, YUU94, Ste11, SWK13].

A hierarchical cascaded control structure where model predictive control is used in each loop is proposed in [BBG08]. The primary control loop calculates optimal furnace zone temperatures assuming a nonlinear furnace model subject to steady-state conditions. A linear model predictive control routine which is implemented in the secondary loop, provides dynamic optimization of both slab advancing rate and fuel supplies. Similar model predictive control system is suggested in [KKY+00]. Fuel flow rates are controlled based on linearized MIMO models of the furnace zone temperatures. Energy consumption and furnace throughput can be both optimized at the same time with model predictive control strategies presented in [ALRS05]. There, the desired final slab temperature is only accounted as constraints of the optimization problem.

Also based on hierarchical cascaded structure, [NvWS] presented a *nonlinear model predictive controller* used in the primary loop where furnace zone temperature set-points are optimized. A transient furnace model is utilized in dynamic optimization of both furnace zone temperatures and production rate. The quadratic cost function accounts for control errors in terms of final slab temperatures and for furnace zone temperatures to reduce the energy consumption.

A highly elaborated dynamic optimization scheme for a slab reheating furnace is reported in [YNY91, YUU94]. Initially, a nonlinear optimization problem is solved to obtain reference trajectories of slab temperatures. They are designed as piece-wise polynomial functions, which allow the formulation of a low-dimensional and unbounded optimization problem. Moreover, the method of weighted residuals for computing the one-dimensional temperature profile inside the slabs is used to keep an acceptable complexity level of the optimization problem. A second optimization problem based on a linearized model is finally solved for selecting fuel flow rates which ensure that the slabs follow their respective optimized temperature trajectories. Here, a *generalized predictive control* method suggested in [CMT87] is applied. Both optimization procedures are executed in real time.

Recent development on *model predictive control* for slab reheating furnace is reported in [Ste11, SWK13]. In these publications, a nonlinear model predictive controller is designed based on a first-principle mathematical model, the controller defines local furnace temperatures so that the slabs reach their desired final temperatures. The controller is able to cope with non-steady-state operating conditions of the furnace and reaches the desired slab temperature profiles. The nonlinear unconstrained dynamic optimization problem is solved by the quasi-Newton method.

In [ZL07], a *distributed model predictive control* based on a neighborhood optimization is demonstrated for slab reheating furnace case. The furnace zones are considered to be serially connected (cascaded processes), that is, interactions between furnace zones are taken into account for furnace control.

In terms of algorithms, there are several standard methods for solving mathematically formulated optimization problems. The best choice among these algorithms for the general reheating furnace problem remains absent. Actually, it depends on the available computing power and the mathematical structure of both the model and the objective function. In the context of furnace control, one can find the utilization of linear programming (simplex algorithm) [CXZ+03, ESYS97, WCGS99, WWC04], in [FUK+10] with mixed integer linear programming, in [BBG08, YNY91, YUU94] with quadratic programming, or in [IZS04, NvWS] by nonlinear programming. Real-time implementations of solving a nonlinear optimization problem are described in [YNY91, YUU94] with method of steepest descent and in [Ste11, SWK13] with quasi-Newton method.

4.2 Existing and proposed control design of the considered furnace

In the considered reheating furnace, temperature control is designed based on hierarchical cascaded structure as shown in Figure 4.2. The scheduling and calculation of target temperature for each slab are at level 3 of the system. At this level, the slab requirements of hot rolling mill corresponding to client demand vary in terms

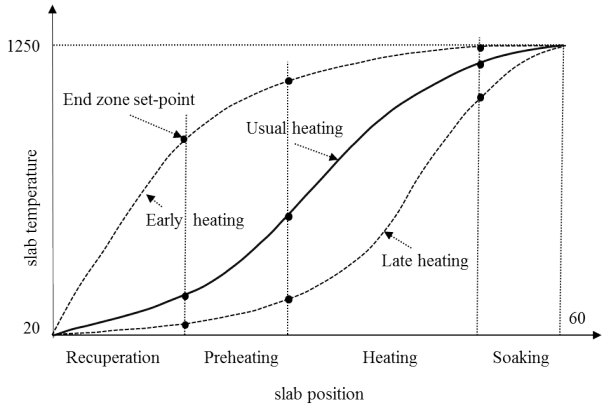

Figure 4.1: Heating trajectory of slab inside reheating furnace

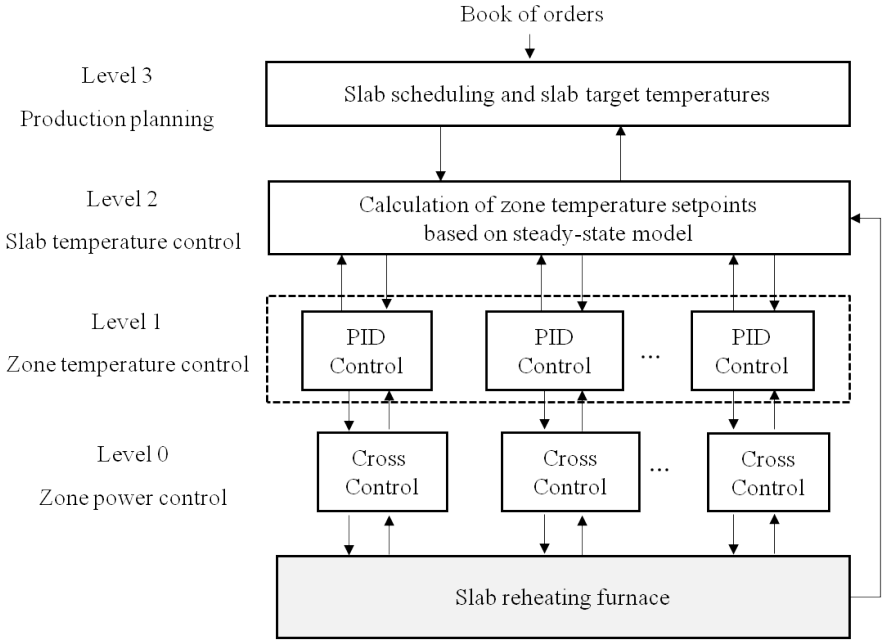

Figure 4.2: Hierarchical cascaded structure of reheating furnace

of slab dimensions, desired final temperatures, the slab steel grade, etc. When slabs are scheduled, necessary amount of heat input for each slab is calculated based on its initial temperature and desired final temperature. A heating trajectory of slab temperature is then constructed *offline* for each slab according also to the available heating power of each furnace zone, see Figure 4.1. Based on this temperature trajectory, target temperatures of slab at the end of controllable zones: preheating, heating, and soaking zones are predetermined.

Data of slab scheduling, slab target temperature at the end of each zone is commanded to level 2 where set-points of furnace zone temperatures are calculated. At this level, the steady-state model described in Chapter 3 is used to predict temperature of a slab at the end of its current zone by imposing a value of temperature set-point for the zone. Similarly, temperature set-point of the zone is calculated for all other slabs currently being inside the zone. Each slab inside a zone will demand a different zone temperature set-point. Therefore, a *weighted averaging technique* is applied. Usually, for a time instant k, slabs that situate at the end or entry of a zone will

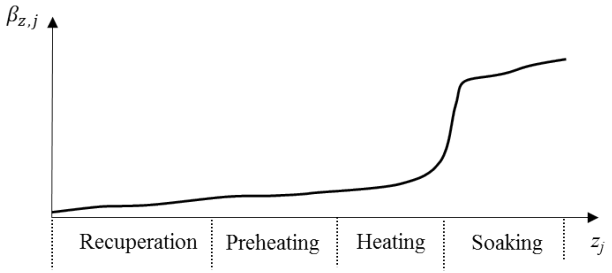

Figure 4.3: Averaging technique of current furnace control

be considered less important than those being at the middle of the zone, see Figure 4.3. That is because zone temperature influences most significantly on slab that is situated at the middle position. The crucial slab of a zone is the one of which temperature success is the most important according to client requirement, thus zone temperature set-points considerably depend on this slab.

In furnace operation, it is generally better to have rather slightly overheated slabs than under-heated slabs because under-heating of slabs causes incidents in hot rolling steps, which costs a lot in terms of maintenance and production delays. However, overheating has also to be restricted because it can lead to slab melting which also stops the heating process. Moreover, overheating generates more scale formation on slab surfaces than usual, therefore more material is lost. Waste of energy is obviously undesirable effect caused by overheating of slabs. As a result, slab overheating significantly contributes to the price of final steel products.

In the current situation of the considered furnace, it happens that slabs have a large variety of dimensions, target temperature, and material due to diversity of client demands. Therefore, between two consecutive slabs, there can be considerable difference between desired final temperature. In these cases, the furnace control will impose furnace temperatures that will successfully heat the slab that has higher desired final temperature. Hence, in average, slabs are overheated. The value representing this overheating tendency is a performance criteria of the furnace which reflects also the accuracy of furnace control. A new control strategy which gives higher accuracy will decrease the gap between discharge and desired final temperature and therefore reduce energy consumption of the furnace.

Level 1 of the furnace control system tracks furnace zone temperatures according to their temperature set-points which are calculated by the level 2. Power set-points for each zone will be computed and be sent to lower level of the control system. In the considered furnace, PI controllers are used to control zone temperatures as shown in Figure 4.2. The fact that furnace zones strongly interact with each other influences considerably the performance of PI controllers. Therefore, new controllers that take into account the zone interactions will improve tracking performance of zone temperatures and also reduce energy consumption of the furnace.

The proposed solution for furnace control bases on the same structure as for the current control system as shown in Figure 4.4. At level 2, a nonlinear model predictive controller is designed. The dynamical furnace model presented in Chapter 3 is expected to cope with non-steady-state furnace operations and gives more accurate estimations of slab temperature profiles. This model is also utilized to predict furnace quantities from a sequence

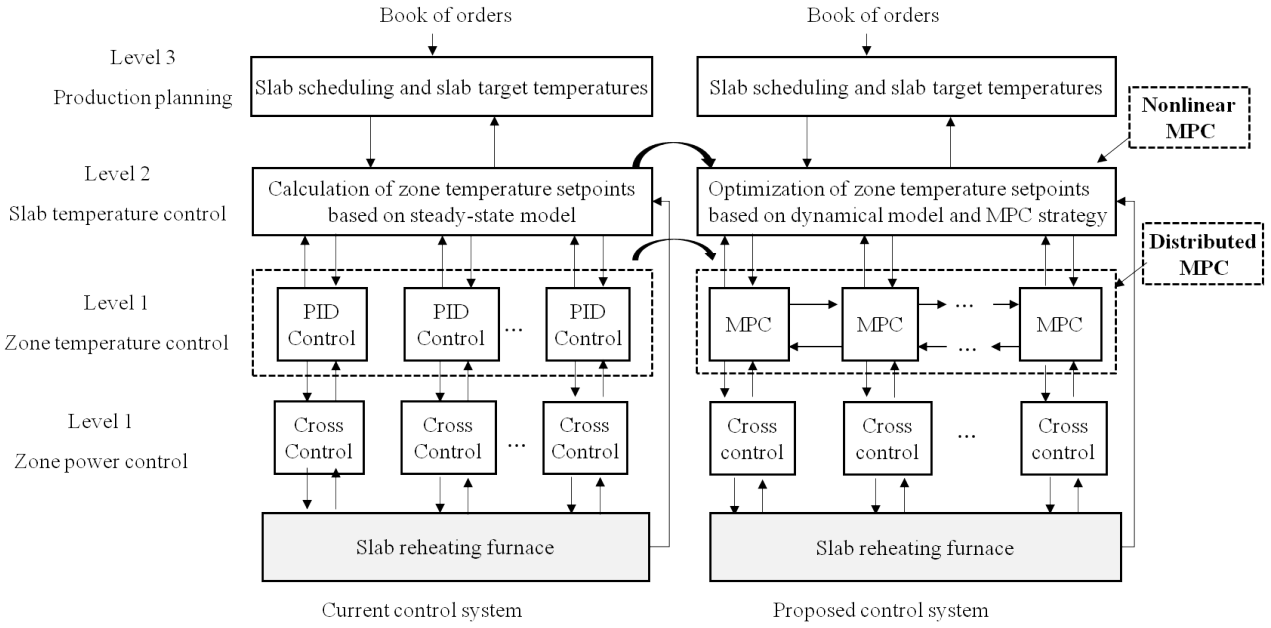

Figure 4.4: MPC structure for walking-beam reheating furnace

of zone temperature setpoints like: furnace consumption, zone consumption, zone input power, etc. From these predictions, an objective function is constructed taking into account errors of discharge temperatures, fuel consumption, constraints of furnace such as: limit wall temperature, maximum power of furnace zone, maximum allowed temperature gradient inside slabs. Optimal zone temperature set-points of furnace zones are then obtained by solving optimization problem.

As described in Chapter 3, the dynamical model of the furnace is a numerical model which uses finite volume method to calculate slab temperatures. Therefore, objective function cannot be written in analytic form of control variables which are temperature set-points of furnace zones. Especially, partial derivatives of slab temperature will not be calculated directly. Even estimation of the derivatives is expensive in term of computational resources. Therefore, optimization algorithm will have to satisfy time constraint of the problem. Here, direct optimization algorithms are investigated to find a suitable one for the reheating furnace control problem.

For the level 1, a distributed model predictive control method is proposed to replace PID controllers as shown in Figure 4.4. This method benefits from knowledge about the dynamic and coupling of furnace zones to calculate optimal input power to each zone. An iterative algorithm is constructed based on a output coupling model of furnace zones. The optimization problem is solved by quadratic programming.

All these control architecture of levels 1 and 2 will be further described in the next chapters.

Chapter 5

Zone temperature control

Control system of the considered furnace has a cascaded hierarchical structure as described in chapter 4. At level 1, the tracking of zone temperature set-points is currently ensured by PID controllers implemented in decentralized scheme. In reheating furnaces, however, furnace zones are significantly coupled with each other. The decentralized PID controllers cannot therefore explicitly tackle these interactions. In this chapter, distributed model predictive controllers (distributed MPCs) are designed to replace decentralized PID controllers, based on furnace zone models. First, an overview on distributed MPC is given. Second, the modeling of furnace zones and their interactions will be described. Identification procedures are detailed for the considered furnace zones. Afterwards, distributed MPCs are constructed with a non-cooperative strategy in which a local MPC controller only optimizes the performance of one zone, or a cooperative strategy in which a local MPC controller not only optimizes the performance of one zone but also that of the neighboring zones. Iterative algorithms are also proposed for each strategy to increase the optimality of the solutions. These algorithms are accompanied with closed-loop stability and convergence analysis. Simulation on an illustrative example is carried out showing the efficiency of the proposed distributed MPCs algorithms. Experimental results show the improvements obtained with distributed MPCs in comparison to decentralized PID controllers.

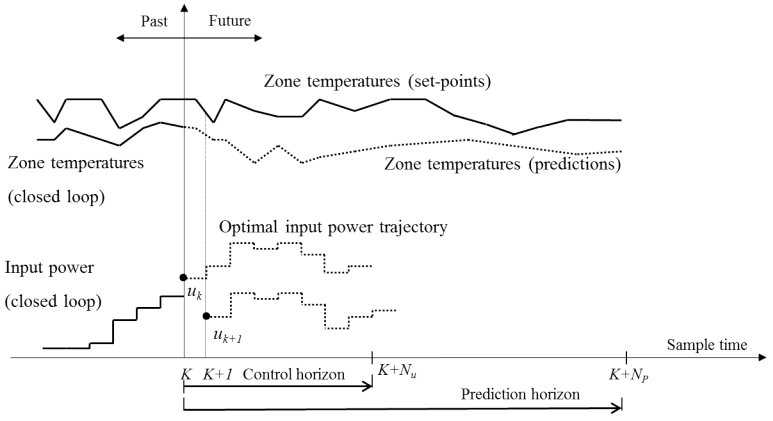

Figure 5.1: Model predictive control principles

5.1 Overview on distributed MPC

MPC is an advanced control method granted with a great interest as much in theoretical studies as in industrial application practice. The basics of MPC is utilization of a process model to predict output variables over a finite horizon. Especially, MPC is able to handle process constraints, to deal with multivariable processes. The control problem is transformed to an optimization one. And, optimal control inputs over a finite horizon are derived by solving an optimization problem which usually represents an economical criterion. Principles of MPC applied to the furnace zone temperature control are illustrated in Figure 5.1. At each time instant k, an optimization problem is formulated accounting, e.g. in our specific case, for zone temperature tracking errors and furnace energy consumption over a finite horizon. Optimal input trajectory for time k is derived by solving this optimization problem. Only the first action of the optimal trajectory is employed. The procedure is repeated for the next sample time $k + 1$. Nowadays, with the fast growth of computational power, process modeling becomes more and more affordable for real-time applications. As a result, MPC method is widely applied to industrial processes replacing the traditional PID controllers, especially in economically important units [RS08].

Reheating furnaces are networked large-scale engineering systems that consist of a number of subsystems interacting with each other through flows of material, energy, and information. The subsystems are represented by furnace zones, flow of material is the steel slab charge moving through the furnace. Information from thermocouples, pressure sensors, etc., allows the monitoring and control of the furnace. Energy flow inside furnace is mainly the heat supplied from fuel combustion.

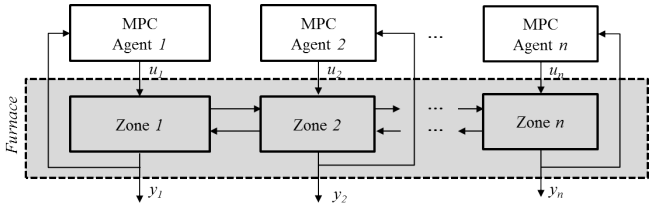

Figure 5.2: Structure of decentralized MPC

MPC can be employed individually for each zone of the furnace with a decentralized structure as shown in Figure 5.2. That is, local models and performance indexes are specified for each furnace zone. The interactions among furnace zones are therefore ignored in controller design. In processes where subsystems weakly interact, local feedback action provided by local controllers may be sufficient to compensate the coupling effects. Therefore, decentralized MPC strategy is expected to work adequately. For many plants where interactions among subsystems are strong, ignoring these interactions leads to significant loss of control performance. In reheating furnaces, steel slabs are heated in high temperature zones. The radiation is a global and dominant heat transfer phenomenon as reasoning in chapter 3. In addition, circulation of high temperature gas also makes couplings between furnace zones considerable. For these reasons, decentralized MPC will not be suitable for the control of zone temperatures.

In contrast to decentralized MPC, centralized MPC structure uses a MIMO model for large-scale systems, therefore takes into account every coupling effect between sub-systems, see Figure 5.3. However, centralized controllers, are viewed by practitioners as inflexible and monolithic solution. For most large-scale processes, and so reheating furnaces, the primary burdens to centralized control are not computational but organizational [Ven06]. On the one hand, operators are usually unwilling to deal with the substantial data collection and data handling effort required to design and maintain a valid centralized control system for a large plant. To the best of our knowledge, no such centralized control systems are operational today for any large, networked system. In fact, zones of furnace were individually controlled by PID controllers. And so, a decentralized structure was already in place and plant personnel does not wish to completely redesign the control system to implement centralized MPC. On the other hand, different zones of the furnace are subject to modifications through time, which makes the model development and maintenance effort required for centralized control impractical. Unless these organizational hurdles change in the future, centralized control of networked, large-scale systems stays useful mainly as a benchmark against which other control strategies can be compared and evaluated.

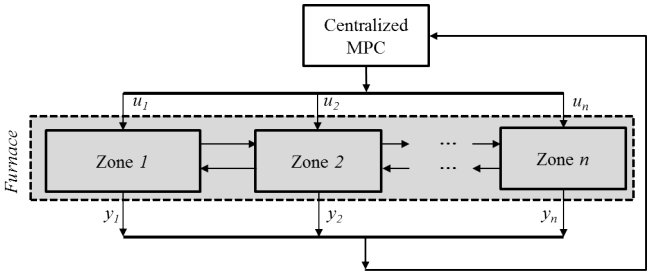

Figure 5.3: Structure of centralized MPC

As a compromise between centralized MPC and decentralized MPC, distributed MPC can use the framework of decentralized MPC, and accounts for the coupling effects of furnace zones, see Figure 5.4. That is, on one hand, local MPC controllers are able to communicate with each other to inform the changes happen in each subsystem. On the other hand, a MPC agent can cooperate with others by optimizing the performance criterion of a set of subsystems [RS08, Ven06], or optimizes independently the performance criterion of one individual subsystem.

In term of optimality of control solution, in a fully cooperative algorithm where each local controller will optimize a global performance criteria, the overall system will converge to a so called Pareto equilibrium [CJKT02, AGBE11, MN14]. On the contrary, in a non-cooperative algorithm where each local controller tries to maximize only the performance of its corresponding subsystem, the overall system will come to the solution called Nash equilibrium [Nas51, ZL07, MN14]. To understand notions of Nash and Pareto equilibriums in an easy way, suppose that there are two controllers which will take action to optimize performance of two coupled subsystems: controller 1 will do action a_1, controller 2 will do action a_2, in order to minimize objective functions: $J_1(a_1, a_2)$ and $J_2(a_1, a_2)$. A set of actions $\left(a_1^N, a_2^N\right)$ constitutes a Nash equilibrium iff: $J_1(a_1^N, a_2^N) \leq J_1(a_1, a_2^N)$ for any a_1 and $J_2(a_1^N, a_2^N) \leq J_2(a_1^N, a_2)$ for any a_2. In other words, Nash equilibrium is where no controllers can act unilaterally to improve their cost knowing action of the other. While a set of actions $\left(a_1^P, a_2^P\right)$ is said to be Pareto optimal if there does not exist another set (a_1, a_2) such as $J_1(a_1, a_2) \leq J_1(a_1^P, a_2^P)$ and $J_2(a_1, a_2) \leq J_2(a_1^P, a_2^P)$. Therefore, Pareto solution is the best solution that two controllers can obtain acting together.

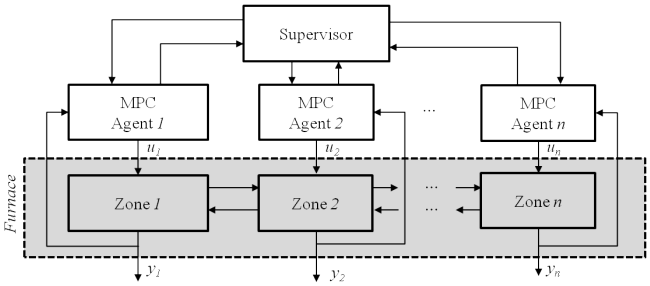

Figure 5.4: Structure of distributed MPC

Distributed MPC structure		
Communication topology	*Optimality*	*Communication protocol*
Fully connected	Nash equilibrium(Non-cooperative strategy)	Iterative
Partially connected	Pareto equilibirum(Cooperative strategy)	Non-iterative

Table 5.1: Classification of distributed MPC structure

In term of communication among controllers, in case of highly interactive system-wide, it requires a huge resource for communication. For a less interactive system-wide the degree of communication is reduced. As a result, communication structure of distributed MPC varies according to coupling topology of the system-wide. Therefore, two communication topologies can be defined for the design of distributed MPCs: 1) MPC agents are fully connected with each other and 2) MPC agents are partially connected in a neighborhood [RS08].

In term of communication protocol, in platforms where hardware resource is abundant, the optimization and communication procedures can be executed in an iterative way to guarantee the optimal solution. Otherwise, non-iterative solution may still be sufficient for control requirements. To summarize different structures that distributed MPC could have, a classification of distributed MPC is given in Table 5.1.

In metallurgical processes, highly interactive system-wide is not the common case. Material, energy and information generally flow sequentially from sub-system to sub-system, which characterizes the major coupling effects. In fact, the interactions inside a reheating furnace are limited in a neighborhood of a zone. Each furnace zone interacts mainly with its two neighboring zones. By taking advantage of the coupling topology, we can reduce the required communication between the MPC agents. In the following, modeling of furnace zones and their interactions will be presented.

5.2 Furnace zone modeling

Modeling of furnace zones concerns identifying dynamic of zone temperatures with respect to variation of input heat powers. Therefore, the modeling involves only furnace zones equipped with fuel burners which are: preheating zone, heating zones and soaking zone. For the considered furnace, heating powers are mainly installed at preheating and heating zones where slabs absorb most of the necessary heat. Two halves of heating zones are separately controlled as shown in Figure 5.5. Soaking zone provides the additional heat for slabs to homogenize

the temperature gradient. In fact, the soaking zone is isolated from the other zones to ensure the homogeneity of temperature. Therefore, as we want to utilize distributed MPC strategy to reduce energy consumption of the furnace, only preheating zone and heating zones are considered in this part of the dissertation.

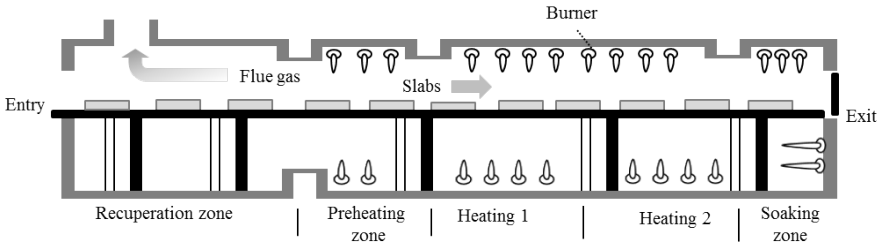

Figure 5.5: Controllable zones of walking-beam reheating furnace

Theoretically, furnace zones are nonlinear time variant systems with temperature dependent parameters such as steel heat conductivity, specific heat, etc. However, in real furnace operation, each furnace zone is often kept at some temperature level which can refer to operating points. For the considered furnace, preheating, heating 1 and heating 2 operate around 950°C, 1150°C and 1300°C, respectively. Therefore, temperature dependent parameters can be considered as constant for each furnace zone.

The load of slabs inside a zone is the most relevant time variant parameter for furnace modeling. This is due to the diversity of slab dimensions and materials. Especially, the load of a zone may change dramatically with presence of considerable space among slabs as demonstrated in Figure 5.6. In normal operating conditions, furnace production rate varies around a nominal value and slab load of a zone varies slightly around a certain value. However, through different experiments on the real furnace, it is observed that in a normal operating condition, the dynamic of zone temperature does not depend significantly on slab load inside the zone.

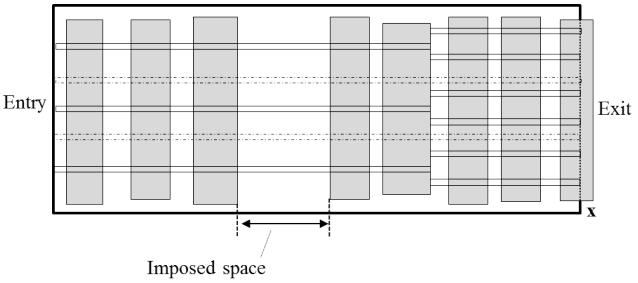

Figure 5.6: Imposed space between slabs

Also by observation on real furnace, the dynamic of zone temperature with respect to variations of input power is similar to a first order system with delays and can be expressed as follows:

$$H_j(s) = \frac{\mathcal{L}[y_j(t)]}{\mathcal{L}[u_j(t)]} = \frac{G_{jj}e^{-\tau_{jj}s}}{1+\lambda_{jj}s}, \tag{5.1}$$

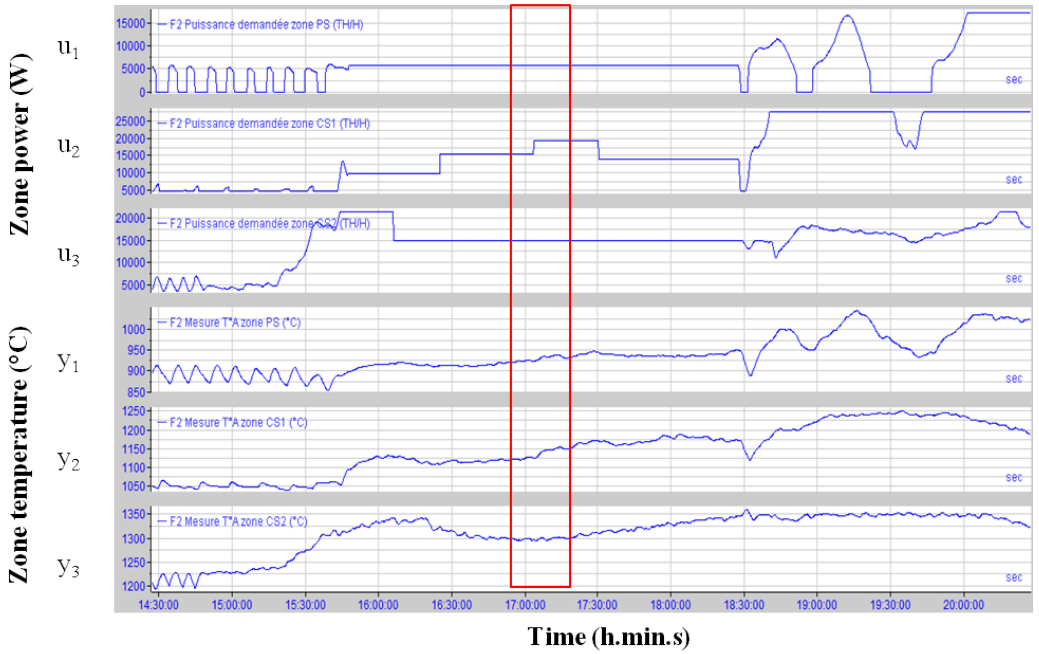

Figure 5.7: Step signal of zone preheating

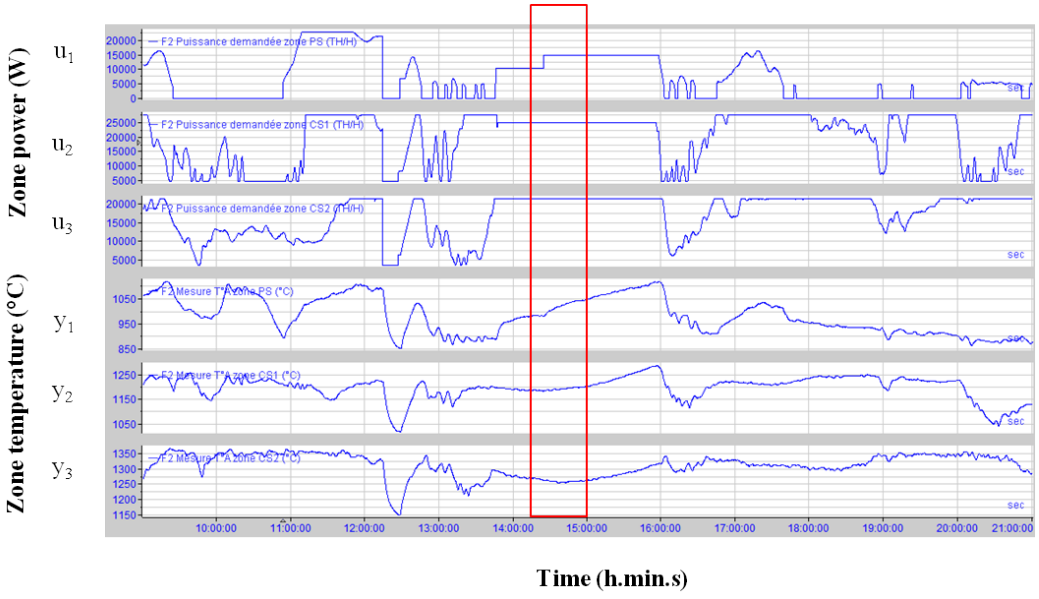

Figure 5.8: Step signal of zone heating 1

where $y_j = \Delta T_j = T_j - T_{j,nom}$ the variation of zone temperature T_j from its nominal value $T_{j,nom}$, $u_j = \Delta P_j = P_j - P_{j,nom}$ the variation of zone input power P_j from its nominal value $P_{j,nom}$, $\mathcal{L}$ represents the operator of Laplace transformation, and s the Laplace variable.

Couplings among furnace zones are mainly due to convective heat transfer of flue gas and radiative heat transfer inside the furnace. The flue gas flows from exit to entry of the furnace in opposite direction of slab movement as depicted in Figure 5.5. As mentioned in furnace modeling chapter, convective heat transfer depends on difference

between participating object's temperature, while radiation heat transfer is proportional to 4^{th} power of temperature difference of participating surfaces. Therefore, coupling between two neighboring zones j and $i = j \pm 1$ can be modeled with respect to temperatures. This topology of interaction refers to an output coupling model. The dynamic of a zone's temperature according to variation of its neighboring zone's temperature is observed to be similar to first order transfer function with delays and can be expressed as follows:

$$H_{ij}(s) = \frac{\mathcal{L}[y_i(t)]}{\mathcal{L}[y_j(t)]} = \frac{G_{ij}e^{-\tau_{ij}s}}{1+\lambda_{ij}s}. \tag{5.2}$$

Interaction of two neighboring zones can also be modeled as input coupling. By observation, dynamic of a zone's temperature with respect to variation of its neighboring zone's input power is also similar to first order transfer function with as shown in Figures 5.8 and 5.7. The system can be represented by

$$H_{ij}(s) = \frac{\mathcal{L}[y_i(t)]}{\mathcal{L}[u_j(t)]} = \frac{G_{ij}e^{-\tau_{ij}s}}{1+\lambda_{ij}s}. \tag{5.3}$$

In conclusion, around a nominal operating condition, furnace zones can be simplified to linear time invariant delayed first order systems. In the following, two models of the considered zones with two distinct coupling structures defined by Equations 5.2 and 5.3 are detailed for the design of distributed MPC strategy.

5.2.1 Input coupling model

Zone model based on input coupling assumption is depicted as shown in Figure 5.9. Three subsystems correspond to preheating, heating 1 and heating 2 zone, respectively. The overall model of three subsystems is written as follows:

$$Y_1(s) = H_{11}(s)U_1(s) + H_{12}(s)U_2(s) + H_{13}(s)U_3(s), \tag{5.4a}$$
$$Y_2(s) = H_{21}(s)U_1(s) + H_{22}(s)U_2(s) + H_{23}(s)U_3(s), \tag{5.4b}$$
$$Y_3(s) = H_{31}(s)U_1(s) + H_{32}(s)U_2(s) + H_{33}(s)U_3(s), \tag{5.4c}$$

with $H_{ii} = H_i$, for $i = 1, 2, 3$, given in Equation 5.1, and $H_{ij}, j \neq i$ described in Equation 5.3. Because interactions are mainly between neighboring zones, H_{13} and H_{31} are neglected. Therefore, in Equation 5.4, coupling terms H_{13} and H_{31} are neglected. The input coupling model of zones is simplified to

$$Y_1(s) = H_{11}(s)U_1(s) + H_{12}(s)U_2(s), \tag{5.5a}$$
$$Y_2(s) = H_{21}(s)U_1(s) + H_{22}(s)U_2(s) + H_{23}(s)U_3(s), \tag{5.5b}$$
$$Y_3(s) = H_{32}(s)U_2(s) + H_{33}(s)U_3(s). \tag{5.5c}$$

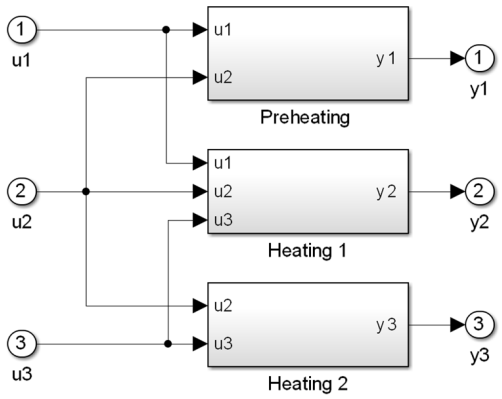

Figure 5.9: Input coupling of furnace zones

The input coupling model of the considered furnace has been identified in previous work reported in [DP, JR] as

$$Y_1(s) = \frac{0.0140e^{-59s}}{1+513.1s}U_1(s) + \frac{0.0048e^{-60s}}{1+460s}U_2(s) \tag{5.6a}$$

$$Y_2(s) = \frac{0.0031a^{-70s}}{1+520s}U_1(s) + \frac{0.0108e^{-60s}}{1+460s}U_2(s) + \frac{0.005e^{-65s}}{1+533.3s}U_3(s) \tag{5.6b}$$

$$Y_3(s) = \frac{0.0038e^{-90s}}{1+438s}U_2(s) + \frac{0.0125e^{-60s}}{1+500s}U_3(s) \tag{5.6c}$$

In this coupling model of furnace zones, we see that temperature of a zone is more rapidly affected by the input power of that zone than by input power of other zones. Specifically, for the temperature of zone preheating (y_1), delay from u_1 to y_1 is 59 s whereas delay from u_2 to y_2 is 60 s. For temperature of the zone heating 1 (y_2), delay from u_2 to y_2 is 60 s while delays from u_1 and u_3 to y_2 are 70 s and 65 s, respectively. And for the zone heating 2, delay from u_3 to y_3 is 60 s when delay from u_2 to y_3 is 90 s which is significantly longer. In addition, the static gain from input power of a zone to temperature of that zone is more significant than that to temperature of neighboring zones. For example, in zone heating 1, static gain from u_2 to y_2 is 0.0108 while the gains from u_1 and u_3 to y_2 are only 0.0031 and 0.005, respectively.

5.2.2 Output coupling model

Zone model based on output coupling assumption is depicted as shown in Figure 5.10. Three subsystems correspond to preheating, heating 1 and heating 2 zone, respectively. The overall model of three subsystems is written as follows:

$$Y_1(s) = H_{11}(s)U_1(s) + H_{12}(s)Y_2(s) + H_{13}(s)Y_3(s), \tag{5.7a}$$

$$Y_2(s) = H_{21}(s)Y_1(s) + H_{22}(s)U_2(s) + H_{23}(s)Y_3(s), \tag{5.7b}$$

$$Y_3(s) = H_{31}(s)Y_1(s) + H_{32}(s)Y_2(s) + H_{33}(s)U_3(s), \tag{5.7c}$$

with $H_{ii} = H_i$, for $i = 1 \div 3$, is given in Equation 5.1, and H_{ij}, $j \neq i$, is described in Equation 5.2. In identifi-

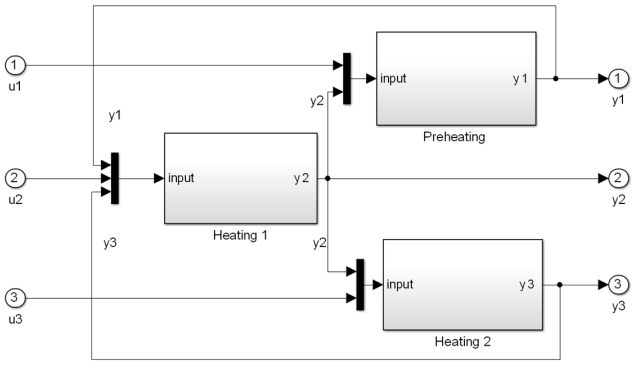

Figure 5.10: Output coupling of furnace zones

cation of output coupling model of furnace zones, some coupling terms are obsolete. For reheating furnaces, the flue gas circulates from furnace exit to furnace entry meaning from high temperature zones to low temperature zones. By that, the supplied thermal energy is redistributed to heat cold slabs. In addition, zone temperatures are affected directly by temperature of the flue gas. Therefore, the coupling effects happen mainly in the direction of circulation of the flue gas. As a result, the influence of zone with lower temperature to zone with higher temperature is neglected. Thus, in the Equation 5.4 coupling terms H_{21}, and H_{32} are null. On the other hand, the temperatures of zones which are not neighbor do not have direct link to each other. Consequently, the coupling terms H_{13}, H_{31} are neglected. The output coupling model of zones is simplified as

$$Y_1(s) = H_{11}(s)U_1(s) + H_{12}(s)Y_2(s), \tag{5.8a}$$

$$Y_2(s) = H_{22}(s)U_2(s) + H_{23}(s)Y_3(s), \tag{5.8b}$$

$$Y_3(s) = H_{33}(s)U_3(s). \tag{5.8c}$$

The obtained output coupling model is given as:

$$Y_1(s) = \frac{0.0143e^{-60s}}{1+605s}U_1(s) + \frac{0.210e^{-20s}}{1+500s}Y_2(s) \tag{5.9a}$$

$$Y_2(s) = \frac{0.012e^{-40s}}{1+742s}U_2(s) + \frac{0.16e^{-20s}}{1+416s}Y_3(s) \tag{5.9b}$$

$$Y_3(s) = \frac{0.0066e^{-70s}}{1+415s}U_3(s) \tag{5.9c}$$

These two models are validated against measurement with a different set of data of zone temperatures as shown in Figures 5.12 and 5.11. Generally, it can be seen that both identified models can capture the temperature dynamic of three considered zones. However, the output coupling model fits more the measurement for temperature of zone: heating 1 and heating 2. This can be explained by the fact that temperature of a zone is directly influenced by the temperatures of neighboring zones. For the next sections, output coupling model will be used for controller design.

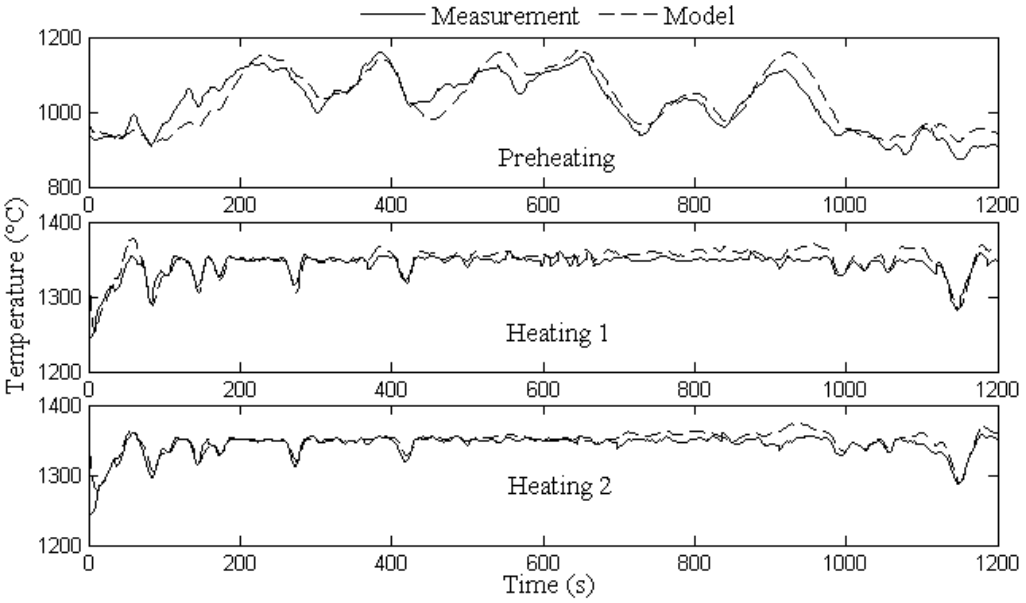

Figure 5.11: Validation of output coupling model

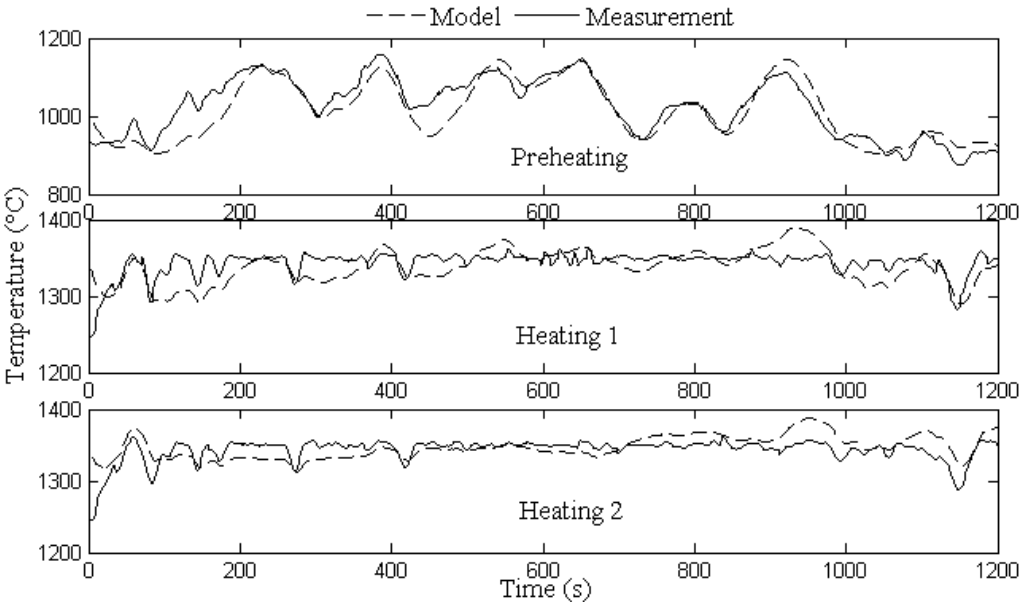

Figure 5.12: Validation of input coupling model

5.3 Distributed MPC design

After examination of furnace zone models, we now generalize the furnace zones as subsystems which are coupled through outputs and distributed MPC strategies are constructed accordingly. First, the predictions of outputs is formulated based on a state space representation of system. Second, the control problem is formulated for two distinct strategies of distributed MPC: cooperative and non-cooperative. Additionally, iterative algorithms are designed assuring solutions with a low level of suboptimality. Also, convergence properties of iterative

algorithms and nominal stability of closed-loop system are analyzed for an unconstrained case. Finally, the centralized MPC strategy is constructed and employed as a benchmark for two distributed MPC algorithms.

5.3.1 Networked system modeling

Assuming that the system-wide is composed of M subsystems coupled with each other by outputs, for the subsystem i, the state space representation can be written as

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j=1, j\neq i}^{M} B_{ij} y_j(k) \\ y_i(k) = C_i x_i(k) \end{cases} \tag{5.10}$$

This representation is based on a linearization around the operating point of the furnace. Vectors $u_i \in \Re^{n_i}$ and $y_i \in \Re^{m_i}$ are respectively the variations of input power, and temperature of zone i. The term B_{ij} is the coupling coefficient of zone i with zone j. It should be noted that a zone i interacts only with its neighboring zones. Consequently, $B_{i,j} = 0$ if $j \neq (i \pm 1)$. The couple $\{C_i, A_i\}$ is assumed to be observable for $i = 1 \div M$. The prediction of output $y_i(k+s)$ at instant k is denoted by $\hat{y}_i(k+s \mid k)$. From the state space representation, $\hat{y}_i(k+s \mid k)$ is calculated as

$$\hat{y}_i(k+s \mid k) = C_i A_i^s x_i(k) + \sum_{p=1}^{\min(s,N_u)} C_i A_i^{s-p} B_i u_i(k+p-1 \mid k) + \sum_{j\neq i} \sum_{p=1}^{s} C_i A_i^{s-p} B_{ij} \hat{y}_j(k+p-1 \mid k), \tag{5.11}$$

N_u is the control horizon. In a vector form, the predictions of output over prediction horizon N_p are gathered as follows:

$$\hat{Y}_i(k \mid k) = \left[\hat{y}_i'(k+1 \mid k), \ldots, \hat{y}_i'(k+N_p \mid k)\right]' \tag{5.12}$$

Similarly, the control input sequence over N_u steps is put into a vector as

$$U_i(k \mid k) = \left[u_i'(k \mid k), \ldots, u_i'(k+N_u-1 \mid k)\right]'. \tag{5.13}$$

Output prediction of zone i in Equation 5.11 is therefore rewritten as

$$\hat{Y}_i(k \mid k) = \Psi_i x_i(k) + \Phi_{i,i} U_i(k \mid k) + \sum_{j\neq i} \Phi_{i,j} \hat{Y}_j(k \mid k), \tag{5.14a}$$

where $\Psi_i = \left[(C_i A_i)' \ldots (C_i A_i^{N_p})'\right]'$, and

$$\Phi_{i,i} = \begin{bmatrix} C_i B_i & \cdots & 0 \\ \vdots & \ddots & \vdots \\ C_i A_i^{N_u-1} B_i & \cdots & C_i B_i \\ \vdots & \vdots & \vdots \\ C_i A_i^{N_p-1} B_i & \cdots & C_i A_i^{N_p-N_u} B_i \end{bmatrix}, \Phi_{i,j} = \begin{bmatrix} C_i B_{i,j} & 0 & \cdots & 0 \\ C_i A_i B_{i,j} & C_i B_{i,j} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ C_i A_i^{N_p-1} B_{i,j} & C_i A_i^{N_p-2} B_{i,j} & \cdots & C_i B_{i,j} \end{bmatrix}.$$

5.3.2 Observer design for output feedback control

In the considered reheating furnaces, available measurements are zone temperatures which are the controlled outputs. In order to design a controller, an observer is needed to estimate the states of subsystem from temperature measurements. The structure of observer is shown in Figure 5.13. In this case, pole-placement technique is employed by introducing state and output estimations: $\bar{x}_i(k)$, $\bar{y}_i(k)$, respectively. The structure of the observer

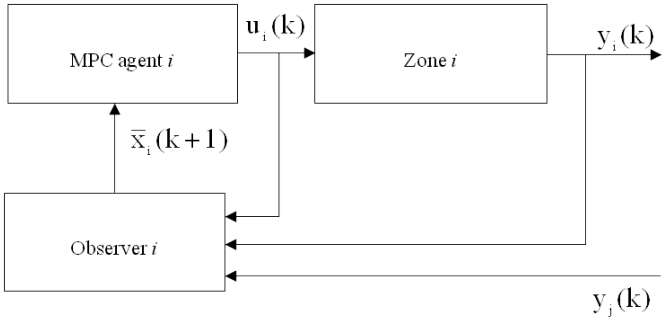

Figure 5.13: Structure of observers

is given with the following state space representation

$$\bar{x}_i(k+1) = A_i\bar{x}_i(k) + B_iu_i(k) + L_i\left[y_i(k) - \bar{y}_i(k)\right] + \sum_{j\neq i} B_{i,j}y_j(k) \tag{5.15a}$$

$$\bar{y}_i(k) = C_i\bar{x}_i(k) \tag{5.15b}$$

with L_i is a feedback vector. State estimation error is defined as $\tilde{x}_i(k) = \bar{x}_i(k) - x_i(k)$. From Equations 5.10 and 5.15 the dynamic of estimation error is given as

$$\tilde{x}_i(k+1) = \bar{x}_i(k+1) - x_i(k+1) \tag{5.16a}$$

$$= A_i\bar{x}_i(k) - A_ix_i(k) + L_i\left(y_i(k) - \bar{y}_i(k)\right) \tag{5.16b}$$

$$= A_i\tilde{x}_i(k) + L_i\left(C_ix_i(k) - C_i\bar{x}_i(k)\right) \tag{5.16c}$$

$$= (A_i - L_iC_i)\tilde{x}_i(k). \tag{5.16d}$$

The estimate state $\bar{x}_i(k)$ will converge to the real state $x_i(k)$ with L_i that satisfies

$$eig\left(A_i - L_iC_i\right) < 1. \tag{5.17}$$

That is, the eigenvalues of $(A_i - L_iC_i)$ are all inside the unit circle. The dynamic of the observer has to be faster than the dynamic of the subsystem to provide the information for the controller. In general, eigenvalues of $(A_i - L_iC_i)$ are chosen and imposed according to eigenvalues of A_i, L_i is then calculated subsequently.

5.3.3 Non-cooperative distributed MPC

Temperature tracking performance is defined with repects to specific criteria, in particular zone temperature accuracy and energy consumption. In a non-cooperative strategy, a local MPC agent optimizes only the performance of the corresponding zone while taking into account influences of other zones. Therefore, the local controller has to gather output predictions of other zones in order to make an accurate predictions of output of the corresponding zone. In return, the controller has to send out its output predictions so that other controllers can make their own predictions.

In this section, first, the formulation of objective function will be given, then followed by the description of an iterative algorithm. Afterwards, the convergence of the iterative algorithm and closed-loop stability for an unconstrained case are examined.

5.3.3.1 Problem formulation

Zone temperature control is subject to limitations of zone power and variation speed of zone power, and to upper/lower bounds of zone temperatures. Thus, optimization problem for local MPC agent i is defined as

$$\min_{u_i(k|k),\ldots,u_i(k+N_u-1|k)} J_i(k) = \sum_{s=1}^{N_p} q_{i,s} \left[\hat{y}_i(k+s \mid k) - y_{i,ref}(k+s)\right]^2 + \sum_{p=0}^{N_u-1} r_{i,p+1} u_i^2(k+p \mid k), \quad (5.18a)$$

subject to

$$u_{i,min} \leq u_i(k+p) \leq u_{i,max}; p = 0 \div N_u - 1, \quad (5.18b)$$

$$\Delta u_{i,min} \leq \Delta u_i(k+p) \leq \Delta u_{i,max}; p = 0 \div N_u - 1, \quad (5.18c)$$

$$y_{i,min} \leq \hat{y}_i(k+s \mid k) \leq y_{i,max}; s = 1 \div N_p, \quad (5.18d)$$

where $y_{i,ref}(k+s)$ is the set-point value of the output of zone i at sample time $(k+s)$ and $\Delta u_i(k+p)$ is the speed of change of the control input defined by $\Delta u_i(k+p) = u_i(k+p) - u_i(k+p-1)$. The coefficients $q_{i,s}$ and $r_{i,p+1}$ are positive weighting terms on tracking errors and input powers, respectively. From Equations 5.13 and 5.12, the local objective function in Equation 5.18 can be rewritten in a vector form as

$$\min_{U_i(k|k)} J_i(k) = \|\hat{Y}_i(k \mid k) - Y_{i,ref}(k)\|_{Q_i}^2 + \|U_i(k \mid k)\|_{R_i}^2 \quad (5.19a)$$

subject to

$$a_i U_i(k) \leq b_i(k) \quad (5.19b)$$

with $Q_i = diag(q_{i,1}, ..., q_{i,N_p})$, $R_i = diag(r_{i,1}, ..., r_{i,N_u})$, and $Y_{i,ref}(k) = \left[y'_{i,ref}(k+1), ..., y'_{i,ref}(k+N_p)\right]'$. a_i and $b_i(k)$ are determined by the following

$$b_i(k) = \begin{bmatrix} U_{i,max} \\ -U_{i,min} \\ Y_{i,max} - \Psi_i x_i(k) - \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j(k \mid k) \\ -Y_{i,min} + \Psi_i x_i(k) + \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j(k \mid k) \\ \Delta U_{i,max} \\ -\Delta U_{i,min} \end{bmatrix}, a_i = \begin{bmatrix} I_{N_u \times n_i} \\ -I_{N_u \times n_i} \\ \Phi_{i,i} \\ -\Phi_{i,i} \\ \Xi_i \\ -\Xi_i \end{bmatrix},$$

with $U_{i,max} = \left[u'_{i,max}, ..., u'_{i,max}\right]'$, $U_{i,min} = \left[u'_{i,min}, ..., u'_{i,min}\right]'$, $Y_{i,max} = \left[y'_{i,max}, ..., y'_{i,max}\right]'$, $Y_{i,min} = \left[y'_{i,min}, ..., y'_{i,min}\right]'$,

$$\Delta U_{i,max} = \begin{bmatrix} \Delta u_{i,max} + u_i(k-1) \\ \Delta u_{i,max} \\ \vdots \\ \Delta u_{i,max} \end{bmatrix}, \Delta U_{i,min} = \begin{bmatrix} \Delta u_{i,min} + u_i(k-1) \\ \Delta u_{i,min} \\ \vdots \\ \Delta u_{i,min} \end{bmatrix},$$

and

$$\Xi_i = \begin{bmatrix} I_{n_i} & 0 & \cdots & \cdots & 0 \\ -I_{n_i} & I_{n_i} & \ddots & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -I_{n_i} & I_{n_i} \end{bmatrix}_{N_u.n_i \times N_u.n_i}.$$

Consequently, a quadratic programming problem is derived as

$$\min_{U_i(k|k)} \frac{1}{2} U'_i(k \mid k) H_i U_i(k \mid k) + f'_i U_i(k \mid k) \tag{5.20a}$$

$$\text{subject to} \tag{5.20b}$$

$$a_i U_i(k \mid k) \leq b_i(k), \tag{5.20c}$$

where $H_i = \left(\Phi'_{i,i} Q_i \Phi_{i,i} + R_i\right)$, $f_i = \Phi'_{i,i} Q_i \left(\Psi_i x_i(k) + \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j(k-1 \mid k) - Y_{i,ref}(k)\right)$. At each sample time, local MPC controllers communicate the output predictions made at the previous sample. Based on this information, individual quadratic problems are solved providing the optimal input power sequence to each subsystem.

5.3.3.2 Iterative algorithm

To ensure optimal solution, iterative procedure can be used when communication resource is sufficient. At each sample time k, each local MPC controller executes five main actions: initializing, communicating with other controllers, solving the local optimization problem, checking the convergence, and applying the optimized control input to the corresponding subsystem [NRAD+12b, ZL07]. These steps are explained as follows

- Step 1: Initialization
  The local MPC controller i initializes its control input sequence $U_i(k \mid k)$ by the optimal one found at the previous sample like $U_i^{(l=0)}(k \mid k) = U_{i,opt}(k-1 \mid k-1)$, and output predictions by those of previous sample $\hat{Y}_i^{(l=0)}(k \mid k) = \hat{Y}_i(k-1 \mid k-1)$, l is the iteration index within a sampling period

- Step 2: Communication
  Local MPC controller i sends the predicted output $\hat{Y}_i^{(l)}(k \mid k)$ to the other controllers, and receives $\hat{Y}_j^{(l)}(k \mid k)$ for $j = 1 \div M$, $j \neq i$.

- Step 3: Optimization.
  MPC controller i solves its local optimal problem and obtains the optimal control input sequence for the iteration $l+1$.

$$U_i^{(l+1)}(k \mid k) = arg \min_{U_i(k|k)} \frac{1}{2} U_i'(k \mid k) H_i U_i(k \mid k) + (f_i^{(l)})' U_i(k \mid k) \tag{5.21a}$$

$$\text{subject to} \tag{5.21b}$$

$$a_i U_i(k \mid k) \leq b_i^{(l)}(k), \tag{5.21c}$$

where $f_i^{(l)} = \Phi_{i,i}' Q_i \left(\Psi_i x_i(k) + \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j^{(l)}(k \mid k) - Y_{i,ref}(k) \right)$, and

$$b_i^{(l)}(k) = \begin{bmatrix} U_{i,max} \\ -U_{i,min} \\ Y_{i,max} - \Psi_i x_i(k) - \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j^{(l)}(k \mid k) \\ -Y_{i,min} + \Psi_i x_i(k) + \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j^{(l)}(k \mid k) \\ \Delta U_{i,max} \\ -\Delta U_{i,min} \end{bmatrix}.$$

- Step 4: Checking and Updating
  The predicted outputs are updated with the obtained control input sequence as follows:

$$\hat{Y}_i^{(l+1)}(k \mid k) = \Psi_i x_i(k) + \Phi_{i,i} U_i^{(l+1)}(k \mid k) + \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j^{(l)}(k \mid k). \tag{5.22}$$

If $\|U_i^{(l+1)}(k \mid k) - U_i^{(l)}(k \mid k)\| \leq \epsilon_i$ for $i = 1 \div M$, where ϵ_i is the convergence precision, or the iteration index $l = l_{max}$ then MPC controllers end the iterations, otherwise $l = l + 1$, and the procedure goes back to Step 2.

- Step 5: Application of optimized control input
  Once the iteration is terminated, from the optimized control input sequence $U_{i,opt}(k)$, local MPC controller i applies the first action $u_{i,opt}(k) = N_i U_{i,opt}(k)$, with $N_i = \left[I_{n_i} 0_{n_i \times (N_u - 1)n_i}\right]$, to the corresponding zone.

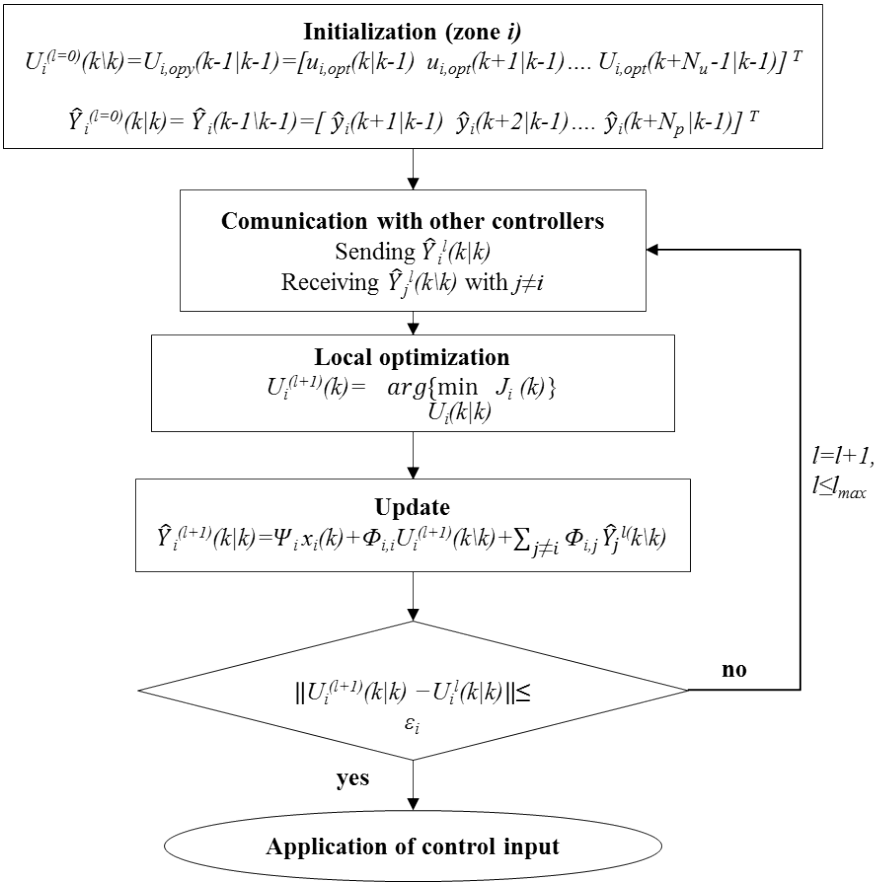

Figure 5.14: Diagram of the non-cooperative iterative algorithm

The steps of the proposed iterative algorithm are recapitulated in the Figure 5.14. The communication between MPC controllers is assured by the a supervisor at high level as shown in Figure 5.15. We shall note that the communications between local controllers are basically predictions of outputs and no control action of one is acknowledged to others.

5.3.3.3 Convergence analysis

For a unconstrained case, optimization problem in Equation 5.21 becomes

$$U_i^{(l)}(k \mid k) = arg \min_{U_i(k|k)} \frac{1}{2} U_i'(k \mid k) H_i U_i(k \mid k) + (f_i^{(l-1)})' U_i(k \mid k) \tag{5.23}$$

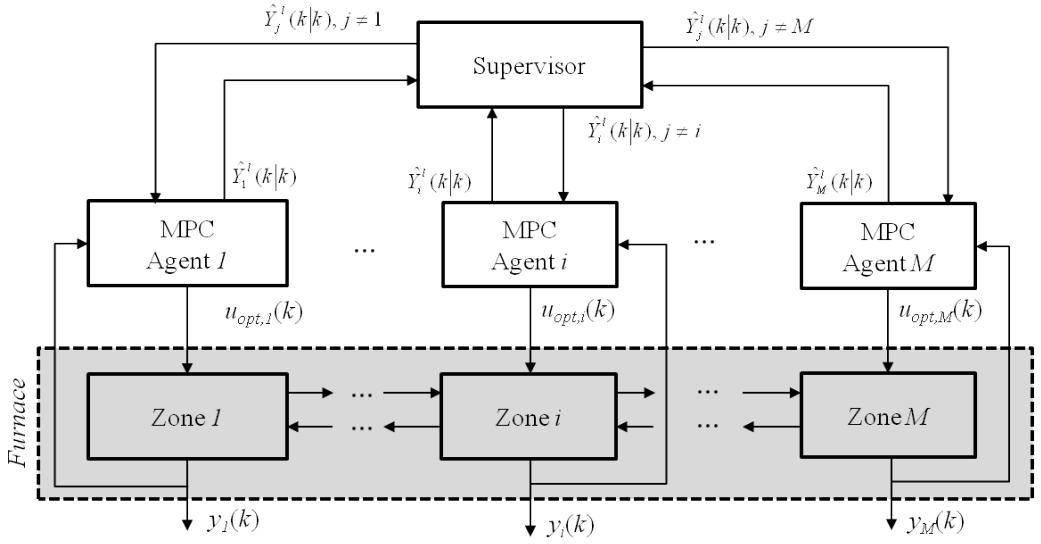

Figure 5.15: Communication between MPC controllers

with $H_i = \left(\Phi'_{i,i}Q_i\Phi_{i,i} + R_i\right)$, and $f_i^{(l-1)} = \Phi'_{i,i}Q_i\left(\Psi_i x_i(k) + \sum_{j\neq i}\Phi_{i,j}\hat{Y}_j^{(l-1)}(k \mid k) - Y_{i,ref}(k)\right)$. The result of unconstrained quadratic problem is given as

$$U_i^{(l)}(k \mid k) = \left(\Phi'_{i,i}Q_i\Phi_{i,i} + R_i\right)^{-1}\Phi'_{i,i}Q_i \times \left[-\Psi_i x_i(k) - \Phi_{i,j}\hat{Y}_j^{(l-1)}(k \mid k) + Y_{i,ref}(k)\right]. \tag{5.24}$$

For the overall system, we use notations $U^{(l)}(k \mid k) = \left[(U_1^{(l)})'(k \mid k), ..., (U_M^{(l)})'(k \mid k)\right]'$ and $\hat{Y}^{(l-1)}(k \mid k) = \left[(\hat{Y}_1^{(l-1)})'(k \mid k), ..., (\hat{Y}_M^{(l-1)})'(k \mid k)\right]'$, as a result the following relation is obtained

$$U^{(l)}(k \mid k) = \Pi(k) + \Gamma\hat{Y}^{(l-1)}(k \mid k), \tag{5.25}$$

where

$$\Gamma = \left[\begin{array}{c|c|c|c} 0 & -\gamma_1\Phi_{1,2} & \cdots & -\gamma_1\Phi_{1,M} \\ \hline -\gamma_2\Phi_{2,1} & 0 & \cdots & -\gamma_2\Phi_{2,M} \\ \hline \vdots & & \ddots & \vdots \\ \hline -\gamma_M\Phi_{M,1} & \cdots & -\gamma_M\Phi_{M,M-1} & 0 \end{array}\right], \gamma_i = \left(\Phi'_{i,i}Q_i\Phi_{i,i} + R_i\right)^{-1}\Phi'_{i,i}Q_i, \text{with } i = 1 \div M, \tag{5.26}$$

and

$$\begin{aligned} \Pi(k) &= \left[\begin{array}{c} \gamma_1\left(-\Psi_1 x_1(k) + Y_{1,ref}(k)\right) \\ \vdots \\ \gamma_M\left(-\Psi_M x_M(k) + Y_{M,ref}(k)\right) \end{array}\right], \\ &= Fx(k) + GY_{ref}(k) \end{aligned} \tag{5.27}$$

with $F = diag(-\gamma_1\Psi_1, ..., -\gamma_M\Psi_M)$, $G = diag(\gamma_1, ..., \gamma_M)$, $x(k) = [x'_1(k), ..., x'_M(k)]'$, and $Y_{ref}(k) = [Y'_{1,ref}(k), ..., Y'_{M,ref}(k)]'$.

In Equation 5.22, by replacing $l = l - 1$, we have

$$\hat{Y}_i^{(l)}(k \mid k) = \Psi_i x_i(k) + \Phi_{i,i} U_i^{(l)}(k \mid k) + \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j^{(l-1)}(k \mid k). \tag{5.28}$$

And for the whole system, it is written like

$$\hat{Y}^{(l)}(k \mid k) = \Omega(k) + DU^{(l)}(k \mid k) + E\hat{Y}_j^{(l-1)}(k \mid k), \tag{5.29}$$

with $\Omega(k) = diag(\Psi_1, ..., \Psi_M)x(k) = Jx(k)$, $D = diag(\Phi_{1,1}, ..., \Phi_{M,M})$, and

$$E = \left[\begin{array}{c|c|c|c} 0 & \Phi_{1,2} & \cdots & \Phi_{1,M} \\ \hline \Phi_{2,1} & 0 & \cdots & \Phi_{2,M} \\ \hline \vdots & & \ddots & \vdots \\ \hline \Phi_{M,1} & \cdots & \Phi_{M,M-1} & 0 \end{array}\right].$$

From Equations 5.25 and 5.29, the recursive equation of system outputs are given as:

$$\begin{array}{rcl} \hat{Y}^{(l)}(k \mid k) & = & \Omega(k) + D\left(\Pi(k) + \Gamma\hat{Y}^{(l-1)}(k \mid k)\right) + E\hat{Y}^{(l-1)}(k \mid k) \\ & = & \Omega(k) + D\Pi(k) + (D\Gamma + E)\,\hat{Y}^{(l-1)}(k \mid k). \end{array} \tag{5.30}$$

The convergence of $\hat{Y}^{(l)}(k \mid k)$ determines the convergence of the algorithm. In the recursive Equation 5.30, $\Omega(k)$ and $\Pi(k)$ do not depend on the iterative index l, therefore the convergence condition of $\hat{Y}^{(l)}(k \mid k)$ is given as

$$\|\rho\,(D\Gamma + E)\| < 1 \tag{5.31}$$

where the spectral radius $\rho\,(D\Gamma + E)$ is the maximum eigenvalues of the matrix $(D\Gamma + E)$.

5.3.3.4 Nominal stability analysis

From the state space representation of subsystem i in Equation 5.10, the overall system is constructed by gathering the states, inputs, outputs and matrices of subsystems as $x(k) = [x_1'(k) \cdots x_M'(k)]'$, $u(k) = [u_1'(k) \cdots u_M'(k)]'$, $y(k) = [y_1'(k) \cdots y_M'(k)]'$, $A = diag(A_1, ..., A_M)$, $B_u = diag(B_1, ..., B_M)$, $C = diag(C_1, ..., C_M)$. Then we have the state space representation of the overall system described as

$$\begin{array}{rcl} x(k+1) & = & (A + B_y C)x(k) + B_u u(k) \\ y(k) & = & Cx(k) \end{array} \tag{5.32}$$

where

$$B_y = \left[\begin{array}{c|c|c|c|c|c} 0 & B_{1,2} & \dots & \dots & \dots & B_{1,M} \\ \hline B_{2,1} & 0 & B_{2,3} & \dots & \dots & B_{2,M} \\ \hline \vdots & \ddots & \ddots & \ddots & & \vdots \\ \hline \vdots & & \ddots & \ddots & \ddots & \vdots \\ \hline B_{M-1,1} & \dots & \dots & B_{M-1,M-2} & 0 & B_{M-1,M} \\ \hline B_{M,1} & B_{M,2} & \dots & \dots & B_{M-1,M} & 0 \end{array}\right].$$

If the convergence condition in 5.31 is satisfied, $\hat{Y}^{(l)}(k \mid k)$ converges to a final value $\hat{Y}^*(k \mid k)$ and Equation 5.30 becomes

$$\begin{aligned} \hat{Y}^*(k \mid k) = &\ \Omega(k) + D\Pi(k) + (D\Gamma + E)\,\hat{Y}^*(k \mid k) \\ \Rightarrow\ \hat{Y}^*(k \mid k) = &\ \underbrace{[I - D\Gamma - E]^{-1}}_{\Upsilon}[\Omega(k) + D\Pi(k)] \end{aligned} . \tag{5.33}$$

From Equations 5.33, 5.27 and 5.25, the optimal control input sequence of the whole system $U_{opt}(k)$ is given as:

$$\begin{aligned} U_{opt}(k \mid k) &= \Pi(k) + \Gamma\hat{Y}^*(k \mid k) \\ &= \Pi(k) + \Gamma\Upsilon\left[\Omega(k) + D\Pi(k)\right] \\ &= \Gamma\Upsilon\Omega(k) + (I + \Gamma\Upsilon D)\,\Pi(k) \\ &= \Gamma\Upsilon Jx(k) + (I + \Gamma\Upsilon D)\left[Fx(k) + GY_{ref}(k)\right] \\ &= \underbrace{\left[\Gamma\Upsilon J + (I + \Gamma\Upsilon D)\,F\right]}_{V} x(k) + \underbrace{(I + \Gamma\Upsilon D)\,G}_{P} Y_{ref}(k) \\ &= Vx(k) + PY_{ref}(k). \end{aligned} \tag{5.34}$$

The control input to be applied at sample time k

$$u(k \mid k) = NU_{opt}(k \mid k) = NVx(k) + NPY_{ref}(k) \tag{5.35}$$

where $N = diag(N_1, ..., N_M), N_i = \left[I_{n_i} 0_{n_i \times (N_u - 1)n_i}\right]$. In fact, $x(k)$ is only obtained through state observer, therefore, optimal control input in Equation 5.35 is actually calculated by

$$\begin{aligned} u(k \mid k) &= NV\bar{x}(k) + NPY_{ref}(k) \\ &= NV(x(k) + \tilde{x}(k)) + NPY_{ref}(k) \end{aligned} \tag{5.36}$$

As a result, the dynamic of closed-loop system-wide is defined by

$$\begin{aligned} x(k+1) &= (A + B_yC)x(k) + B_u u(k) \\ &= (A + B_yC)x(k) + B_u\left[NVx(k) + NV\tilde{x}(k) + NPY_{ref}(k)\right] \\ &= (A + B_yC + B_uNV)x(k) + B_uNV\tilde{x}(k) + B_uNPY_{ref}(k) \end{aligned} . \tag{5.37}$$

To analyze the stability of a subsystem and its state observer, we have to add an extended state which is the state estimation error as provided in Equation 5.16 as $\tilde{x}_i(k+1) = (A_i - L_iC_i)\,\tilde{x}_i(k)$. For the overall system, the

extended state is $\tilde{x}(k+1) = [\tilde{x}_1'(k+1), ..., \tilde{x}_M'(k+1)]'$ which can be expressed as

$$\tilde{x}(k+1) = (A - LC)\,\tilde{x}(k) \tag{5.38}$$

where $L = diag(L_1, ..., L_M)$. The dynamic of closed-loop system including distributed MPC controllers and state observers are given as follows

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} A + B_y C + B_u NV & B_u NV \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} B_u NP \\ 0 \end{bmatrix} Y_{ref}(k). \tag{5.39}$$

Therefore, the stability conditions in this case is

$$eig\left(\begin{bmatrix} A + B_y C + B_u NV & B_u NV \\ 0 & A - LC \end{bmatrix} \right) < 1 \tag{5.40}$$

That is, the eigenvalues must be all inside the unit circle to guarantee the nominal stability of the system.

5.3.4 Serial cooperative distributed MPC

As a furnace zone i interacts only with two neighboring zones ($i \pm 1$), and adding to that, for the considered furnace, $H_{2,1}, H_{3,2} = 0$, the local MPC controller i can optimize the performance of a neighborhood of three zones ($i, i+1, i-1$). That is, local controllers not only communicate with each other to take into account the coupling effects, but also compromise between individual performance of each zone to achieve a best solution for the entire neighborhood.

First, the formulation of objective function is given. Second, an iterative algorithm is presented. However, due to high complexity, the corresponding convergence analysis and nominal stability of closed-loop will not be examined as in non-cooperative strategy.

5.3.4.1 Problem formulation

Optimization problem for a local MPC controller i is defined based on performance of its corresponding subsystem and two neighboring subsystems $i \pm 1$ as

$$\min_{U_i(k|k)} J_{i\pm}(k) = \sum_{j=i-1}^{i+1} \|\hat{Y}_j(k \mid k) - Y_{j,ref}(k)\|_{Q_j}^2 + \|U_i(k \mid k)\|_{R_i}^2 \tag{5.41a}$$

subject to

$$u_{i,min} \leq u_i(k+p) \leq u_{i,max}; p = 0, ..., N_u - 1, \tag{5.41b}$$

$$\Delta u_{i,min} \leq \Delta u_i(k+p) \leq \Delta u_{i,max}; p = 0, ..., N_u - 1, \tag{5.41c}$$

$$y_{j,min} \leq \hat{y}_j(k+s \mid k) \leq y_{j,max}; s = 1, ..., N_p, j = i, i+1, i-1. \tag{5.41d}$$

with $Q_j = diag(q_{j,1}, ..., q_{j,N_p})$, and $R_i = diag(r_{i,1}, ..., r_{i,N_u})$. Now, we want to formulate $J_{i\pm}(k)$ as function of $U_i(k \mid k)$. Let's consider $\hat{Y}_{i+1}(k \mid k)$ by substituting Equation 5.14 as

$$\begin{aligned}
\hat{Y}_{i+1}(k \mid k) =& \Psi_{i+1}x_{i+1}(k) + \Phi_{i+1,i+1}U_{i+1}(k \mid k) + \sum_{j\neq i+1} \Phi_{i+1,j}\hat{Y}_j(k \mid k) && (5.42a)\\
=& \overbrace{\Psi_{i+1}x_{i+1}(k) + \Phi_{i+1,i+1}U_{i+1}(k \mid k) + \sum_{j\neq i,i+1} \Phi_{i+1,j}\hat{Y}_j(k \mid k)}^{\Theta_{i+1}(k)} + \\
&+ \Phi_{i+1,i}\hat{Y}_i(k \mid k) && (5.42b)\\
=& \Theta_{i+1}(k) + \Phi_{i+1,i}\hat{Y}_i(k \mid k) && (5.42c)\\
=& \Theta_{i+1}(k) + \Phi_{i+1,i}[\Psi_i x_i(k) + \Phi_{i,i}U_i(k \mid k) + \sum_{j\neq i+1} \Phi_{i,j}\hat{Y}_j(k \mid k)] && (5.42d)\\
=& \Phi_{i+1,i}\Phi_{i,i}U_i(k \mid k) + \underbrace{\Theta_{i+1}(k) + \Phi_{i+1,i}[\Psi_i x_i(k) + \sum_{j\neq i+1} \Phi_{i,j}\hat{Y}_j(k \mid k)]}_{\Lambda_{i+1}(k)}. && (5.42e)
\end{aligned}$$

In a similar approach, $\hat{Y}_{i-1}(k \mid k)$ can be formulated as

$$\hat{Y}_{i-1}(k \mid k) = \Phi_{i-1,i}\Phi_{i,i}U_i(k \mid k) + \Lambda_{i-1}(k) \tag{5.43}$$

where $\Lambda_{i-1}(k) = \Theta_{i-1}(k) + \Phi_{i-1,i}[\Psi_i x_i(k) + \sum_{j\neq i-1} \Phi_{i,j}\hat{Y}_j(k \mid k)]$. The objective function is now expressed by

$$\begin{aligned}
J_{i\pm}(k) =& \|\Phi_{i-1,i}\Phi_{i,i}U_i(k \mid k) + \overbrace{\Lambda_{i-1}(k) - Y_{i-1,ref}(k)}^{\alpha_{i-1}(k)}\|^2_{Q_{i-1}} + \|U_{i-1}(k \mid k)\|^2_{R_{i-1}} && (5.44a)\\
&+ \|\Phi_{i+1,i}\Phi_{i,i}U_i(k \mid k) + \overbrace{\Lambda_{i+1}(k) - Y_{i+1,ref}(k)}^{\alpha_{i+1}(k)}\|^2_{Q_{i+1}} + \|U_{i+1}(k \mid k)\|^2_{R_{i+1}}\\
&+ \|\Phi_{i,i}U_i(k \mid k) + \underbrace{\Psi_i x_i(k) + \sum_{j\neq i} \Phi_{i,j}\hat{Y}_j(k \mid k) - Y_{i,ref}(k)}_{\beta_i(k)}\|^2_{Q_i} + \|U_i(k \mid k)\|^2_{R_i} && (5.44b)\\
=& U_i'(k \mid k)H_{i\pm}U_i(k \mid k) + 2f'_{i\pm}U_i(k \mid k) + g_{i\pm}, && (5.44c)
\end{aligned}$$

where $H_{i\pm} = [\Phi'_{i,i}(\Phi'_{i-1,i}Q_{i-1}\Phi_{i-1,i} + \Phi'_{i+1,i}Q_{i+1}\Phi_{i+1,i} + Q_i)\Phi_{i,i} + R_i]$, $f_{i\pm} = \Phi'_{i,i}(\Phi'_{i-1,i}Q_{i-1}\alpha_{i-1}(k) + \Phi'_{i+1,i}Q_{i+1}\alpha_{i+1}(k) + Q_i\beta_i(k))$, and $g_{i\pm} = \alpha'_{i-1}(k)Q_{i-1}\alpha_{i-1}(k) + \alpha'_{i+1}(k)Q_{i+1}\alpha_{i+1}(k) + \beta'_i(k)Q_i\beta_i(k) + \|U_{i+1}(k \mid k)\|^2_{R_{i+1}} + \|U_{i-1}(k \mid k)\|^2_{R_{i-1}}$. As a result, at sample time k, the optimization problem of Equation 5.41 becomes a quadratic programming problem as

$$\min_{U_i(k|k)} \frac{1}{2}U_i'(k \mid k)H_{i\pm}U_i(k \mid k) + f'_{i\pm}U_i(k \mid k) \tag{5.45a}$$

subject to

$$a_{i\pm}U_i(k \mid k) \leq b_{i\pm}(k), \tag{5.45b}$$

where

$$b_{i\pm}(k) = \begin{bmatrix} U_{i,max} \\ -U_{i,min} \\ Y_{i,max} - \Psi_i x_i(k) - \sum_{j\neq i} \Phi_{i,j}\hat{Y}_j(k \mid k) \\ -Y_{i,min} - \Psi_i x_i(k) + \sum_{j\neq i} \Phi_{i,j}\hat{Y}_j(k \mid k) \\ Y_{i+1,max} - \Lambda_{i+1}(k) \\ -Y_{i+1,min} + \Lambda_{i+1}(k) \\ Y_{i-1,max} - \Lambda_{i-1}(k) \\ -Y_{i-1,min} + \Lambda_{i-1}(k) \\ \Delta U_{i,max} \\ -\Delta U_{i,min} \end{bmatrix}, a_{i\pm} = \begin{bmatrix} I_{N_u \times n_{u_i}} \\ -I_{N_u \times n_{u_i}} \\ \Phi_{i,i} \\ -\Phi_{i,i} \\ \Phi_{i+1,i}\Phi_{i,i} \\ -\Phi_{i+1,i}\Phi_{i,i} \\ \Phi_{i-1,i}\Phi_{i,i} \\ -\Phi_{i-1,i}\Phi_{i,i} \\ \Xi_i \\ -\Xi_i \end{bmatrix},$$

with $\Xi_i, \Delta U_{i,max}, \Delta U_{i,min}, Y_{i+1,max}, Y_{i+1,min}$ defined in Equation 5.19.

5.3.4.2 Iterative algorithm

Assume that the communication resource is sufficient, then an iterative algorithm should be designed to ensure the optimal solution of the problem 5.45. At each sample time k, each local MPC controller undergoes the procedures of initialization, communication with other controllers, solving the local optimal problem, checking the convergence and application of optimized control input to the corresponding subsystem, i.e.,

- Step 1: Initialization
  The local MPC controller i initializes its control input sequence $U_i(k \mid k)$ by the optimal one found at the previous sample like $U_i^{(l=0)}(k \mid k) = U_{i,opt}(k-1)$, the output prediction $\hat{Y}_i(k-1 \mid k)$ by the prediction of previous sample as $\hat{Y}_i^{(l=0)}(k \mid k) = \hat{Y}_i(k-1 \mid k-1)$, l is the iteration index.

- Step 2: Communication
  For iteration l, local MPC controllers exchange their predicted output $\hat{Y}_m^{(l)}(k \mid k)$, current state $x_m(k)$, and control input sequence $U_m^{(l)}(k \mid k)$ within a neighborhood. Therefore, controller i receives $\hat{Y}_j^{(l)}(k \mid k)$, $U_j^{(l)}(k \mid k)$, and $x_j(k)$ with $j = i \pm 1$ to calculate necessary terms of quadratic programming problem in Equation 5.45.

- Step 3: Optimization.
  Each MPC controller solves its local quadratic problem expressed in Equation 5.45 and obtains the new control input sequence for the iteration.

$$U_i^{(l+1)}(k \mid k) = arg \min_{U_i(k|k)} \frac{1}{2} U_i'(k \mid k) H_{i\pm} U_i(k \mid k) + (f_{i\pm}^{(l)})' U_i(k \mid k) \tag{5.46a}$$

subject to

$$a_{i\pm} U_i(k \mid k) \leq b_{i\pm}^{(l)}(k), \tag{5.46b}$$

- Step 4 :Updating and Checking
  The predicted outputs are updated with the new control input sequence as

$$\hat{Y}_i^{(l+1)}(k \mid k) = \Psi_i x_i(k) + \Phi_{i,i} U_i^{(l+1)}(k \mid k) + \sum_{j \neq i} \Phi_{i,j} \hat{Y}_j^{(l)}(k \mid k). \tag{5.47}$$

  If $|U_i^{(l+1)}(k \mid k) - U_i^{(l)}(k \mid k)| \leq \epsilon_i$ for $i = 1 \div M$, where ϵ_i is the convergence precision, or the iteration index $l = l_{max}$ then MPC controllers end the iterations and $U_{i,opt}(k) = U_i^{(l+1)}(k \mid k)$. Otherwise $l = l+1$, and the procedure goes back to *Step 2*.

- Step 5: Application of optimized control input
  Once the iteration is terminated, from the optimized control input sequence $U_{i,opt}(k)$, local MPC controller i applies the first action $u_{i,opt}(k) = N_i U_{i,opt}(k)$ to the corresponding zone.

The communication between local controllers is illustrated in Figure 5.16. In contrast to non-cooperative distributed MPC, local controllers communicate not only their output predictions but also system states, and control input sequences in order to calculate objective functions associated to control performance of other zones. Nevertheless, the communication is restricted between neighbor zones, which helps reducing the resource requirements.

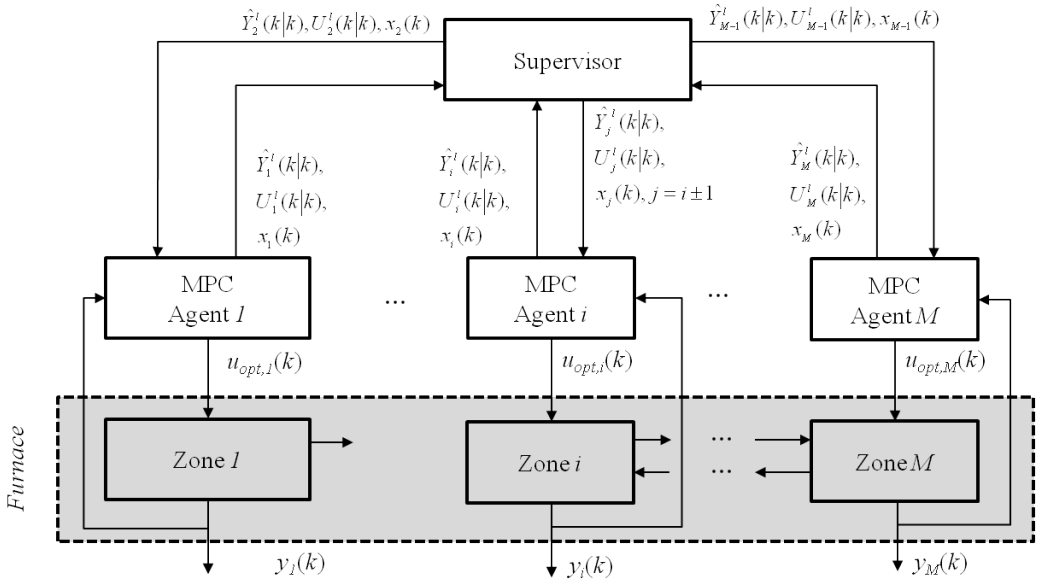

Figure 5.16: Diagram of cooperative distributed MPC algorithm

5.3.5 Centralized MPC as a benchmark

From the state space representation of the overall system given in Equation 5.32, output can be predicted with the following formula

$$\hat{y}(k+s \mid k) = C(A+B_yC)^s x(k) + \sum_{p=1}^{\min(s,N_u)} C(A+B_yC)^{s-p} B_u u(k+p-1 \mid k), \tag{5.48}$$

with $s = 1 : N_p$. The optimization problem of centralized MPC is expressed as

$$\min_{u(k|k),\ldots,u(k+N_u-1|k)} J(k) = \sum_{s=1}^{N_p} q_s \left[\hat{y}(k+s \mid k) - y_{ref}(k+s)\right]^2 + \sum_{p=0}^{N_u-1} r_{p+1} u^2(k+p \mid k), \tag{5.49a}$$

subject to

$$u_{min} \leq u(k+p) \leq u_{max}; p = 0 \div N_u - 1, \tag{5.49b}$$

$$\Delta u_{min} \leq \Delta u(k+p) \leq \Delta u_{max}; p = 0 \div N_u - 1, \tag{5.49c}$$

$$y_{min} \leq \hat{y}(k+s \mid k) \leq y_{max}; s = 1 \div N_p, \tag{5.49d}$$

In a vector form, output prediction of the system-wide is written as

$$\hat{Y}(k \mid k) = \left[\hat{y}'(k+1 \mid k), \ldots, \hat{y}'(k+N_p \mid k)\right]' \tag{5.50a}$$

$$= \Psi x(k) + \Phi U(k \mid k), \tag{5.50b}$$

where $U(k \mid k) = [u'(k \mid k), \ldots, u'(k+N_u-1 \mid k)]'$ is the control input sequence at time instant k, $\Psi = \left[(C(A+B_yC))', \ldots, (C(A+B_yC)^{N_p})'\right]'$, and

$$\Phi = \begin{bmatrix} CB_u & \cdots & 0 \\ \vdots & \ddots & \vdots \\ C(A+B_yC)^{N_u-1}B_u & \cdots & CB_u \\ \vdots & \vdots & \vdots \\ C(A+B_yC)^{N_p-1}B_u & \cdots & C(A+B_yC)^{N_p-N_u}B_u \end{bmatrix}.$$

The optimization problem in Equation 5.49 is rewritten in vector form as

$$\min_{U(k|k)} J(k) = \| \hat{Y}(k \mid k) - Y_{ref}(k) \|_Q^2 + \| U(k \mid k) \|_R^2 \tag{5.51a}$$

subject to

$$U_{min} \leq U(k \mid k) \leq U_{max}, \tag{5.51b}$$

$$\Delta U_{min} \leq \Delta U(k \mid k) \leq \Delta U_{max}, \tag{5.51c}$$

$$Y_{min} \leq \hat{Y}(k \mid k) \leq Y_{max}. \tag{5.51d}$$

From Equation5.50, the objective function can be expressed in form of quadratic function as

$$J(k) = \parallel \hat{Y}(k \mid k) - Y_{ref}(k) \parallel_Q^2 + \parallel U(k \mid k) \parallel_R^2 \tag{5.52a}$$

$$= \parallel \Psi x(k) + \Phi U(k \mid k) - Y_{ref}(k) \parallel_Q^2 + \parallel U(k \mid k) \parallel_R^2 \tag{5.52b}$$

$$= U'(k \mid k)(\underbrace{(\Phi' Q \Phi + R)}_{H} U(k \mid k) + 2[\underbrace{\Phi' Q(\Psi x(k) - Y_{ref}(k)}_{f}]' U(k \mid k) + \tag{5.52c}$$

$$+ (\Psi x(k) - Y_{ref}(k))' Q(\Psi x(k) - Y_{ref}(k)) \tag{5.52d}$$

$$= U'(k \mid k) H U(k \mid k) + 2f' U(k \mid k) + (\Psi x(k) - Y_{ref}(k))' Q(\Psi x(k) - Y_{ref}(k)). \tag{5.52e}$$

As a result, the optimization problem of Equation 5.53 can be solved with a quadratic programming optimization

$$\min_{U(k|k)} J(k) = U'(k \mid k) H U(k \mid k) + 2f' U(k \mid k) \tag{5.53a}$$

subject to

$$aU(k \mid k) \leq b(k), \tag{5.53b}$$

where

$$b(k) = \begin{bmatrix} U_{max} \\ -U_{min} \\ Y_{max} - \Psi x(k) \\ -Y_{min} - \Psi x(k) \\ \Delta U_{i,max} \\ -\Delta U_{i,min} \end{bmatrix}, a = \begin{bmatrix} I_{N_u \times n} \\ -I_{N_u \times n} \\ \Phi \\ -\Phi \\ \Xi \\ -\Xi \end{bmatrix},$$

and

$$\Xi = \begin{bmatrix} I_n & 0 & \cdots & \cdots & 0 \\ -I_n & I_n & \ddots & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -I_n & I_n \end{bmatrix}_{N_u.n \times N_u.n}.$$

For an unconstrained case, solution of optimization problem 5.53 is calculated by

$$U_{opt}(k) = H^{-1} f \tag{5.54}$$

and the control input applied at sample time k is $u_{opt}(k) = NU_{opt}(k)$ with N given in 5.35. For output feedback control $u_{opt}(k)$ is actually calculated as

$$\begin{aligned} u_{opt}(k) &= NU_{opt}(k) \\ &= NH^{-1}f\bar{x}(k) && \text{(5.55a)} \\ &= NH^{-1}f\left[x(k) + \tilde{x}(k)\right] && \text{(5.55b)} \end{aligned}$$

The whole closed-loop system with centralized MPC controller and state observers is determined by the state space representation of overall system in Equation 5.32 and the dynamic of state estimation error $\tilde{x}(k+1) = (A - LC)\tilde{x}(k)$ given in Equation 5.38. With the control solution in Equation 5.55, the closed-loop can be written as

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} A + B_yC + B_uNH^{-1}f & B_uNH^{-1}f \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix}. \tag{5.56}$$

The closed-loop stability condition is therefore

$$eig\left(\begin{bmatrix} A + B_yC + B_uNH^{-1}f & B_uNH^{-1}f \\ 0 & A - LC \end{bmatrix}\right) < 1. \tag{5.57}$$

5.4 Simulation results

We want to examine the performance of the proposed non-cooperative and serial cooperative algorithms. Because in a networked system where subsystems interacts weakly the difference between the non-cooperative method and the cooperative method will not be significant. Therefore, a networked system where three subsystems strongly interact with each other through outputs is constructed in order to clarify difference between two methods. The proposed non-cooperative and cooperative distributed MPC algorithms are applied to solve tracking problem of step signals. The results are compared with the centralized MPC to verify the optimality of distributed MPC methods. Afterwards, the algorithms are applied to zone temperature control of the considered walking-beam reheating furnace. The PID controllers are also simulated for zone temperature control so that the results can be appropriately compared.

5.4.1 Illustrative example

Consider a system composed of three subsystems given in the following

$$\text{subsystem 1:} \begin{cases} x_1(k+1) &= \begin{bmatrix} 0.15 & 0 & 1 \\ 0 & 0.23 & 0.6 \\ 0 & 0 & 0.4 \end{bmatrix} x_1(k) + \begin{bmatrix} 0.41 \\ 0.53 \\ 0.62 \end{bmatrix} u_1(k) + \begin{bmatrix} 0.03 & 0.01 \\ 0.1 & 0.03 \\ 0.05 & 0.2 \end{bmatrix} \begin{bmatrix} y_2(k) \\ y_3(k) \end{bmatrix} \\ y_1(k) &= \begin{bmatrix} 0.2 & 0.16 & 0.25 \end{bmatrix} x_1(k) \end{cases} \tag{5.58}$$

$$\text{subsystem 2:} \begin{cases} x_2(k+1) &= \begin{bmatrix} 0.4 & 0.16 \\ 0 & 0.25 \end{bmatrix} x_2(k) + \begin{bmatrix} 0.6 \\ 0.2 \end{bmatrix} u_2(k) + \begin{bmatrix} 0.4 & 0.4 \\ 0.1 & 0.3 \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_3(k) \end{bmatrix} \\ y_2(k) &= \begin{bmatrix} 0.08 & 0.09 \end{bmatrix} x_2(k) \end{cases} \tag{5.59}$$

$$\text{subsystem 3:} \begin{cases} x_3(k+1) &= \begin{bmatrix} 0.7 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix} x_1(k) + \begin{bmatrix} 0.16 \\ 0.27 \\ 0.13 \\ 0.31 \end{bmatrix} u_3(k) + \begin{bmatrix} 0.1 & 0.02 \\ 0.3 & 0.09 \\ 0.2 & 0.3 \\ 0.4 & 0.5 \end{bmatrix} \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \\ y_3(k) &= \begin{bmatrix} 0.1 & 0.2 & 0.1 & 0.3 \end{bmatrix} x_3(k) \end{cases} \tag{5.60}$$

With the proposed non-cooperative distributed MPC algorithm, the local performance index for each subsystem

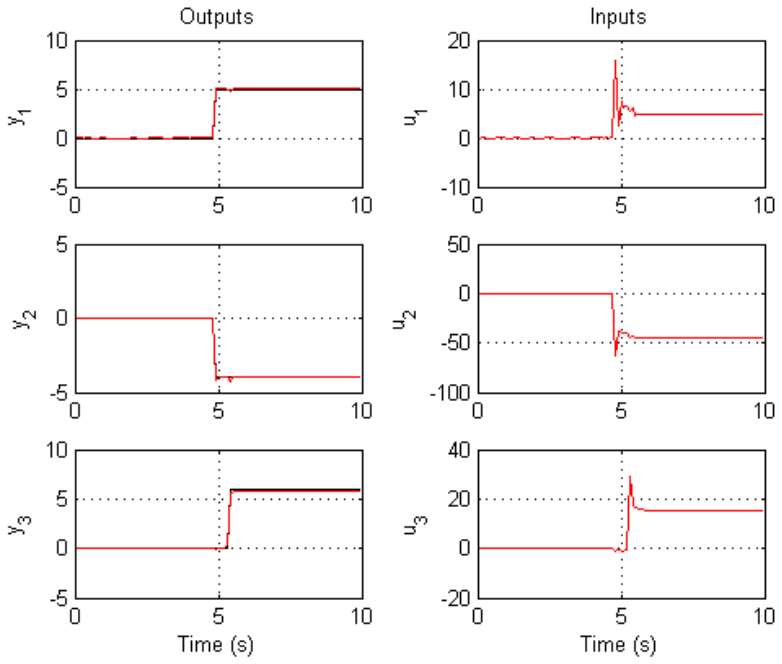

Figure 5.17: Results of non-cooperative distributed MPC

is

$$J_i(k) = \sum_{s=1}^{N_p} q_i \left[\hat{y}_i(k+s \mid k) - y_{i,ref}(k+s)\right]^2 + \sum_{p=1}^{N_u} r_i u_i^2(k+p-1 \mid k), i = 1 \div 3. \tag{5.61}$$

With regards to centralized MPC, the performance index for overall system will be

$$J(k) = J_1(k) + J_2(k) + J_3(k). \tag{5.62}$$

The tuning parameters for each subsystem are set with $N_p = 10, N_u = 5, q_i = 1, r_1 = 10^{-3}, r_2 = 10^{-4}, r_3 = 4.10^{-3}$ with $(i = 1 \div 3)$, using sampling time $T_s = 0.1$ s, and convergence precision $\epsilon_i = 0.01$ $(i = 1 \div 3)$. Simulation on Matlab is carried out to evaluate the proposed distributed MPC algorithms through comparison with centralized MPC. As shown in Figures 5.17, 5.18 and 5.19, distributed MPC gives good performance in

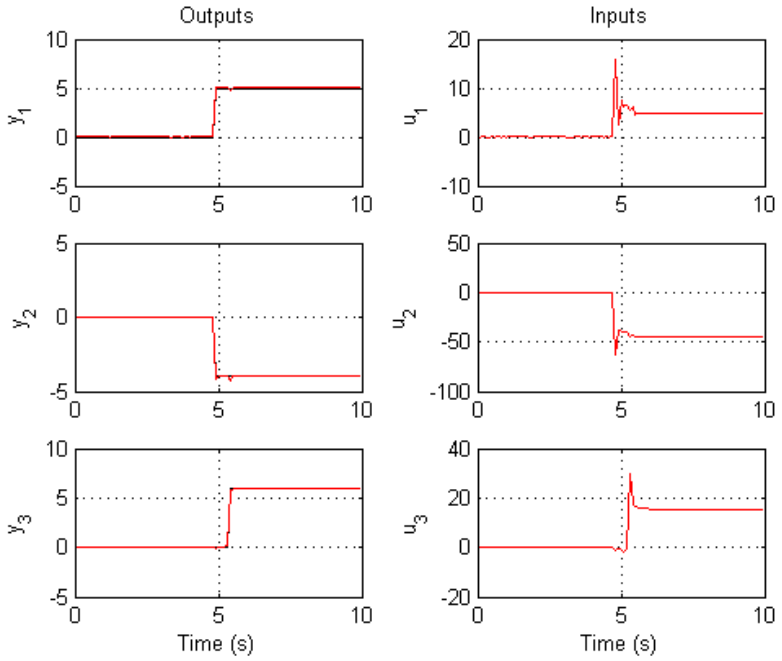

Figure 5.18: Results of cooperative distributed MPC

responses to step signals. The overall performance index $J(k)$ is shown in Figure 5.21. The distributed MPC algorithms converge and assure the stability of the overall system. The number of iterations through time is

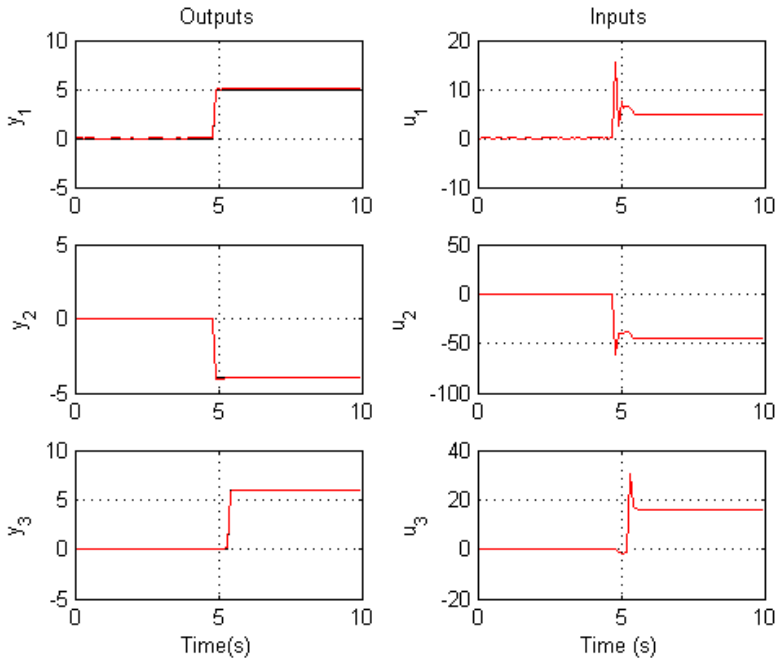

Figure 5.19: Output and control signals under centralized MPC

shown in Figure 5.20. We see that there is little different between cooperative and non-cooperative distributed MPC in term of convergence of iterative algorithm.

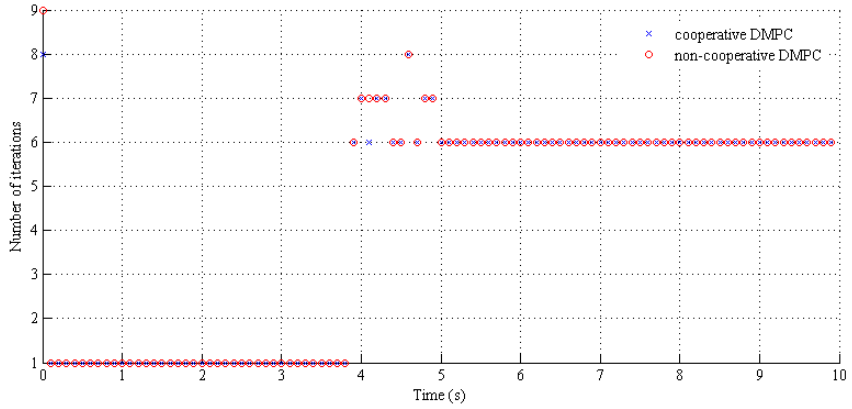

Figure 5.20: Number of iteration

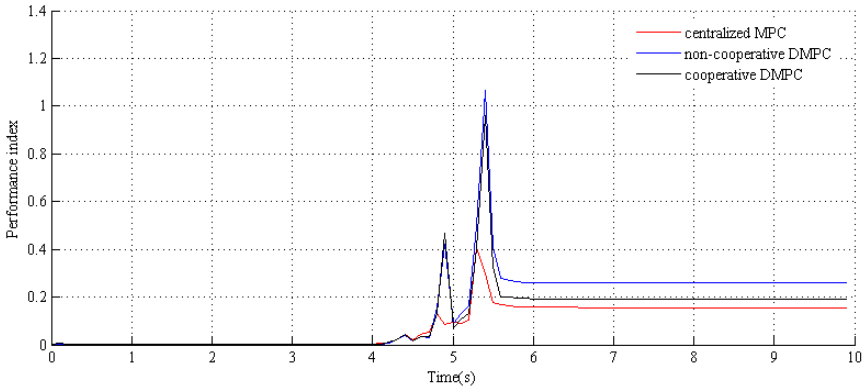

Figure 5.21: Performance index of overall system

5.4.2 Control of walking beam reheating furnace

The proposed distributed MPC algorithms for a networked system with subsystems coupled with outputs have been proved to be efficient with the illustrative example. Because the furnace zones are not fully coupled we will utilize only non-cooperative distributed MPC to compare with the PID controllers which is currently used. Here, the output coupling model is provided as:

$$Y_1(s) = \frac{0.0143e^{-60s}}{1+605s}U_1(s) + \frac{0.210e^{-20s}}{1+500s}Y_2(s), \tag{5.63a}$$

$$Y_2(s) = \frac{0.012e^{-40s}}{1+742s}U_2(s) + \frac{0.16e^{-20s}}{1+416s}Y_3(s), \tag{5.63b}$$

$$Y_3(s) = \frac{0.0066e^{-70s}}{1+415s}U_3(s). \tag{5.63c}$$

It shall be noted that this model of furnace zone does not have full couplings between subsystems. The interactions are mainly in the direction of gas flow inside the furnace and considerable only between two neighbor zones. Therefore, coupling effects between zones preheating (y_1) and heating 2 (y_3) are neglected. Similarly, influences of y_2 on y_3, y_1 on y_2 are obsoleted. In order to use this model for design of MPC controllers, the

delays are approximated by $Pade$ method as

$$e^{-\tau s} = \frac{1}{1+\tau s}. \tag{5.64}$$

The interaction model of furnace zone is constituted from second order transfer functions, i.e.,

$$Y_1(s) = \frac{1}{1+60s}\frac{0.0143}{1+605s}U_1(s) + \frac{1}{1+20s}\frac{0.210}{1+500s}Y_2(s) \tag{5.65a}$$

$$Y_2(s) = \frac{1}{1+40s}\frac{0.012}{1+742s}U_2(s) + \frac{1}{1+20s}\frac{0.16}{1+416s}Y_3(s) \tag{5.65b}$$

$$Y_3(s) = \frac{1}{1+70s}\frac{0.0066}{1+415s}U_3(s) \tag{5.65c}$$

This model can be transformed into state space representation then discretized with sample time $T_s = 20$. The dimension of the obtained model is 6. The model of Equation 5.63 corresponds with operating point defined

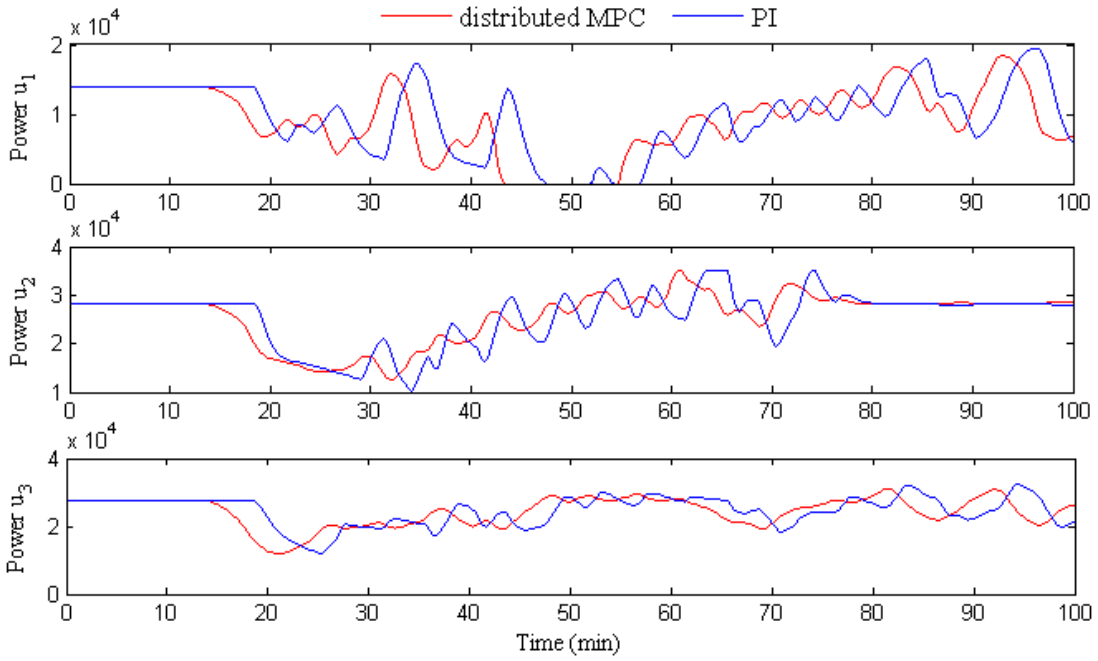

Figure 5.22: Input power of furnace zones

by nominal zone powers ($P_{1,nom}$, $P_{2,nom}$, $P_{3,nom}$)=(12, 20, 20) MW and nominal zone temperature ($T_{1,nom}$, $T_{2,nom}$, $T_{3,nom}$)=(1000, 1150, 1300)°C. The tuning of parameters for distributed MPC is set prediction horizon $N_p = 15$, control horizon $N_u = 5$, limit of iterations $l_{max} = 10$, and convergence coefficient $\epsilon = 0.1$.

In general, a trade-off between tracking performance and power consumption must be fulfilled with a proper choice of weighting matrices. For an energy reduction mode, the control efforts are minimized while sustaining temperature tracking satisfaction. On the contrary, for a tracking mode, the tracking errors are minimized as much as possible which will cost more control effort. In Figures 5.22, it is noticed that distributed MPC does not have delay on control action as seen with PID controllers, this can be explained due to anticipation feature of MPC strategy. Moreover, tracking performance with distributed MPC can be visually observed in Figure 5.23 to

Zone	Average temperature error(°C)			Power consumption(Watt)		
	PID	DMPC	Improvement	PID	DMPC	Gain
Preaheating	8	6	2	9572	9163	4.87%
Heating 1	6	4	2	26026	25768	1%
Heating 2	4	3	1	25678	25224	1.78%

Table 5.2: Comparison between PID controllers and distributed MPC

be improved from that of PID controllers. Especially in transient phase, distributed MPC reduces the overshoot effects which is produced with PID controllers. More precisely, as shown in Table 5.2, although only slight improvement on tracking temperature is obtained, energy consumption can be saved from 1% to 5% according to zones.

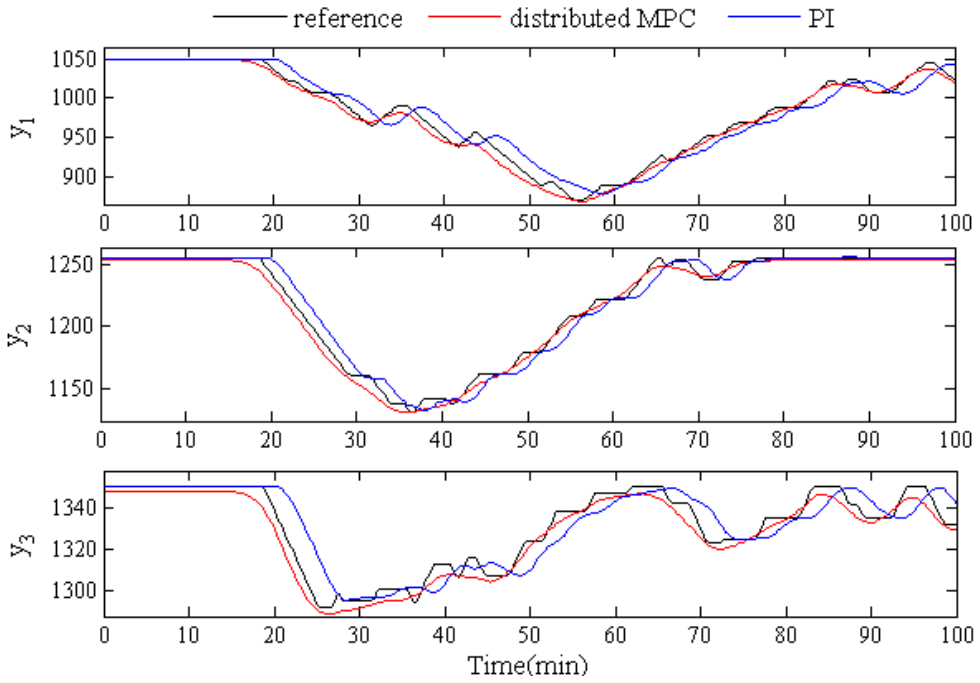

Figure 5.23: Zone temperatures

5.5 Industrial results

In the current situation, ArcelorMittal Florange steel plant has two walking-beam reheating furnace operating in parallel. The proposed distributed MPC algorithm has been implemented in one furnace (furnace *A*) and works in full-time. Another furnace (furnace *B*) is still utilizing PID controllers for zone temperature control. Two furnaces have identical control system technologies and reheat at the same time similar products in the same industrial constraints, therefore their thermal behaviors are quite similar. The behaviors of PID in furnace *B* can be compared to behaviors of distributed MPC of furnace *A* using data collected from these furnaces.

In Figure 5.24, the temperature of zone heating 1 of furnace A controlled by distributed MPC are compared to temperature of the equivalent zone of furnace B controlled by PID controllers. The blue curves are temperature set-points. The red curves represent the measured temperatures. We see that temperature set-points of two zones given by level 2 of two furnaces have similar dynamic. In particular,at time 11.30.00 to time 12.30.00, both furnace undergo a transient phase when temperature set-points change largely. In this transient condition, distributed MPC provides more stable temperature response while PID controllers show important temperature fluctuations. In general, distributed MPC showed less temperature overshoot than PID controllers, but also failed to attain the temperature set-points as can be seen at time 10.30.00 or 12.30.00. If we look closer in a shorter period of time as shown in Figure 5.25, distributed MPC can really reduced the overshoot effects of zone temperature which are produced with PID controllers. Now, an important question should be asked is: does distributed MPC change the accuracy of slab final temperatures?

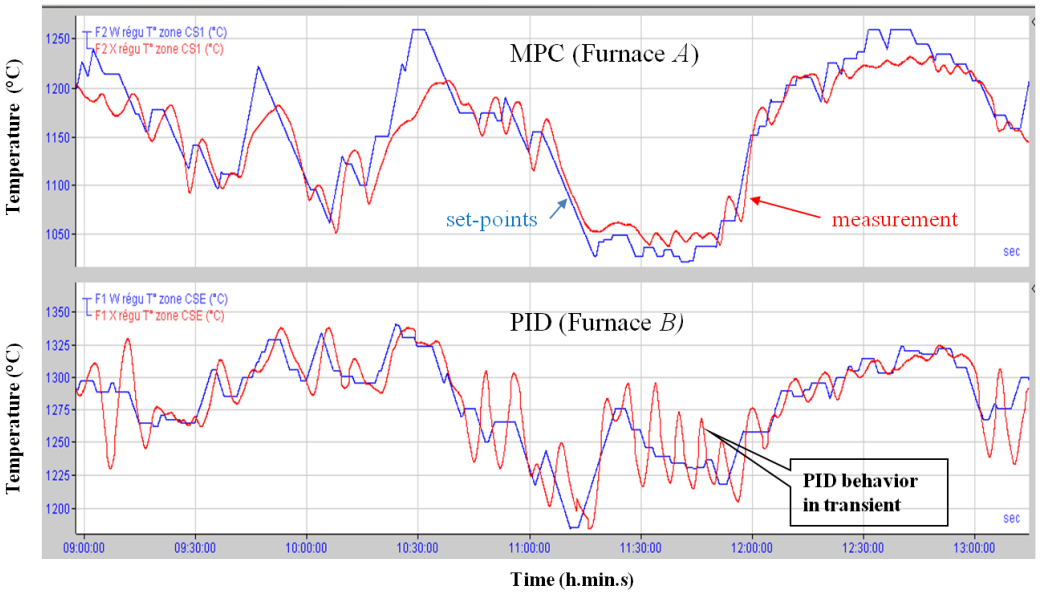

Figure 5.24: Transient behavior of zone temperature controlled by PID controllers and distributed MPC

Zone temperature tracking determines the success of final slab temperatures. As the primary objective of replacing PID controllers by distributed MPC is to reduce the energy consumption while maintaining the precision of zone temperatures, the influence of distributed MPC on slab discharged temperature is expected to be not significant. In Figure 5.26, the error of slab discharged temperatures is shown in form of histogram. The errors of final slab temperatures are normalized by a nominal value ΔT_0. We see that there are slight differences between distributed MPC and PID controllers. These insignificant differences confirm that distributed MPC does not change accuracy of final slab temperature compared to that given by PID controllers.

Zone temperatures are fed back to level 2 of the furnace control system where temperature set-point of furnace zone are calculated. Therefore, any fluctuation of zone temperature can cause disturbances in the calculation of temperature set-points which are shown in Figure 5.27. In furnace operation, bad tracking of zone temperature will causes invalidation for automatic calculation of temperature set-points, in this case manual intervention of human operator based on experiences will replace automatic calculation. Evaluation on how frequently the

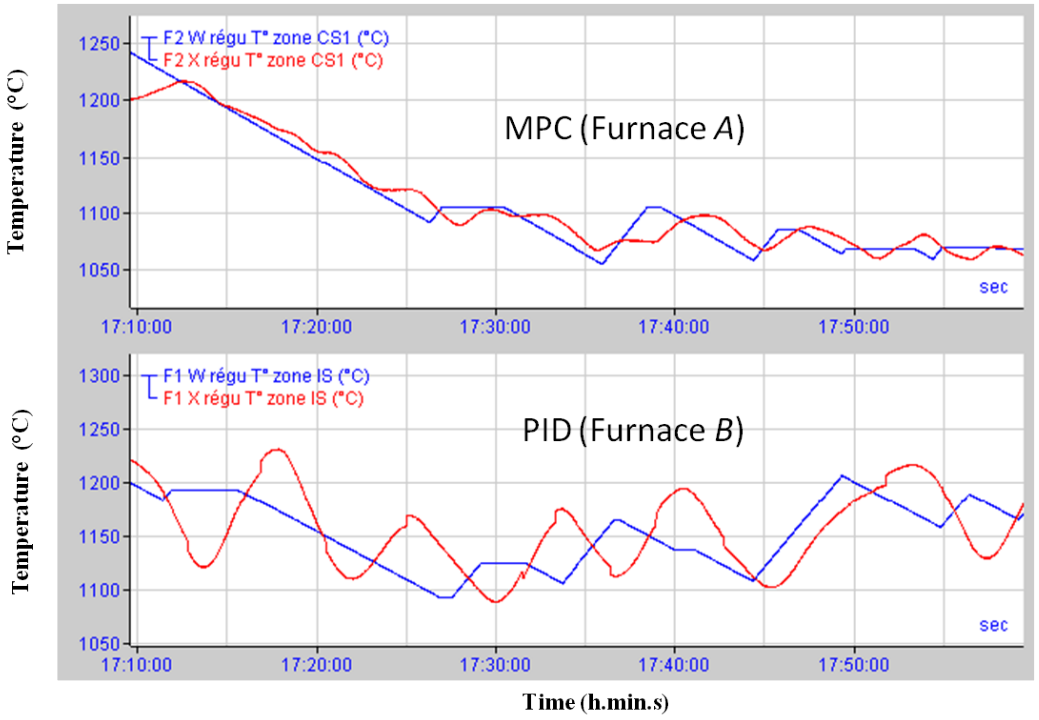

Figure 5.25: Overshoot effect of temperature controlled by distributed MPC and PID controllers

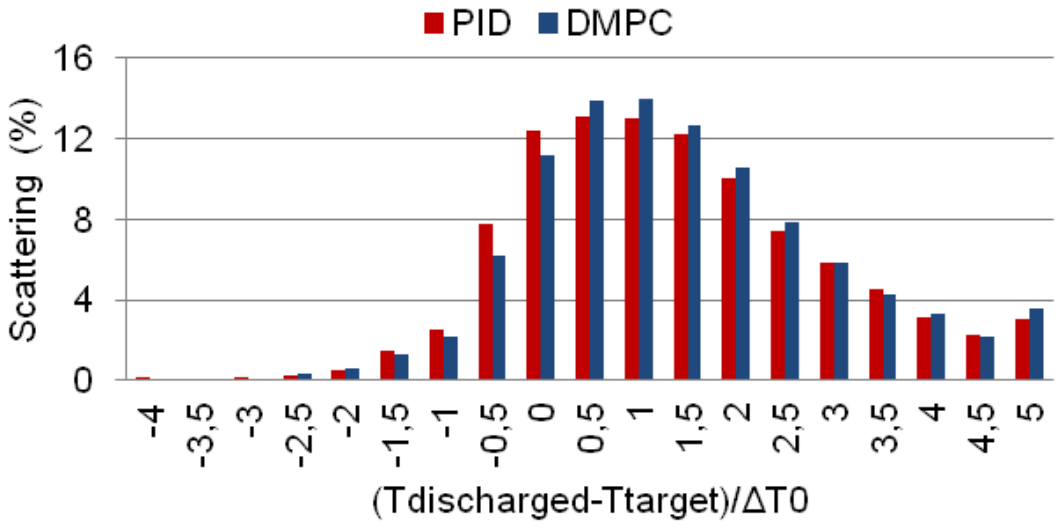

Figure 5.26: Histogram of final slab temperature errors

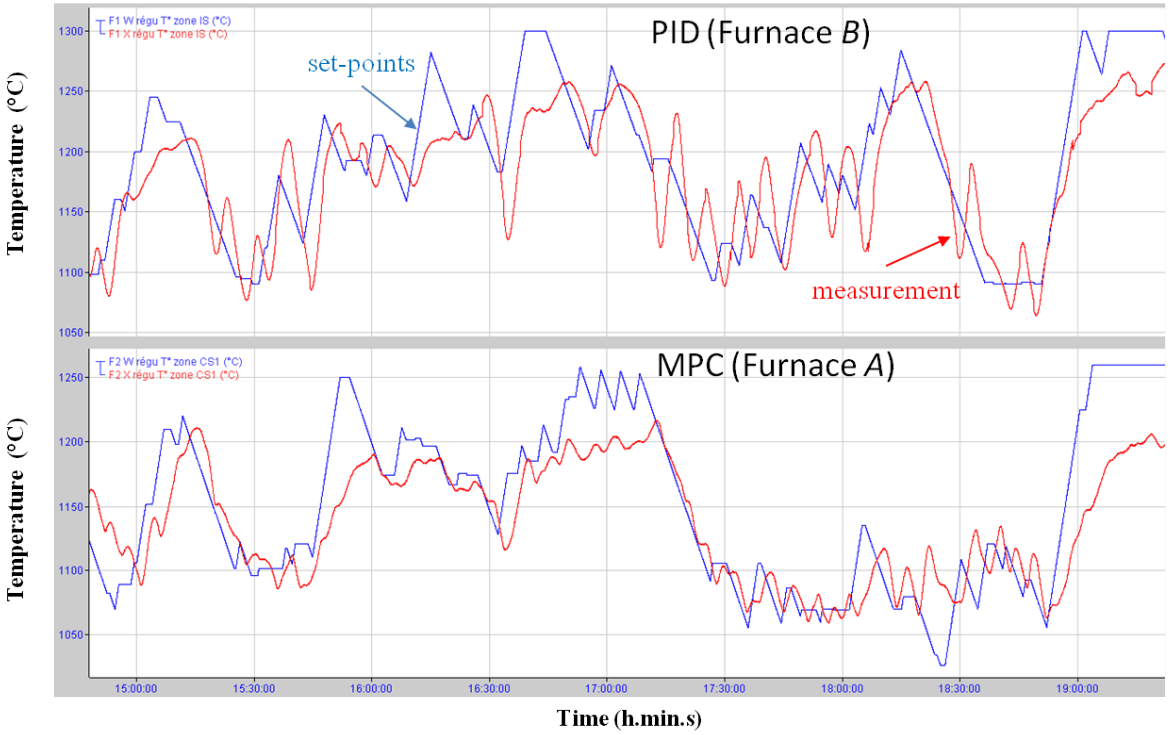

Figure 5.27: Variations of temperature set-points calculated by level 2

automatic mode is used can indicate the efficiency of temperature tracking at level 1 of the control system. In Figure 5.28, we see that with distributed MPC automatic usage of furnace control is more frequent than that with PID controllers. Especially in soaking zone where slab temperature profiles are homogenized, the automatic calculation of temperature set-points is significantly more frequent.

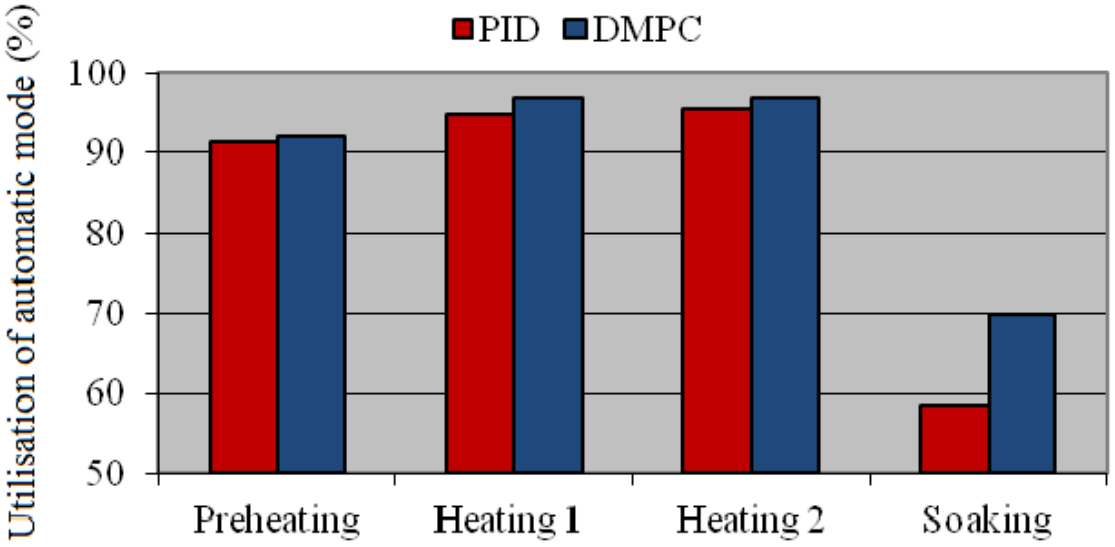

Figure 5.28: Utilization of automatic mode

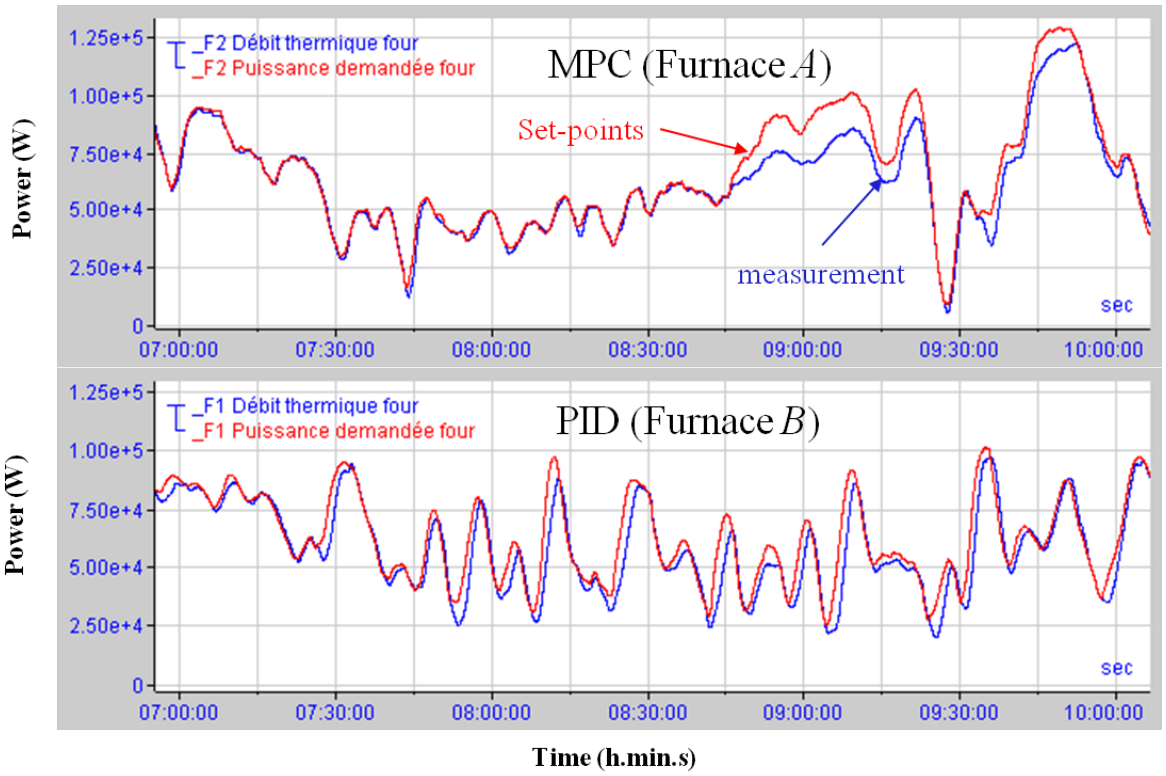

Figure 5.29: Power demands of controllers

Energy reduction is the main objective of replacing PID controllers by distributed MPC. As shown in Figure 5.29, distributed MPC gives smoother control input than that provided with PID controllers. This is coherent with the fact that zone temperatures controlled by distributed MPC are less fluctuated than that controlled by PID cotrollers. The data on specific energy consumption of the furnace A in two different periods is given in Figures 5.30 and 5.31. The first period is the first semester 2013 when furnace A was still operating with PID controllers. And the second period is the first semester 2014 when the furnace was operating with distributed MPC. In Figure 5.30, we have data of the daily average specific energy consumption of the furnace in function of daily furnace production. It can be noticed that the more the furnace produces in a day, the less energy it consumes to heat one ton of slab. And, the distributed MPC has reduced furnace consumption compared to PID controllers since red dots are concentrated at a lower level than the blue dots. The results can be seen more clearly in the histogram of daily specific consumption shown in Figure 5.31. The general energy reduction of replacing PID controllers with distributed MPC is 2.5% which correspond to the energy consumption 2500 houses.

In addition to reduction of energy consumption, the oxygen ratio inside the furnace is also reduced and less fluctuated by distributed MPC in comparison to result of PID controllers as shown in Figure 5.32. Since lower and more stable oxygen ratio will reduce the decaburization process, the quality of slabs is therefore improved by distributed MPC.

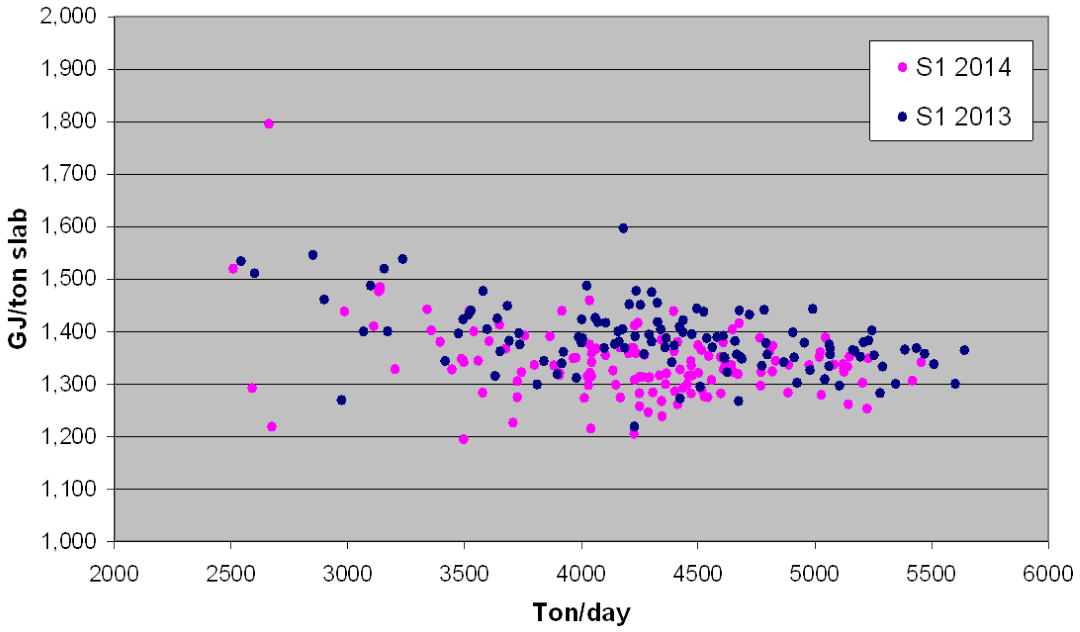

Figure 5.30: Specific energy consumption

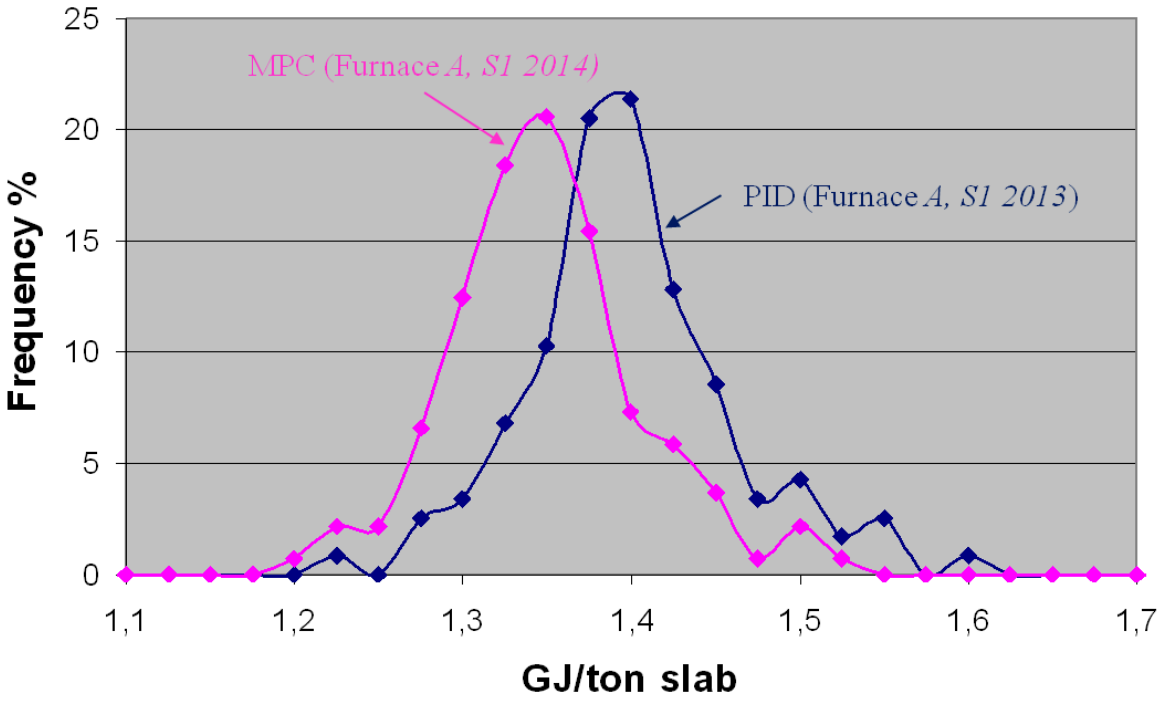

Figure 5.31: histogram of specific energy consumption

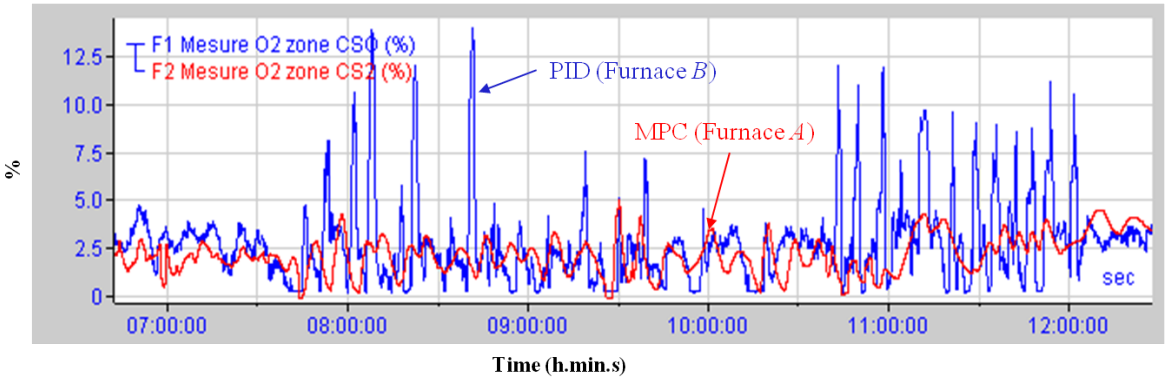

Figure 5.32: Oxygen ratio inside the furnace

5.6 Conclusions

To conclude, in this chapter the interaction effects between furnace zones are examined. Coupling between zones can be modeled as input coupling through zone input power or output coupling through zone temperature. The output coupling mode is validated and adopted for the control of furnace. According to the adopted zone model, two distributed MPC algorithms are developed for system, where subsystems are coupled by outputs. The first algorithm is a non-cooperative strategy in which local MPC controllers do not account for control performance of other controllers. The second algorithm is constructed so that local MPC controllers account for performance of neighboring controllers. This is due to the fact that a furnace zone interacts mainly with its neighbors. Numerical simulation is carried out for an illustrative case using centralized MPC and distributed MPC strategies, which allows verifying the optimality of proposed non-cooperative distributed MPC algorithm. Simulation for a realistic scenario of furnace operation shows improvement of proposed control strategy in comparison with PID decentralized controllers. The proposed non-cooperative distributed MPC has been implemented in industrial conditions for the level 1 of the considered furnace and provides significant reduction of energy consumption for the furnace. In next chapter, MPC strategy is examined for level 2 of furnace control system to optimize zone temperature set-points.

Chapter 6

Slab temperature control

The previous chapter has dealt with zone temperature control, so-called level 1 of Figure 6.1. In this chapter, the tracking problem of zone temperature set-points has been resolved with distributed MPC strategy. Zone temperature set-points are in fact calculated by the level 2 of the control system (see Figure 6.1). This latter level is also mentioned as slab temperature control because it determines heating process of slabs through out the furnace. Therefore, optimization of temperature set-points calculation is crucial to obtain efficient furnace operation. In this chapter dealing with the elaboration of these zone temperature set-points, the problem statement is firstly given describing qualitative and quantitative control objectives of level 2. Secondly, model predictive control strategy is clarified for slab temperature control. Next, mathematical optimization methods are presented, among which Nelder-Mead simplex algorithm is further utilized as the optimization method of the MPC strategy. A numerical simulation is carried out for an example scenario of the real furnace system. The obtained results showed promising improvements in term of energy consumption and temperature accuracy.

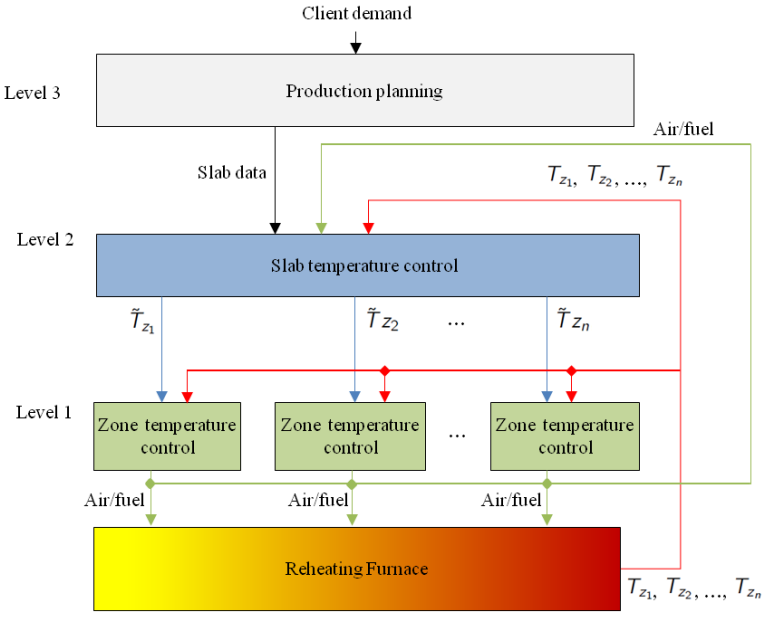

Figure 6.1: Feedback in hierarchical structure

6.1 Problem statement

The problem of slab temperature control in this chapter is limited to a fixed scheduling data which comes from level 3. That is, loading and discharging events of a slab are predefined. As a result, residence time of a slab, component of load inside furnace at a given time is known in advance.

The closed-loop of furnace control system is clarified in Figure 6.2. The desired final temperature of slab j is denoted $\tilde{T}_{j,end}$; it is provided by the level 3 (see Figure 6.1). The final slab temperature is estimated as $T_{j,end}$ via a dynamic simulator of furnace, this simulator is designed based on the non-steady-state model presented in chapter 3 and utilizes the measurements of furnace zone power P_i and zone temperature $T_{z,i}$, with $i = 1 \div M$. The distributed MPC assures the tracking of zone temperature set-points $\tilde{T}_{z,i}$ by determining the optimal zone power $\tilde{P}_i$ which minimize zone temperature error and energy consumption criterion. In general, the control of level 1 is considered transparent to level 2 meaning the tracking of zone temperature set-points has high accuracy and fast response in comparison to calculation frequency of level 2. MPC strategy in level 2 computes the optimal zone temperature set-points which minimize error of final slab temperature and energy consumption. This optimization of temperature set-points is based on a simplified version of the non-steady-state model. In the following, different control objectives and constraints are specified.

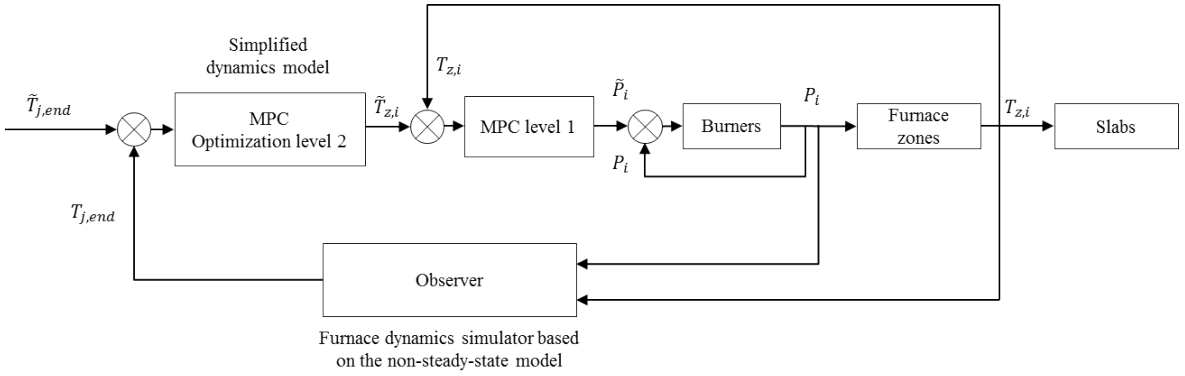

Figure 6.2: Cascaded closed-loop of the furnace control system with MPC strategy

6.1.1 Qualitative control objectives and constraints

A temperature control system of a slab reheating furnace should satisfy the following control *objectives*:

- Minimum deviation between the desired and the realized final slab temperature profiles.
- Minimum specific energy consumption which is defined by the ratio $\frac{\text{Energy supplied by fuel combustion}}{\text{Mass of heated slabs}}$
- Minimum loss of material through scale formation
- Minimum decarburization which has an influence on the material quality

In general, these control objectives are motivated by economical reasons, in particular operating costs and increasing demands in terms of quality and diversity of products. Obviously, the main objective of slab reheating furnace is that sufficient heat is absorbed and distributed homogeneously into slabs. This objective is defined by a desired value which is referred to as *desired final slab temperature profile*. Typically, it is imposed by downstream process which is hot rolling mills. The second objective is minimum specific energy consumption which is measure by amount of energy consumed to sufficiently heat one ton of steel slab. This implies minimizing fuel consumption and at the same time minimizing material losses. Scale formation is the process of oxidation of steel which creates iron oxides Fe_xO_y on slab surfaces and therefore leads to loss of material. The decarburization is the process of reduction of carbon content in slabs, this can undesirably modify the quality of steel. As a result, both scale formation and decarburization are phenomenons to be kept at a minimum. Unfortunately, many mathematical models of reheating furnace, including the one suggested in Chapter 3, are not sufficiently detailed to provide quantitative estimation of the two later objectives. However, a good approach to the first two objectives usually leads to good results of the two latter ones.

The reheating furnace is subject to various constraints, specially temperature constraints due to the extreme environment of furnace interior. A slab temperature control scheme should account the following essential constraints:

- Bounds on the temperature of the furnace interior and the wall in order to protect them from heat damage, this is usually expressed as bounds on zone temperature set-points.
- Limitations of the control inputs: fuel and air feeds.
- Constraint on the slab temperature trajectories.

Temperature control is a complicated task due to following reasons:

- The temperatures inside slabs can not be measured, they are only estimated by the furnace model.
- A slab reheating furnace is a nonlinear MIMO distributed parameter system and exhibits a switched dynamic behavior. That is because the furnace steel load varies over time due to changes of number of slabs inside furnace, and characteristic of these slabs. Furnace components span a wide range of dynamics, that is for example: response times range from fractions of a second for fuel/air valves up to several hours associated with the thermal inertia of furnace walls.
- Slab movement is generally discontinuous and sometimes unforeseeable due to scheduled and unscheduled halts or delays of upstream and downstream process in the work-flow.

Furthermore, the references [Soc, Tot04, HBK09, RSCG86, Sta04] report some recent trends in furnace operation which should be considered in furnace temperature control design. These trends are listed as follows:

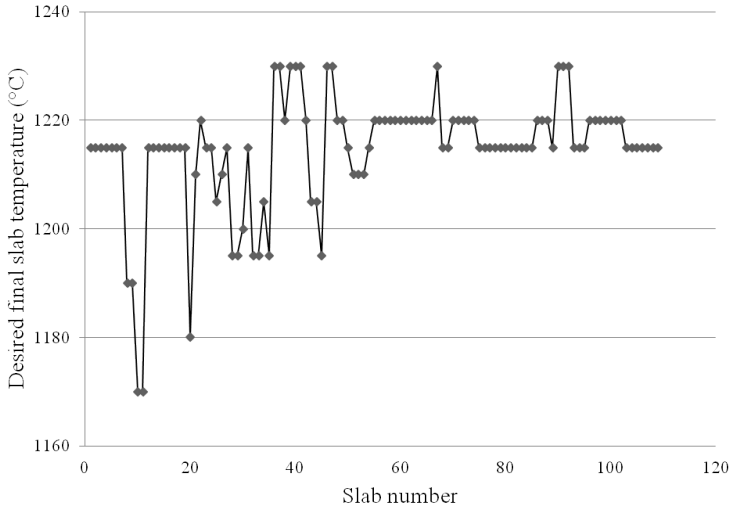

Figure 6.3: Variation of desired final slab temperatures in an operation of the considered furnace

- An increasing diversification of products with more exigent requirements entails large variations of the desired final temperatures as shown in Figure 6.3, the slab geometry as illustrated in Figure 6.4, the material properties, the residence time, the monetary values, etc. Indeed, there are more and more operating scenarios where each slab inside the furnace is the only one of its kind.
- For higher mechanical properties of product, the requirement concerning temperature gradient inside slabs may be tighter
- Utilization of available furnace capacity should be maximized to avoid situation where the reheating furnace may constitute a bottleneck of the production process. However, this problem is out of the scope of this chapter and will be examined in the next one.
- Non-standard operating phases like furnace start-up or shut-down require particular control strategies. An automatic furnace temperature control system should cope with such situations and keep manual interventions at a minimum level.
- Nowadays, furnace operators have to confront with official emission limits. The minimization of exhaust emissions directly links to maximum efficiency and minimum fuel consumption. Therefore, furnace control design striving for the latter challenges is not only an economical but also an environmental solution.

These trends cause an increased deviation from an ideal *steady-state* operation of the furnace. Here, the steady-state operation is referred to furnace conditions where all processed slabs have the similar properties, dimensions, desired final temperature profiles, residence time, etc., thus the production rate of the furnace is constant, and loading and discharging events occur at regular time intervals. The greater deviation of an operating scenario from the ideal design point of a furnace and the greater diversification of slab properties, the more difficult is the slab temperature control task. In fact, *non-steady-state* operation of the furnace has become the *normal* rather than

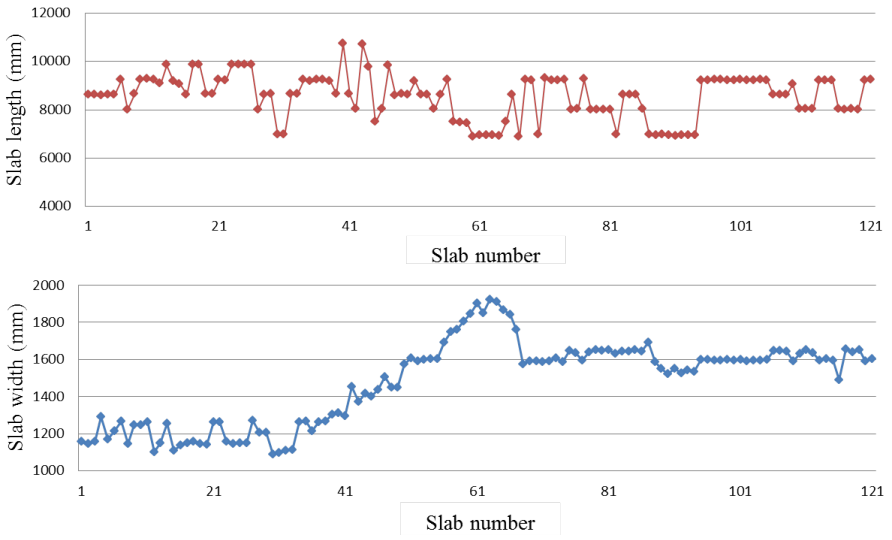

Figure 6.4: Variation of slab width and length

the exceptional operating scenario. In practice, if the difference between the desired temperature of neighboring slabs is considerable, the slab with highest desired final temperature will have the priority.

Finally, it shall be noted that reheating process is sensitive with the thickness of slab because thicker slabs have slower dynamics behavior compared to that of thin slabs. The slab volume grows linearly with the slab thickness, while the heat exchange area, depends only marginally on the thickness because slab width and length are much bigger than slab thickness. As a result, slab thickness has a direct impact on the necessary heating time for obtaining the desired final temperature profile. A furnace is usually designed for specific slabs with a certain value of thickness to assure that residence times do not deviate too much. As an example, the considered furnace heats slab with thickness of 235 mm.

6.1.2 Quantitative control objectives and constraints

The qualitative control objectives and constraints are now restated in a mathematical form. As temperatures are relevant process variables, it seems reasonable to express control constraints in terms of temperature or their time derivatives, even if these constraints correspond to other physical quantities. There are constraints on zone temperatures and slab temperatures. To facilitate the formulation, the constraints will be presented in the continuous-time domain. The transformation to the discrete-time domain is straightforward.

6.1.2.1 Constraint of zone temperature set-points

The fuel flow rates to the burners are subject to certain limits. Some burner types cannot be automatically switched off, which leads to a lower bound on the fuel flow rate that is considerably above zero. Moreover, furnace operation needs to maintain the zone temperatures at some minimum values. Generally, these bounds on zone

temperatures vary over time. For these reasons, temperature set-points of controllable zones and their rate of change are limited as

$$\tilde{T}_{z,i,min}(t) \leq \tilde{T}_{z,i}(t) \leq \tilde{T}_{z,i,max}(t) \tag{6.1a}$$

$$\dot{\tilde{T}}_{z,i,min}(t) \leq \dot{\tilde{T}}_{z,i}(t) \leq \dot{\tilde{T}}_{z,i,max}(t), \tag{6.1b}$$

where $i = 1 \div M$ is the zone number and M is the number of controllable zones. For the considered furnace, $M = 4$ which corresponds to four controllable zones: preheating, heating 1, heating 2, soaking. The constraints $\tilde{T}_{z,i,min}(t)$, $\tilde{T}_{z,i,max}(t)$, $\dot{\tilde{T}}_{z,i,min}(t)$, and $\dot{\tilde{T}}_{z,i,max}(t)$ are required for preventing the furnace zone from thermal or thermomechanical damage.

The inequalities 6.1 imply that any satisfactory temperature set-point trajectory $\tilde{T}_{z,i}(t)$ can be followed by the real furnace system. This is the substantial condition because $\tilde{T}_{z,i}(t)$ $(i = 1, ..., M)$ are used as control inputs. In a vector form, zone temperature set-point vector is denoted by $\tilde{T}_z(t) = [\tilde{T}_{z,1}(t), ..., \tilde{T}_{z,M}(t)]'$, and the corresponding lower and upper bounds are denoted as $\tilde{T}_{z,min}(t) = [\tilde{T}_{z,1,min}(t), ..., \tilde{T}_{z,M,min}(t)]'$ and $\tilde{T}_{z,max}(t) = [\tilde{T}_{z,1,max}(t), ..., \tilde{T}_{z,M,max}(t)]'$, respectively. Then, the constraints on zone temperature set-points can be rewritten as:

$$\tilde{T}_{z,min}(t) \leq \tilde{T}_z(t) \leq \tilde{T}_{z,max}(t) \tag{6.2a}$$

$$\dot{\tilde{T}}_{z,min}(t) \leq \dot{\tilde{T}}_z(t) \leq \dot{\tilde{T}}_{z,max}(t) \tag{6.2b}$$

Generally, the bounds of inequalities 6.1: $\tilde{T}_{z,i,min}(t)$, $\tilde{T}_{z,i,max}(t)$, $\dot{\tilde{T}}_{z,i,max}(t)$, and $\dot{\tilde{T}}_{z,i,min}(t)$ are given with nominal values as recapitulated in Table 6.1. However, if other constraints are active, e.g., heating capacity of burners, while zone temperatures have not yet reached their nominal bounds, the nominal values should be updated. Therefore, in practice, zone temperature constraints may be variable with time.

Variables	Nominal values
$\tilde{T}_{z,min}(t)$	$[830, 1085, 1200, 1210]$ (°C)
$\tilde{T}_{z,max}(t)$	$[1100, 1250, 1355, 1300]$ (°C)
$\dot{\tilde{T}}_{z,min}(t)$	-4.5 (°C/min)
$\dot{\tilde{T}}_{z,max}(t)$	9 (°C/min)

Table 6.1: Constraints on zone temperature set-points

Other modifications of $\tilde{T}_{z,i,min}(t)$, $\tilde{T}_{z,i,max}(t)$, $\dot{\tilde{T}}_{z,i,max}(t)$, and $\dot{\tilde{T}}_{z,i,min}(t)$ may come from furnace operators. In fact, a time-varying design of zone temperature constraints facilitates occasional interventions of furnace operator in some particular situations. For example, the furnace operator may want to temporarily keep the zone temperature at some constant values or within some narrow range. This happens usually for manual shut-down, start-up, or production halts of the furnace. In these conditions, furnace is deviated from nominal operating point,

which affects the reheating quality of the slabs. The next section presents how the reheating quality of a slab is defined.

6.1.2.2 Constraints of slab temperatures

We will consider the heating constraints of a single slab j. The main control objective is that the slab has the final temperature profile close to the desired one denoted $\tilde{T}_{j,end}(y_j)\ \forall\ y_j \in [-D_j/2, D_j/2]$ (D_j is thickness of slab j) when it is withdrawn from the furnace at the time $t_{j,1}$. Let $t_{j,0}$ be the time at which the slab is loaded into the furnace, $T_j(y_j,t)\ \forall\ y_j \in [-D_j/2, D_j/2]$ is the temperature profile of slab at time $t \in [t_{j,0}, t_{j,1}]$. As detailed in chapter 3 where the heat conduction problem inside the slabs are presented, the slab temperature profile is determined at limited points inside the slab. The more estimated points the more slab temperature profile is accurate with the price of more required computational resource. For control purpose, a considerably small number of points can be sufficient. For instance, temperatures of 10 points inside a slab are estimated to constitute the temperature profile of the slab. Then, the control would achieve

$$T_j(y_j, t_{j,1}) = \tilde{T}_{j,end}(y_j),\ \forall\ y_j \in [-D_j/2, D_j/2]. \tag{6.3}$$

Slab temperatures are subject to *hard* and *soft* constraints. Violation of soft constraints is only allowed if oth-

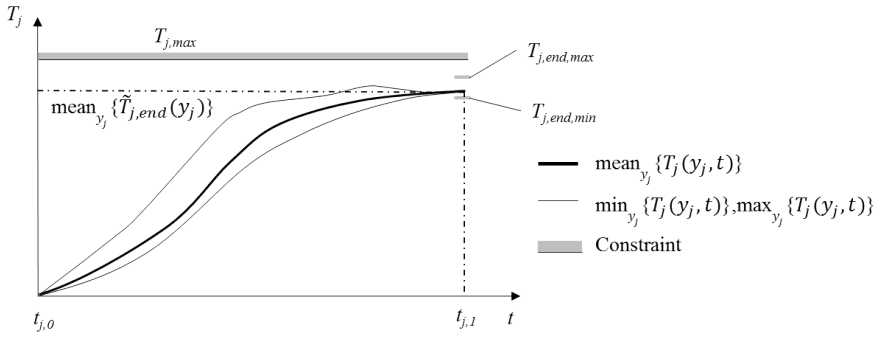

Figure 6.5: Constraints on slab temperature trajectory

erwise a solution satisfying all hard constraints is infeasible. Nevertheless, because the distinction between hard and soft constraints is a qualitative one, hard constraints may also be broken as a last alternative. The restrictions are usually destined to satisfy metallurgical requirements, and requirements of downstream processes, e.g., hot rolling mills. Constraints may be imposed on absolute temperature or relative temperature. Figure 6.5 shows a typical slab temperature trajectory inside the furnace. Slab temperature profile is hard-constrained through the residence period of the slab with the upper limit $T_{j,max}$ preventing waste of energy as well as damage and loss of material, i.e.,

$$T_j(y_j,t) \le T_{j,max} \qquad \forall\, t \in [t_{j,0}, t_{j,1}],\ \forall\, y_j \in [-D_j/2, D_j/2]. \tag{6.4}$$

In practice, although this constraint is categorized as hard constraint; it is generally easier to satisfy than other constraints which are presented in the following.

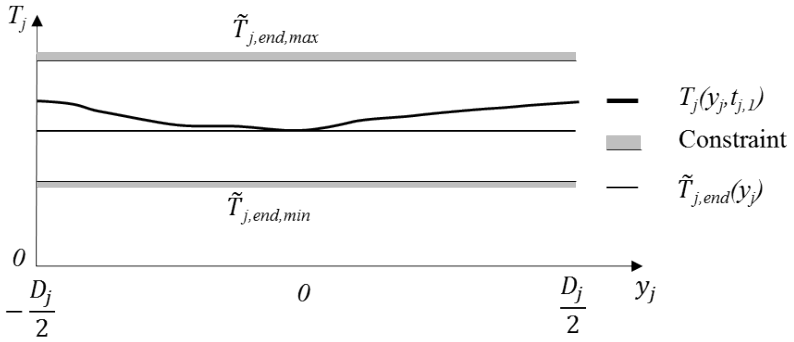

Figure 6.6: Constraints on final slab temperature profile

The final temperature profile of the slab is constrained with soft upper (lower) limit $T_{j,end,max}$ ($T_{j,end,min}$) as shown in Figure 6.6. These bounds allow a certain level of tolerance of final slab temperature profile error which comes from the control error. The formal expressions are

$$T_{j,end,min} \leq T_j(y_j, t_{j,1}) \leq T_{j,end,max} \qquad \forall\, t \in [t_{j,0}, t_{j,1}],\ \forall\, y_j \in [-D_j/2, D_j/2]. \tag{6.5}$$

In spite of being a soft constraint, this constraint is more difficult to be satisfied than the hard constraint on slab temperature provided in Equation 6.4 because it demands more precision of the furnace control. For downstream process, e.g., hot rolling mills, a homogeneous temperature distribution is usually preferred. However, in practice, a certain level of temperature inhomogeneity can be allowed. Let $\Delta T_j(t)$ denote the temperature gradient of the slab which is defined as

$$\Delta T_j(t) = \max_{y_j}\{T_j(y_j, t)\} - \min_{y_j}\{T_j(y_j, t)\},\ y_j \in [-D_j/2, D_j/2],\ \forall\, t \in [t_{j,0}, t_{j,1}]. \tag{6.6}$$

Large temperature differences within a slab can cause undesirable global deformation like bending of slab. In addition, excessive inhomogeneities of final temperature profile may lead to unwanted incident during the transport of hot slab, and rejection at the rolling process because average temperature of slab at entrance of rolling mill may be not sufficiently high. Therefore, the inhomogeneity $\Delta T_j(t)$ of the slab temperature is subject to an upper bound hard constraint and a tighter final bound but considered as a soft constraint as shown in Figure 6.7. The formal expressions are written as:

$$\Delta T_j(t) \leq \Delta T_{j,max} \quad \forall\, t \in [t_{j,0}, t_{j,1}], \tag{6.7a}$$

$$\Delta T_j(t_{j,1}) \leq \Delta T_{j,end,max}, \tag{6.7b}$$

where $\Delta T_{j,max}$ is a constant, and generally $\Delta T_{j,end,max} \leq T_{j,end,max} - T_{j,end,min}$. In addition to forgoing constraints and objectives, to reduce energy consumption, one has to take into account the following objectives:

- The slab temperature trajectories should be kept as low as possible, which may also reduce material losses through scale formation and allow lower zone temperatures

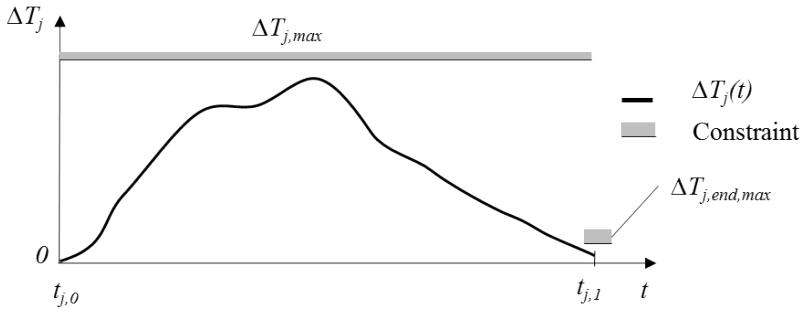

Figure 6.7: Constraints on inhomogeneity of slab temperature

- The zone temperature is kept at minimum sufficient level, which reduces directly fuel supply into the furnace

In Table 6.2, the constraints of a slab temperature profile are recapitulated. In the next section, the presented objectives and constraints will be accounted for a set of slabs in the design of MPC strategy for slab temperature control.

Variables	Description	Nominal values
$T_{j,max}$	upper bound on slab temperature	1350 °C
$\tilde{T}_{j,end,max} - \tilde{T}_{j,end}$	upper margin of final slab temperature	20 °C
$\tilde{T}_{j,end} - \tilde{T}_{j,end,min}$	lower margin of final slab temperature	20 °C
$\Delta T_{j,max}$	maximum inhomogeneity of slab temperature profile	350 °C
$\Delta T_{j,end,max}$	maximum inhomogeneity of final slab temperature profile	100 °C

Table 6.2: Constraints of slab temperature profile

6.2 MPC strategy applied to slab temperature control

This section describes the MPC approach for the control at level 2 of furnace control system. The discussion will begin with how the system can be seen from level 2 point of view by defining its inputs and outputs. Next, the formulation of the problem is given for the design of MPC method. And finally, some important issues on computational requirement and stability are discussed.

6.2.1 Principle

As previously mentioned in chapter 5, MPC is a model-based feedback control concept. For level 2 of furnace control system, control input denoted $u(t)$ are defined by the zone temperatures as

$$u(t) = [u_1(t), ..., u_M(t)]' \tag{6.8a}$$

$$= [\tilde{T}_{z,1}(t), ..., \tilde{T}_{z,M}(t)]'. \tag{6.8b}$$

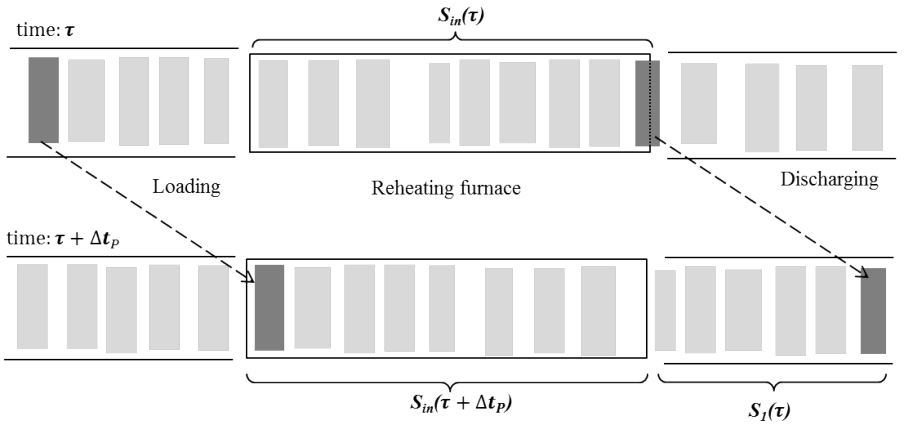

Figure 6.8: Distinction between $S_{in}(\tau)$ and $S_1(\tau)$ for furnace operation during prediction horizon $\tau, \tau + \Delta t_P$

The control problem is clearly the tracking of the desired final slab temperature profile. Preferably, the desired final slab temperature profile would be homogeneous. However, in real process, tolerance of a certain level of inhomogeneity is adopted. As a result, the control problem can be defined by the tracking of an average desired final slab temperatures and the minimization of the slab temperature gradients. The outputs to be controlled are therefore average final slab temperatures, and slab temperature gradients, respectively defined as follows

$$T_s(t) = \begin{cases} mean_{y_j}\{T_j(y_j, t_{j,1})\}, \text{ if } \exists\ j \in S, t = t_{j,1} \\ 0, \text{ otherwise} \end{cases} \tag{6.9a}$$

$$g(t) = \begin{cases} \Delta T_j(t_{j,1}), \text{ if } \exists\ j \in S, t = t_{j,1} \\ 0, \text{ otherwise} \end{cases} \tag{6.9b}$$

where S denotes the set of slabs to be processed and the subscript s stands for *slab*. It can be noticed that outputs of the system can also be considered as discontinuous variables where discontinuous points correspond with discharging times of slabs. Consequently, set-point for output $T_s(t)$, and its bounds can be defined as

$$\tilde{T}_s(t) = \begin{cases} \tilde{T}_{j,end}, \text{ if } \exists\ j \in S, t = t_{j,1} \\ 0, \text{ otherwise} \end{cases} \tag{6.10}$$

$$\tilde{T}_{s,min}(t) = \begin{cases} \tilde{T}_{j,end,min}, \text{ if } \exists\ j \in S, t = t_{j,1} \\ 0, \text{ otherwise} \end{cases} \tag{6.11}$$

$$\tilde{T}_{s,max}(t) = \begin{cases} \tilde{T}_{j,end,max}, \text{ if } \exists\ j \in S, t = t_{j,1} \\ 0, \text{ otherwise} \end{cases} \tag{6.12}$$

And the constraint on maximum allowed gradient temperature can be defined as

$$g_{max}(t) = \begin{cases} \Delta T_{j,end,max}, \text{ if } \exists\ j \in S, t = t_{j,1} \\ 0, \text{ otherwise} \end{cases} \tag{6.13}$$

Since slabs are loaded and discharged gradually, the furnace system has actually a switched dynamic. Clearly, the variables of interest of the system at each time t are temperature profiles of all slabs that are currently inside the furnace. However, we see that slab temperature constraints can be expressed in terms of the temperature gradient $\Delta T_j(t)$ and maximum temperature inside slabs: $\max_{y_j} T_j(y_j, t)$. Because most of the constraints concerns the upper bounds of slab temperature profile and the temperature gradient, we chose the vector $x(t) = [x_1'(t), x_2'(t)]'$ as the variable of interest, with $x_1(t)$ and $x_2(t)$ defined as follows:

$$x_1(t) = [\max_{y_{j+1}} T_{j+1}(y_{j+1}, t), \max_{y_{j+2}} T_{j+2}(y_{j+2}, t), ..., \max_{y_{j+N_{in}}} T_{j_{in}}(y_{j+N_{in}}, t)]', \tag{6.14a}$$

$$x_2(t) = [\Delta T_{j+1}(t), \Delta T_{j+2}(t), ..., \Delta T_{j+N_{in}}(t)]', \quad y_j \in [-D_j/2, D_j/2], \tag{6.14b}$$

with $T_{j+1}(y_{j+1}, t)$ is temperature profile of slab situating at entrance of the furnace, and $T_{j+N_{in}}(y_{j+N_{in}}, t)$ is the temperature profile of slab being at the exit door of the furnace. N_{in} is the number of slab residing inside the furnace at time instant t and can change over time. $S_{in}(t)$ defined by $[j+1, j+2, ..., j+N_{in}]$ denotes the index set of slabs that are inside the furnace at time t. Model predictive control principles applied to slab temperature control can be now explained by the following steps.

- At the present time τ the variable $x(\tau)$ is estimated by the non-steady state model. Based on this model the system outputs $T_s(t), g(t)$ are predicted over the prediction horizon $[\tau, \tau + \Delta t_P]$ with Δt_P the prediction horizon. The basic data necessary for the forecast is composed of: future system control input $u(t)\ \forall\, t \in [\tau, \tau + \Delta t_C]$ (Δt_C is the control horizon) and future knowledge about furnace configuration, disturbances, etc.

- The control input $u(t)\ \forall\, t \in [\tau, \tau + \Delta t_C]$ is calculated so that the predicted system outputs $T_s(t), g(t)$ $\forall\, t \in [\tau, \tau + \Delta t_P]$ match the desired values as much as possible. This matching criterion is specified by an *objective function*. In general, the control input $u(t)$ is obtained as a solution of an open-loop optimal control problem defined in the interval $[\tau, \tau + \Delta t_P)$.

- The optimized control input $u(t)$ is applied to the real system over the interval $[\tau, \tau + \Delta\tau_s)$, with $\Delta\tau_s$ represents the time interval between consecutive optimization events.

- The procedure is repeated for the optimization at $\tau + \Delta\tau_s$ to redesign the optimal control input $u(t)\ \forall\, t \in [\tau + \Delta\tau_s, \tau + \Delta\tau_s + \Delta t_C]$.

The whole procedure is presented in Figure 6.9. In the following, formulation of the optimal control problem is clarified with mathematical equations accounting for furnace control objectives and constraints.

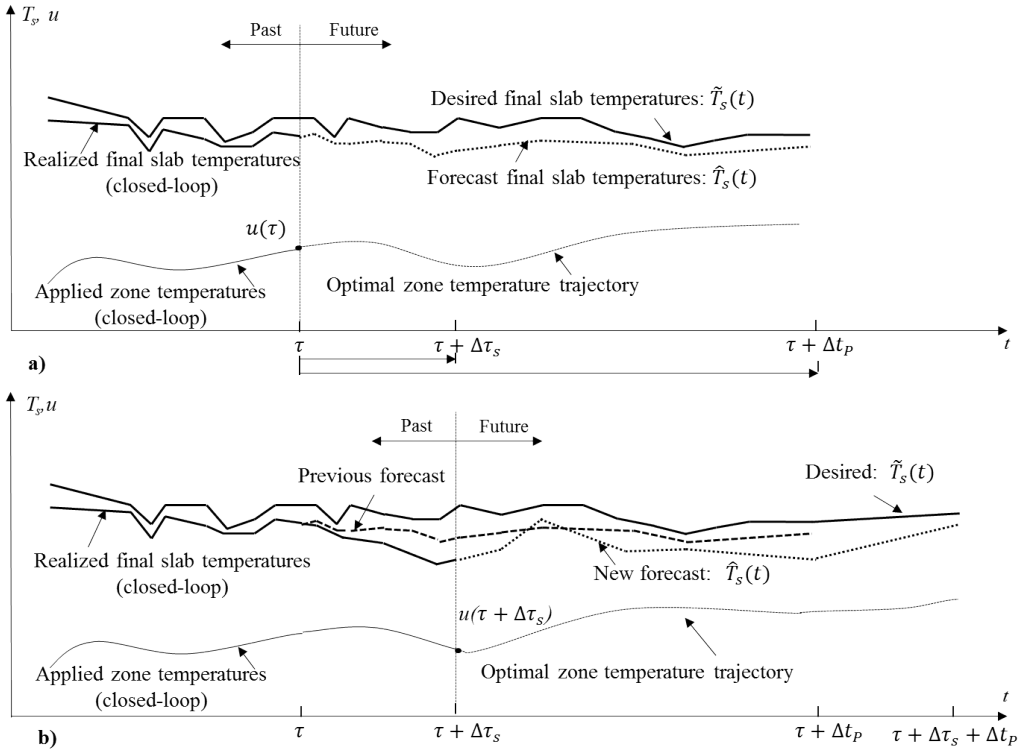

Figure 6.9: MPC applied for slab temperature control: a) at time τ, b) at time $\tau + \Delta\tau_s$

6.2.2 Continuous-time problem

6.2.2.1 Problem formulation

In order to understand more the furnace control task, we will first approach the problem in continuous-time domain. To formulate the optimization problem, some terminological definitions are provided for time instant τ:

- Control input sequence $U(\tau) = \{u(t), \forall t \in [\tau, \tau + \Delta t_P\}$. It shall be noted that $u(t) = u(\tau + \Delta t_C),\ \forall t \in [\tau + \Delta t_C, \tau + \Delta t_P]$.
- Performance indexes on error of final average slab temperature, on inhomogeneity of final slab temperature, on energy consumption of the furnace are denoted by $J_1(U(\tau))$, $J_2(U(\tau))$, $J_3(U(\tau))$, respectively. General control objectives of low overall furnace temperature and low slab temperature trajectories are accounted by the term $J_4(U(\tau))$ and $J_5(U(\tau))$, respectively. These performance indexes will be described in the next section.
- Figure 6.8 illustrates the sets $S_{in}(\tau)$: index set of slabs that are inside the furnace at time τ, and $S_1(\tau)$: the index set of slabs that are discharged during the prediction horizon period $[\tau, \tau + \Delta t_P]$.
- Upper bounds on slab temperature trajectories and temperature gradients are denoted $T_{max}(t)$ and $\Delta T_{max}(t)$ respectively. These variable are switched accordingly with loading and discharging events (changes of

$S_{in}(t)$).

$$T_{max}(t) = [T_{j+1,max}, T_{j+2,max}, ..., T_{j+N_{in},max}], \tag{6.15a}$$

$$\Delta T_{max}(t) = [\Delta T_{j+1,max}, \Delta T_{j+2,max}, ..., \Delta T_{j+N_{in},max}], \tag{6.15b}$$

with $j \in S_{in}(t) = [j_0, j_0 + 1, ..., j_{in}]$ and $t \in [\tau, \tau + \Delta t_P]$. The optimization problem is then defined as minimization of the overall objective function $J_o(U(\tau))$ which is the sum of weighted objectives concerning slab temperature, zone temperature, and energy consumption. This minimization problem is subjected to the constraints of zone temperature set-points and slab temperature profiles. The formal expression is given as:

$$\min_{U(\tau)} J_o(U(\tau)) = \alpha_1 J_1(U(\tau)) + \alpha_2 J_2(U(\tau)) + \alpha_3 J_3(U(\tau)) + \alpha_4 J_4(U(\tau)) + \alpha_5 J_5(U(\tau)) \tag{6.16a}$$

subject to

$$T_{z,i,min}(t) \leq u_i(t) \leq T_{z,i,max}(t), \ \forall i = 1 \div M, t \in [\tau, \tau + \Delta t_P] \tag{6.16b}$$

$$\dot{T}_{z,i,min}(t) \leq \dot{u}_i(t) \leq \dot{T}_{z,i,max}(t), \ \forall i = 1 \div M, t \in [\tau, \tau + \Delta t_P] \tag{6.16c}$$

$$T_{j,end,min} \leq T_s(t) \leq T_{j,end,max}, \text{ if } j \in S_1(\tau), t = t_{j,1}, t \in [\tau, \tau + \Delta t_P] \tag{6.16d}$$

$$g(t) \leq \Delta T_{j,end,max}, \text{ if } j \in S_1(\tau), t = t_{j,1}, t \in [\tau, \tau + \Delta t_P] \tag{6.16e}$$

$$x_1(t) \leq T_{max}(t), \ \forall t \in [\tau, \tau + \Delta t_P] \tag{6.16f}$$

$$x_2(t) \leq \Delta T_{max}(t), \ \forall t \in [\tau, \tau + \Delta t_P] \tag{6.16g}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ are positive weighting coefficients. The choice of these coefficients will determine the priority order of different control objectives. In general, the objectives of high homogeneity and accurate final slab temperatures are the most important, then comes the objective of energy efficiency that we want to improve.

6.2.2.2 Performance indexes

The criterion $J_1(U(\tau))$ accounts for the deviation of final average slab temperatures from the desired ones. Let j be the index of a slab that is discharged from the furnace during $[\tau, \tau + \Delta t_P]$, i.e., $j \in S_1(\tau)$, then $J_1(U(\tau))$ is defined by the sum of weighted error of final temperature of slabs discharged during the prediction horizon, i.e.,

$$J_1\left(U(\tau)\right) = \sum_{j \in S_1(\tau)} w_1(z_j(\tau)) \times |T_s(t_{j,1}) - \tilde{T}_s(t_{j,1})|, \quad t_{j,1} \in [\tau, \tau + \Delta t_P], \tag{6.17}$$

where $w_1(z_j(\tau))$ is a positive weighting coefficient corresponding with slab j, and $z_j(\tau)$ is the position of slab j at time τ inside the furnace which refers to the distance from the slab to the furnace exit. The adjustment of $w_1(z_j(\tau))$ is given in Section 6.2.4.

The objective of homogeneous slab temperature profile is expressed as

$$J_2(U(\tau)) = \sum_{j \in S_1(\tau)} w_2(z_j(\tau)) \times g(t_{j,1}), \qquad t_{j,1} \in [\tau, \tau + \Delta t_P], \tag{6.18}$$

where $w_2(z_j(\tau))$ is a positive weighting coefficient corresponding to slab j.

The criterion on consumed energy can be formulated directly with the power of furnace zone as:

$$J_3(U(\tau)) = \int_{\tau}^{\tau+\Delta t_P} w_3(t) \sum_{i=1}^{M} P_i(t) dt \tag{6.19}$$

where $P_i(t)$ is the heat power supplied to zone i at time t, $w_3(t)$ is a weighting coefficient dependent on time instant t. In fact, $P_i(t)$ is predicted by the furnace model, therefore, prediction errors may constitute a problem. As a result, instead of directly calculate total consumption of the the furnace over the prediction horizon, we introduce the time-dependent weighting coefficient $w_3(t)$ to reduce the effects of prediction errors.

Nevertheless, the zone temperatures are linked proportionally to the energy consumption of the furnace. Thus, an optimization strategy will have to maintain the overall furnace temperature as low as possible. The corresponding objective function $J_4(U(\tau))$ accounting for zone temperature is given as

$$J_4(U(\tau)) = \int_{\tau}^{\tau+\Delta t_P} w_4(t) \sum_{i=1}^{M} u_i(t) dt \tag{6.20}$$

where $w_4(t)$ is a weighting coefficient dependent on time instant t.

The slab temperatures should be kept as low as possible to prevent scale formation and decarburization. Therefore, the corresponding objective function $J_5(x(t))$ will take into account the maximum temperatures inside slabs which are represented by the vector $x_1(t)$ defined in Equation 6.14. The formulation of $J_5(x(t))$ is given by

$$J_5(U(\tau)) = \int_{\tau}^{\tau+\Delta t_P} [x_1'(t) \times w_5(t) \times x_1(t)] dt \tag{6.21}$$

where $w_5(t)$ is a weighting positive definite square matrix having the same dimension as length of $x_1(t)$.

We have introduced different coefficients in the description of objective functions. In contrast to weighting coefficients α_i ($i = 1 \div 5$) (c.f Equation 6.16) which determine the priority order of the control objectives in the overall objective function, $\omega_1(z_j(\tau))$ and $\omega_2(z_j(\tau))$, associating with slab j, define the priority order of the discharged slabs in the corresponding objective functions $J_1(U(\tau))$ and $J_2(U(\tau))$. That is because the latter objective functions involve with discontinuous variables: the final temperature gradient $g(t)$ and the final slab temperature $T_s(t)$. For the objective functions that concern continuous variables such as: the heat power $P_i(t)$, the zone temperature set-points (control inputs) $u_i(t)$, and the maximum slab temperature vector $x_1(t)$, the corresponding weighting coefficient $\omega_3(t)$, $\omega_4(t)$, and $\omega_5(t)$ are time-dependent in order to reduce the effect of the prediction errors.

In the following, constraints in the minimization problem 6.16 are reformulated in forms of penalty terms in order to transform it into an unconstrained minimization problem.

6.2.2.3 Approximate constraint criterion

In the problem statement, we have identified the constraints of control task. Hard constraints of a slab j are the upper limit of slab temperature $T_{j,max}$ and maximum values of temperature gradient $\Delta T_{j,max}$. Besides, soft constraints of the slab j are the upper (lower) bound of final temperature $T_{j,end,max}$ ($T_{j,end,min}$), and the upper bound of temperature gradient $\Delta T_{j,end,max}$. In practice, hard constraints are crucial to prevent incidents such as furnace stoppages due to slab bending or deformation. Fortunately, these hard constraints can be satisfied more easily than soft constraints which needs high accuracy of temperature control. As a results, we will corporate not only the soft constraints as penalty terms but also the hard constraints. In this way, the constrained optimization problem is transformed into an unconstrained one. Firstly, limits of zone temperatures and their rate of change are taken into account as follows:

$$J_{c,1}(\tau) = \int_{\tau}^{\tau+\Delta t_C} \sum_{i=1}^{M} [\max(0, u_i(t) - T_{z,i,max}(t)) + |\min(0, u_i(t) - T_{z,i,min}(t))|] dt \tag{6.22a}$$

$$J_{c,2}(\tau) = \int_{\tau}^{\tau+\Delta t_C} \sum_{i=1}^{M} [\max(0, \dot{u}_i(t) - \dot{T}_{z,i,max}(t)) + |\min(0, \dot{u}_i(t) - \dot{T}_{z,i,min}(t))|] dt \tag{6.22b}$$

The idea in these formulations is that if the control inputs or their rate of change get out of their upper and lower limits the penalty terms $J_{c,1}(\tau)$ and $J_{c,2}(\tau)$ will take a positive values . Otherwise, they take values of zero.

As defined in Equation 6.9, the final temperature gradient variable $g(t)$ is a discontinuous output of the system. Therefore, constraints on $g(t)$ can be expressed either in integral continuous form or as sum of discontinuous values, i.e.,

$$J_{c,3}(\tau) = \int_{\tau}^{\tau+\Delta t_P} \max(0, g(t) - g_{max}(t)) dt \tag{6.23a}$$

$$= \sum_{j \in S_1(\tau)} \max\left(0, g(t_{j,1}) - \Delta T_{j,end,max}\right). \tag{6.23b}$$

If the final temperature gradient of one slab or more exceeds its upper bound, $J_{c,3}(\tau)$ will take a positive values, otherwise, it becomes zero.

Similarly, soft constraint on final temperature of slabs is formulated as:

$$J_{c,4}(\tau) = \int_{\tau}^{\tau+\Delta t_P} [\max(0, T_s(t) - \tilde{T}_{s,max}(t)) + |\min(0, T_s(t) - \tilde{T}_{s,min}(t))|] dt \tag{6.24a}$$

$$= \sum_{j \in S_1(\tau)} \max(0, T_s(t_{j,1}) - \tilde{T}_{s,max}(t_{j,1})) + |\min(0, T_s(t_{j,1}) - \tilde{T}_{s,min}(t_{j,1}))|] \tag{6.24b}$$

Hard constraints on slab temperatures and temperature gradients can be formulated as

$$J_{c,5}(\tau) = \int_{\tau}^{\tau+\Delta t_P} [\max(0, [x_1(t) - T_{max}(t)]'[x_1(t) - T_{max}(t)])]dt \tag{6.25}$$

$$J_{c,6}(\tau) = \int_{\tau}^{\tau+\Delta t_P} [\max(0, [x_2(t) - \Delta T_{max}(t)]'[x_2(t) - \Delta T_{max}(t)])]dt \tag{6.26}$$

Global objective function takes into account the control objectives and the temperature constraints:

$$J_o(U(\tau)) = \sum_{i=1}^{5} \alpha_i \times J_i(U(\tau)) + \mu \sum_{i=1}^{6} J_{c,i}(U(\tau)) \tag{6.27}$$

where μ is a positive weighting coefficient. It can be noticed that the penalty terms do not contain weighting coefficient in contrast to objective functions (previous section) in which weighting coefficients are introduced to impose a priority order between slabs or to cope with prediction errors. μ is often imposed with huge values compared to α_i ($i = 1 \div 5$). The main idea is to be able to penalize the overall objective function $J_o(U(\tau))$ whenever a single constraint is not satisfied.

The optimization becomes an unconstrained minimization problem.

$$\min_{U(\tau)} J_o(U(\tau)) = \sum_{i=1}^{5} \alpha_i \times J_i(U(\tau)) + \mu \sum_{i=1}^{6} J_{c,i}(U(\tau)). \tag{6.28}$$

6.2.3 Discrete-time problem

In order to implement the control strategy online, the problem has to be discritized in discrete-time domain. Sampling time for level 2 is generally higher than for level 1. Some terminological definitions are added or redefined for discrete time k:

- Sampling time of level 2: Δt_s
- Prediction horizon $[k, k + N_P]$, with $N_P = int(\Delta t_P / \Delta t_s)$
- Control horizon $[k, k + N_C - 1]$, with $N_C = int(\Delta t_C / \Delta t_s) \leq N_P$
- Control input sequence $U(k) = \{u(k), \forall t \in [k, k+N_C-1]\}$. It shall be noted that $u(k) = u(N_C), \ \forall k \in (N_C, N_P]$.
- Discrete times of loading and discharging events of a slab j are denoted: $k_{j,0}$ and $k_{j,1}$, respectively. Their relations with continuous times of loading and discharging events is defined by the follows:

$$k_{j,0} = int(t_{j,0} / \Delta t_s) \tag{6.29a}$$

$$k_{j,1} = int(t_{j,1} / \Delta t_s). \tag{6.29b}$$

A recapitulation of continuous parameters and their corresponding discrete forms is given in Table 6.3

Parameter Names	Continuous notation	Discrete notation
Loading time of slab j ...	$t_{j,0}$	$k_{j,0}$
Discharging time of slab j	$t_{j,1}$	$k_{j,1}$
Time instant	τ	k
Sampling time	$\Delta\tau_s$	Δt_s
Prediction horizon	Δt_P	N_P
Control horizon	Δt_C	N_C

Table 6.3: Correspondence of continuous and discrete parameters

The unconstrained minimization problem can be written as

$$\min_{U(k)} J_o(U(k)) = \sum_{i=1}^{5} \alpha_i \times J_i(U(k)) + \mu \sum_{i=1}^{6} J_{c,i}(U(k)) \tag{6.30}$$

where $J_i(U(k))$, $i = 1 \div 5$ and $J_{c,i}(k)$, $i = 1 \div 6$ are straightforwardly obtained from continuous forms:

$$J_1\left(U(k)\right) = \sum_{j \in S_1(\tau)} w_1(z_j(k)) \times |T_s(k_{j,1}) - \tilde{T}_s(k_{j,1})|, \qquad k_{j,1} \in [k, k + N_P]. \tag{6.31}$$

$$J_2(U(k)) = \sum_{j \in S_1(\tau)} w_2(z_j(k)) \times g(k_{j,1}), \qquad k_{j,1} \in [k, k + N_P]. \tag{6.32}$$

$$J_3(U(k)) = \sum_{i=1}^{N_P} w_3(k+i) \sum_{j=1}^{M} P_j(k+i) \tag{6.33}$$

$$J_4(U(k)) = \sum_{i=1}^{N_C} w_4(k+i) \sum_{j=1}^{M} u_j(k+i) \tag{6.34}$$

$$J_5(U(k)) = \sum_{i=1}^{N_P} [x_1'(k+i) \times w_5(k+i) \times x_1(k+i)] \tag{6.35}$$

$$J_{c,1}(k) = \sum_{j=1}^{N_P} \sum_{i=1}^{M} [\max(0, u_i(k+j) - T_{z,i,max}(k+j)) + |\min(0, u_i(k+j) - T_{z,i,min}(k+j))|] \tag{6.36a}$$

$$J_{c,2}(k) = \sum_{j=1}^{N_P} \sum_{i=1}^{M} [\max(0, \dot{u}_i(k+j) - \dot{T}_{z,i,max}(k+j)) + |\min(0, \dot{u}_i(k+j) - \dot{T}_{z,i,min}(k+j))|] \tag{6.36b}$$

$$J_{c,3}(k) = \sum_{i=1}^{N_P} \max(0, g(k+i) - g_{max}(k+i)) \tag{6.37a}$$

$$= \sum_{j \in S_1(\tau)} \max\left(0, g(k_{j,1}) - \Delta T_{j,end,max}\right). \tag{6.37b}$$

$$J_{c,4}(k) = \sum_{i=1}^{N_P} [\max(0, T_s(k+i) - T_{s,max}(k+i)) + |\min(0, T_s(t) - T_{s,min}(k+i))|] \tag{6.38a}$$

$$= \sum_{j \in S_1(\tau)} \max(0, T_s(k_{j,1}) - T_{s,max}(k_{j,1})) + |\min(0, T_s(k_{j,1}) - T_{s,min}(k_{j,1}))|] \tag{6.38b}$$

$$J_{c,5}(k) = \sum_{i=1}^{N_P} [\max(0, [x_1(k+i) - T_{max}(k+i)]'[x_1(k+i) - T_{max}(k+i)])] \tag{6.39}$$

$$J_{c,6}(k) = \sum_{i=1}^{N_P} [\max(0, [x_2(k+i) - \Delta T_{max}(k+i)]'[x_2(k+i) - \Delta T_{max}(k+i)])] \tag{6.40}$$

6.2.4 Parameters adjustment

Based on the dynamic optimization problem (6.30), optimal control input $u(k) = [\tilde{T}_{z,1}(k), ..., \tilde{T}_{z,M}(k)]$ is computed for the system-wide. Therefore, the design of cost function is crucial for the performance of the control. The cost function contains objective functions and penalty terms of furnace constraints that ensure the accurate heating of slabs. The fuel consumption of furnace predicted by non-steady-state model is accounted in the cost function, which assures the energy efficiency of the heating. General objectives that contribute to reduction of energy, loss of material are the minimization of furnace zone temperatures and the slab temperatures.

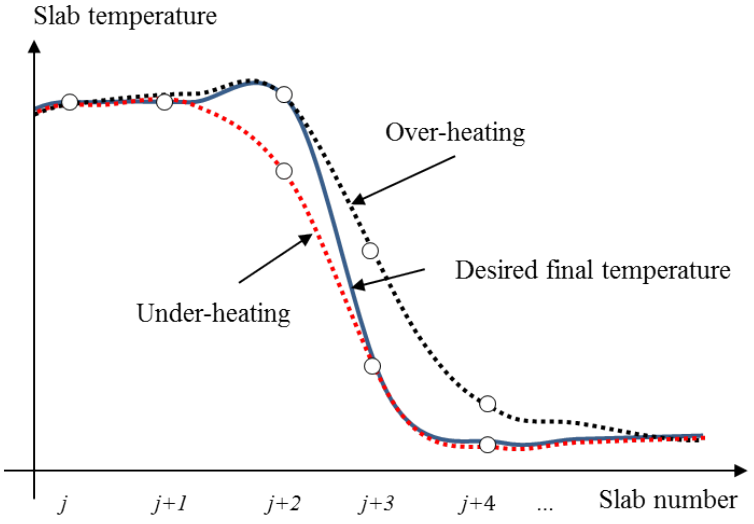

Figure 6.10: Heating of consecutive slabs that have big difference on desired final temperature

In the formulation of objective functions, we use weighting coefficients associated with slabs as: $w_1(z_j(k))$, $w_2(z_j(k))$ with $j \in S_1(\tau)$ the index set of slabs that are discharged during the prediction horizon. The relative ranking of these coefficients is the information of interest rather than their absolute values. That is, these coefficients will be defined by how a slab is considered more important than another. On one hand, a slab that will be discharged from the furnace before another must have a higher heating priority. This is to ensure that desired average final slab temperature ($\tilde{T}_s(t)$) and required slab temperature homogeneity (imposed with $g_{max}(t)$) are achieved. On the other hand, in a normal furnace operation, it happens that two consecutive slabs may have very different desired final temperatures. In this case, due to the slow dynamics of the furnace, it is difficult or even impossible for the furnace control to correctly realize desired final temperature of both. The control may either under-heat one slab and correctly heat the other, or overheat one slab and correctly heat the other as illustrated in Figure 6.10. Under-heated slab risks to be rejected by hot rolling mills, which represents a loss of energy and material. Overheating of both of slabs will need more energy, but generally less expensive than the handling of slab rejections. For this reason, overheating is preferred to under-heating. Therefore, between two slabs, one with higher desired final temperature will be more prioritized. In addition, some slabs may have extra requirements of quality, this could be also taken into account into the weighting coefficients. To recapitulate $w_1(z_j(k))$, $w_2(z_j(k)$ can have the following forms accounting for criteria of: slab position, desired final slab temperature, quality requirement, i.e.,

$$w_1(z_j(k)) = \gamma(z_j(k)) \times \gamma_{T,j} \tag{6.41a}$$

$$w_2(z_j(k)) = \beta(z_j(k)), \tag{6.41b}$$

provided that $\gamma(z_j(k))$, $\beta(z_j(k))$ are functions of slab position at time k and $\gamma_{T,j}$ associates with desired final slab temperature. Let slabs l, j be discharged with desired final temperature $\tilde{T}_{l,end}$, and $\tilde{T}_{j,end}$, respectively. If the slab l is discharged before slab j meaning $z_l(k) < z_j(k)$ then the corresponding weighting coefficients of slab l have bigger values, i.e., $\gamma(z_l(k)) > \gamma(z_j(k))$ and $\beta(z_l(k)) > \beta(z_j(k))$. This implies also that the more a slab approaches the furnace exit the heavier it is weighted in the objective functions as illustrated in Figures 6.11 and 6.13. $\gamma(z_j(k))$ can take the following form:

$$\gamma(z_j(k)) = \frac{L - z_j(k)}{L} \tag{6.42}$$

where L is the furnace length. Similarly, if slab j has a higher desired final temperature, i.e., $\tilde{T}_{l,end} > \tilde{T}_{j,end}$ then it is weighted more, i.e.,$\gamma_{T,j} > \gamma_{T,i}$. For example, $\gamma_{T,j}$ can take the following form accounting the desired final slab temperature

$$\gamma_{T,j} = \frac{\tilde{T}_{j,end}}{\tilde{T}_{end,nom}} \tag{6.43}$$

where $\tilde{T}_{end,nom}$ is a nominal value of desired final slab temperature.

Furthermore, each furnace zone is dedicated to a specific stage of heating process, e.g., zone soaking heats slabs from 1200°C to 1250°C, zone heating 2 from 1000°C to 1200°C, zone heating 1 from 600°C to 1000°C, zone preheating from 300°C to 600°C as shown in Figure 6.12. The zone soaking is dedicated mostly to equalize the

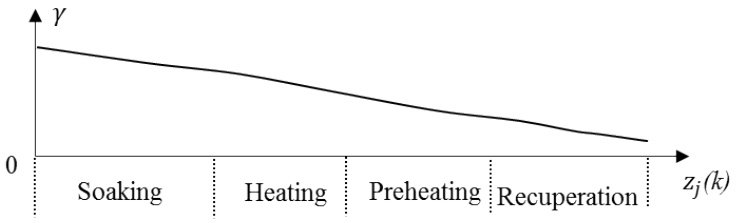

Figure 6.11: Variation of weighting coefficient on final slab temperature error according to slab position

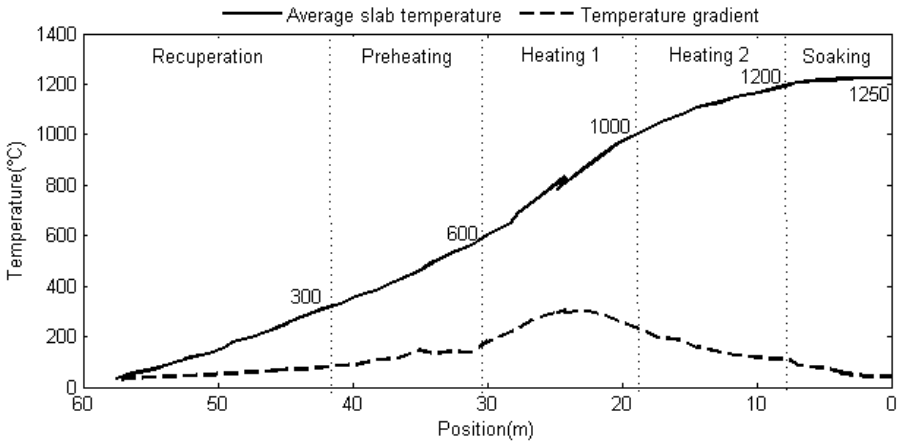

Figure 6.12: Different stages of heating process

temperature inside slabs, which means that temperature gradient of slabs situating inside soaking zone must be smaller than of slabs in other zones. Therefore, $\beta(z_j(k))$ should have big values if slab j is inside soaking zone and much smaller if inside other zones as depicted in Figure 6.13. $\beta(z_j(k))$ can be calculated with the following form

$$\beta(z_j(k)) = \begin{cases} \frac{L-z_j(k)}{L}, \text{ if } j \text{ is located in soaking zone} \\ \eta\frac{L-z_j(k)}{L}, \eta \in [0,1], \text{ otherwise} \end{cases} \tag{6.44}$$

where η is a positive constant rescaling the weighting coefficient of slabs located in heating zones, preheating and recuperation zones.

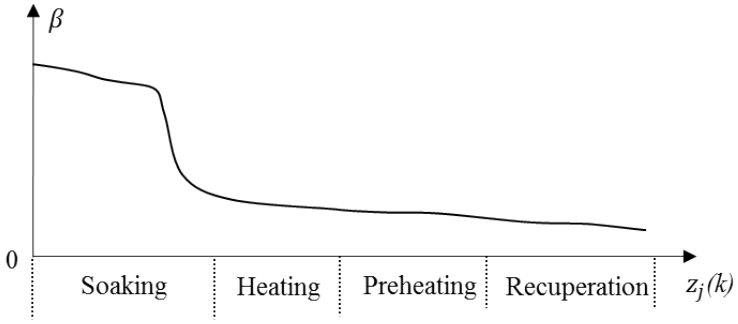

Figure 6.13: Variation of weighting coefficient on temperature gradient according to slab position

Other coefficients $w_4(k+i)$ $(i \in [1,2,...,N_C])$, $w_3(k+i), w_5(k+i)$ $(i \in [1,2,...,N_P])$ associate with fuel

consumption, general temperature level of furnace, and slab temperature trajectory, respectively. The later variables are predicted based on the non-steady-state model. Error from the predictions can deviate the solution of the optimization problem. Therefore, the mentioned coefficients are decreasing from the beginning to the end of the horizons (N_P or N_C), i.e.,

$$\begin{aligned} &\text{if } (j > i) \\ &\text{then : } \; w_3(k+j) < w_3(k+i), w_4(k+j) < w_4(k+i), \text{and } w_5(k+j) < w_5(k+i). \end{aligned} \tag{6.45}$$

Simple linear form of $w_3(k), w_4(k)$ and $w_5(k)$ can be defined as

$$\begin{aligned} w_4(k+i) &= (N_C - i)/N_C, \;\; \forall i \in [1, 2, ..., N_C], \\ w_3(k+i) &= (N_P - i)/N_P, \;\; \forall i \in [1, 2, ..., N_P], \\ w_5(k+i) &= (N_P - i)/N_P, \;\; \forall i \in [1, 2, ..., N_P]. \end{aligned} \tag{6.46}$$

A recapitulation of different weighting coefficient is given in Figure 6.14.

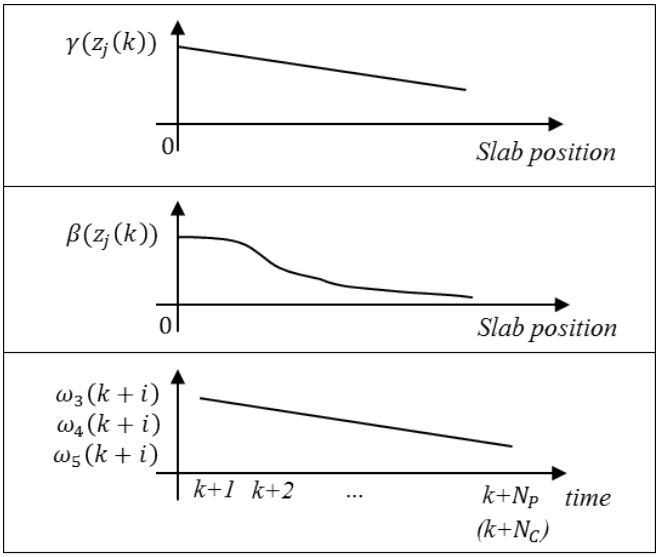

Figure 6.14: Weighting coefficients according to slab position and time

The weighting coefficients $\alpha_i (i = 1 \div 5)$ determine the priority order between control objectives. In general, low error of final slab temperatures and low temperature gradients are the most important control objectives. And, reduction of energy consumption comes as the less important control objective, however it is the main objective of the work. Penalty terms are weighted by the coefficient μ which is usually imposed with a big value in order to penalize the overall objective function $J_o(U(k))$ whenever a single constraint is not satisfied. To recap, the relative rank of the coefficients μ and $\alpha_i (i = 1 \div 5)$ can be described as

$$\mu \gg \alpha_1, \alpha_2 > \alpha_3, \alpha_4, \alpha_5. \tag{6.47}$$

In order to determine the prediction horizon, we should note that, if a long optimization horizon N_P is chosen, the optimal solution of time instant k, i. e., $U(k)$ may be completely different from optimal solution of time instant $k + N_P$, i. e., $U(k + N_P)$. Especially, in reheating furnaces which have switched dynamics due to the continuous flow of slabs passing through the furnace interior, the control input $u(k)$ applied at time k has significant influence in a *near* future, but this influence is weakened as time goes on and practically negligible after some time. In the furnace control problem, the prediction horizon should be greater than the time that a slab resides in controllable zones (preheating, heating 1, heating 2, soaking). In average, this period corresponds to $2/3$ the residence time of a slab inside the furnace. Therefore we have the conditions for prediction horizon:

$$\Delta t_P \geq \frac{2}{3}(t_{j,1} - t_{j,0}), \tag{6.48}$$

with j a representative slab. For the considered furnace the average residence time of slabs is 3 h. Thus, $\Delta t_P = 120$ min is an appropriate choice.

The control horizon N_C should be chosen with consideration of the available computational resource of the control system. $N_C \times \Delta t_s$ is lower bounded by the delay of the furnace which has at least the value of sampling period Δt_s. In fact, Δt_s should be chosen according to the changes of system dynamics. These changes are associated with loading and discharging events. For the considered furnace, average of time interval between two consecutive loading (discharging) events is about 5 min, thus we might chose $\Delta t_s = 5$min. As a result, the prediction horizon N_P takes a value of $\Delta t_P / \Delta t_s = 24$.

6.3 Optimization methods

In order to solve the unconstrained minimization problem in 6.30, the objective function is calculated based on variables predicted by a simulation of the furnace. This simulation is constructed based on the non-steady-state model presented in chapter 3. The numerical simulation involves heat exchange between slabs and their surrounding environment, switching of furnace load, movement of slabs inside furnace, energy losses into door openings or the cooling system, etc. Therefore, an analytic formula of the objective function cannot be obtained. As a result, derivatives of objective function are not explicitly formulated.

In term of optimization methods, there are two major classes: gradient-based methods, and gradient-free methods. Gradient-based methods demand the function to be continuously differentiable. If only first order gradients are needed, the method is often referred to as *gradient search* or *first order gradient search*. Examples of first order methods are method of steepest descent and line search method. If first and second order gradients are demanded, the optimization methods are referred to as *second order methods* such as Newton-Raphson and quasi-Newton methods.

Gradient-free methods are methods that need only values of objective function or the ranking of these values. These methods are also referred to as direct search methods [LTT00a, Tor89, HJ61, KLT03].

For reheating furnaces, utilization of gradient-based methods will need finite-differencing of the objective function. This choice will require repeating furnace simulation in order to evaluate the objective function $J_o(U(k))$ with different values of the control input $U(k)$. Because real-time implementation needs the optimization algorithm to provide optimal control input in period of one sample time, the repetition of furnace simulation is limited within a certain number l_{max}. Besides, provided non-steady-state model is more elaborated than the steady-state model and its computation time is also considerably longer. But we also want to benefit the advantage of non-steady-state model in the optimization. Therefore, gradient-based optimization methods will not be studied further. Utilization of gradient-free methods will be more appropriate for optimization problem of reheating furnace.

In the following, a more detailed of furnace control system is described. The existing level 2 of the furnace is presented showing how furnace model is used in furnace observer and set point optimizer. Afterwards, the basic ideas of different direct search methods are presented. Considering the characteristic of reheating furnace optimization problem, an appropriate method will then be chosen and adapted for the considered furnace.

6.3.1 Simulation-based optimization

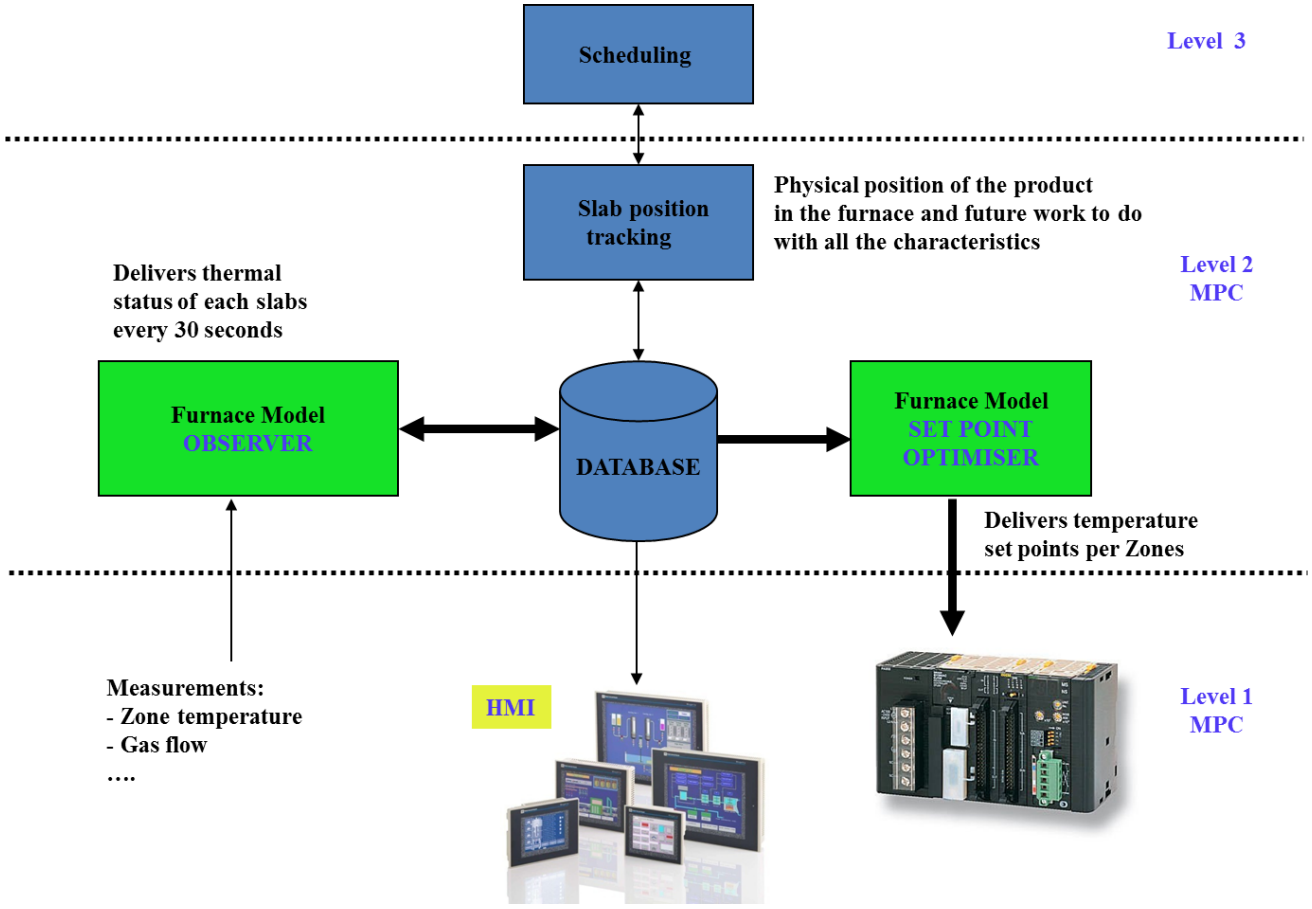

Figure 6.15: Architecture of the furnace control system

In the existing level 2 of reheating furnace, the steady-state model of the furnace is used for two purposes: furnace observer and set point optimizer as shown in the architecture of the furnace control system on Figure 6.15. In this architecture, level 3 provides the predefined scheduling of slabs to the slab position tracking program of level 2. This program tracks the exact physical position of slabs at each time instant and notify the new move of slabs. In the furnace observer, real-time measurements of zone temperatures are fed to the steady-state model to estimate furnace variables that cannot be measured directly such as slab temperature profiles. The observer has to run in high frequency (every 30 seconds) and precision to provide sufficient and accurate information of slab temperature profiles (stored in database of the system) for the furnace control and monitoring task. In the set point optimizer, the steady-state model is utilized to simulate the furnace operation. This simulation predicts the slab temperature profiles from given zone temperature set-points. In comparison to the model used by observer, the steady-state model used by set point observer is executed with less exigence on precision meaning greater sample time and spatial sampling as the example of Table 6.4. Instead of delivering slab temperature profiles every 30 seconds at 20 points inside the slab thickness as for the furnace observer, the furnace model will only do the calculation every 60 seconds at 10 points for the set-point optimizer.

Parameter	Observer model	Set point optimizer model
Sample Time ...	30 seconds	60 seconds
Spatial sampling	20 points	10 points

Table 6.4: Difference between steady-state model used in furnace observer and set-point optimizer

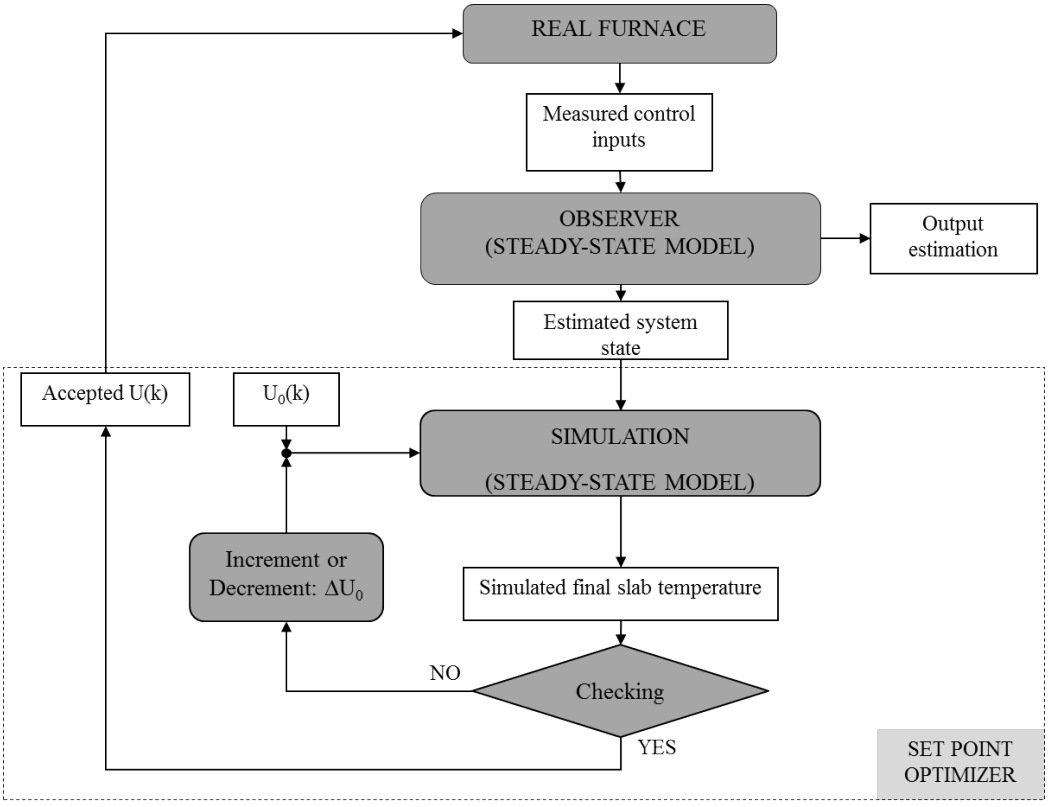

Figure 6.16: Simulation-based calculation of the existing level 2 of the furnace

The zone temperature set points are calculated by repeating computer program which simulates the furnace operations as shown in Figure 6.16. With given zone temperature set points, simulation results indicate if the requirement of desired end-zone slab temperatures would be satisfied or not. If the simulated end-zone slab temperatures are lower than their desired values, the zone temperature set points are incremented with a step ΔU_0. Inversely, if the simulated end-zone slab temperatures are too much higher than their desired values (overheating), the zone temperature set points are decremented with the step ΔU_0. When the simulated end-zone temperatures of all slabs located inside a furnace match their requirements, the zone temperature set points are accepted and applied to the furnace. Zone temperatures realized by the temperature control of level 1 are measured and utilized in the observer to estimates the system state (slab temperatures). The system state is then utilized by simulation program to predict end-zone slab temperature. These procedures restart at each sampling time.

It can be noticed that the method used in the existing level 2 is simply calculating adequate zone temperature set points to achieve the desired final slab temperature. The method is not optimizing the furnace heating process in terms of energy consumption and final slab temperature at the same time. In the following section, the solution for optimization problem of the furnace previously defined in Equation 6.30 is studied. Different optimization methods are described. The appropriate method is then chosen for our problem.

6.3.2 Direct search methods

Direct search methods are reviewed in [LTT00a, KLT03]. The authors show the distinguishing characteristic of direct search methods compared to other methods where either explicit gradients or their approximated values are used. From practical point of view, three classical direct search methods concerning unconstrained minimization are classified as: pattern search method, simplex method (different from simplex method for linear programming), and method with adaptive sets of search directions.

6.3.2.1 Pattern search methods

A generalized formulation of pattern search methods is given in [Tor97]. We consider a problem of minimizing a continuously differentiable function

$$\min_{x \in \Re^n} f(x) \tag{6.49}$$

$$\text{where } f(x) : \Re^n \rightarrow R. \tag{6.50}$$

We denote by $\Re$, Q, Z, and N the sets of real, rational, integer, and natural numbers, respectively. If we use direct search methods for this problem then neither analytic nor approximated derivatives of $f(x)$ are needed. Basic idea of pattern search methods is utilization of a series of *exploratory moves* that consider the behavior of the function f at a pattern of points (see Figure 6.17) to find the minimum.

Firstly, to define a pattern, we need two components called a *basic matrix* and a *generating matrix*. The basic matrix can be given by any nonsingular matrix $B \in \Re^{n \times n}$. And the generating matrix is denoted $C_k \in Z^{n \times p}$,

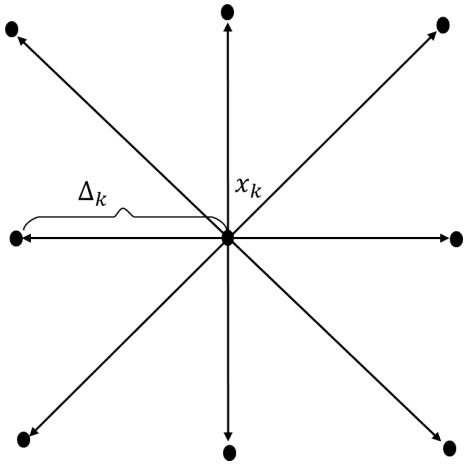

Figure 6.17: Example of search pattern in $\Re^2$ with a given step length parameter Δ_k

with k is the iterate index and $p > 2n$, that is C_k may be actualized every iterate k. And C_k is partitioned into components as

$$C_k = [M_k \mid -M_k \mid L_k] = [\Gamma_k \;\; L_k] \tag{6.51}$$

where $M_k \in M \subset Z^{n\times n}$, with M is a finite set of nonsingular matrices, and $L_k \in Z^{n\times(p-2n)}$ and has at least one column of zeros. Note that both B and C_k have rank n.

Then, we define pattern P_k by

$$P_k = BC_k = [BM_k \mid -BM_k \mid BL_k] = [B\Gamma_k \mid BL_k] \tag{6.52}$$

Because B and C_k have rank n, the column vectors of P_k span $\Re^n$. Provided $\Delta_k \in \Re$, and $\Delta_k > 0$ as a step length parameter, we define a *trial step* s_k^i as a vector calculated by

$$s_k^i = \Delta_k B c_k^i, \tag{6.53}$$

where c_k^i denotes a column of $C_k = [c_k^1 ... c_k^p]$. We can see that Bc_k^i determines the direction of the trial step. Then, a *trial point* is defined as any point having the form $x_k^i = x_k + s_k^i$, with x_k is the current iterate. In Figure 6.17, an example of search pattern in $\Re^2$ is illustrated. From he current iterate x_k, eight trial points are defined with the step length Δ_k. In this example, the basic matrix B is the identity matrix, i.e., $B = [1\ 0; 0\ 1]$. The generating matrix C_k is fixed for all k. In this case, $C_k = C_0$, where

$$C_0 = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}.$$

These parameters are defined for the *coordinate search algorithm* provided in the Appendix B. The pattern P_k is used to guide a series of search from the current iterate x_k for a new iterate $x_{k+1} = x_k + s_k$ giving a lower function value, i.e., $f(x_{k+1}) \leq f(x_k)$. These series of search is referred to the explanatory moves mentioned previously. At one extreme, an effective explanatory moves algorithm would be one that somehow guesses which of the

steps defined by the columns of the matrix $\Delta_k P_k$ will produce a simple decrease (to be differed from *sufficient decrease* as for gradient-base methods like *steepest descent method*), and subsequently evaluate $f(x)$ at only one point. This will reduce a lots of required computational resource. However, it is not always easy to do such guessing. At the other extreme, an explanatory moves algorithm would evaluate all possible moves defined by $\Delta_k s_k^i \in \Delta_k P_k$, $i = 1 \div p$ (s_k^i is a column of P_k), and returns the trial step that produces the smallest function values.

A pattern search method is distinguished partly with other methods by its explanatory moves algorithm. In a generalized description of pattern search methods, we want to allow the maximum possible choice of explanatory moves algorithm and still maintain the required properties for the convergence analysis. These requirements are given in the following hypothesis of explanatory moves.

Hypostheses of explanatory moves

- (1) $s_k \in \Delta_k P_k \equiv \Delta_k B C_k \equiv \Delta_k [B\Gamma_k \; BL_k]$.
- (2) If $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

The *algorithm of generalized pattern search method* is given as follows:

Algorithm of generalized pattern search method

Let initial variables x_0 and Δ_0 be given a priori. Then for $k = 0, 1, ...,$

- (1) Compute $f(x_k)$.
- (2) Determine a step s_k utilizing an explanatory moves algorithm.
- (3) Compute $\epsilon_k = f(x_k) - f(x_k + s_k)$.
- (4) If $\epsilon_k > 0$ then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
- (5) Update C_k and Δ_k.

A particular pattern search method specified by: the basic matrix, the generating matrix, the explanatory moves algorithm utilized to determine the step s_k, and the algorithms for updating C_k and Δ_k.

Two components: the generating matrix C_k and the step length Δ_k need to be updated every iterate k. Generally, we want to update Δ_k so that $\epsilon_k > 0$ which means the iteration k is successful, otherwise it fails to find new iterate that decreases the function value. It should be recalled that to accept a step, we only need a simple, as different from sufficient, decrease of function value. The idea for updating Δ_k is to reduce it when the iteration

is unsuccessful and to increase it when the iteration is successful. Therefore, updating algorithm can be given by

---

Updating of step length Δ_k

- If $\epsilon_k \leq 0$ then $\Delta_{k+1} = \theta\Delta_k$ where $0 < \theta < 1$.
- If $\epsilon_k > 0$ then $\Delta_{k+1} = \lambda_k\Delta_k$ where $\lambda_k \geq 1$.

θ and λ_k can be determined with given parameters $\tau \in Q$ and $\{\omega_0, \omega_1, ..., \omega_N\} \subset Z$, with $N < +\infty$, $\tau > 1$, $\omega_0 < 0$, and $\omega_i \geq 0$, $i = 1, ..., N$, as follows

$$\theta = \tau^{\omega_0} \tag{6.54}$$

$$\lambda_i = \tau^{\omega_i},\ i = 1, ..., N. \tag{6.55}$$

The algorithm for updating C_k varies with each particular pattern search method. It is sufficient to choose C_k satisfying the form in 6.51 with corresponding conditions placed on matrices $M_k \in M \subset Z^{n\times n}$ and $L_k \in Z^{n\times(p-2n)}$.

For analysis of general convergence results, we define the set $L(x_0)$ as

$$L(x_0) = \{x : f(x) \leq f(x_0)\}. \tag{6.56}$$

Assume that $L(x_0)$ is compact. And that function $f(x)$ is continuously differentiable on a neighborhood of $L(x_0)$. Then for the sequence x_k produced by the *algorithm of genralizedd pattern search method* given above, the following result of convergence is assured:

$$\lim_{k\to+\infty} \inf \|\nabla f(x_k)\| = 0. \tag{6.57}$$

To obtain a stronger convergence results, the hypotheses of explanatory moves must be strengthened as

---

Strong hypotheses of explanatory moves

- (1) $s_k \in \Delta_k P_k \equiv \Delta_k BC_k \equiv \Delta_k[B\Gamma_k \ \ BL_k]$.
- (2) If $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) \leq \min f(x_k + y), y \in \Delta_k B\Gamma_k$.

Assume that $L(x_0)$ is compact, and function $f(x)$ is continuously differentiable on a neighborhood of $L(x_0)$. In addition, if the columns of the generating matrix $C_k = [c_k^1, ..., c_k^p]$ is bounded in norm, i.e., $\exists$ constant $C > 0$,

and $c_k^i \leq C,\ \forall k,\ \forall i = 1, ..., p$, and $\lim_{k\to+\infty} \Delta_k = 0$. Then, the convergence of x_k produced by the *algorithm of generalized pattern search method* satisfying the *strong hypotheses of explanatory moves* is given as

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0. \tag{6.58}$$

6.3.2.2 Simplex methods

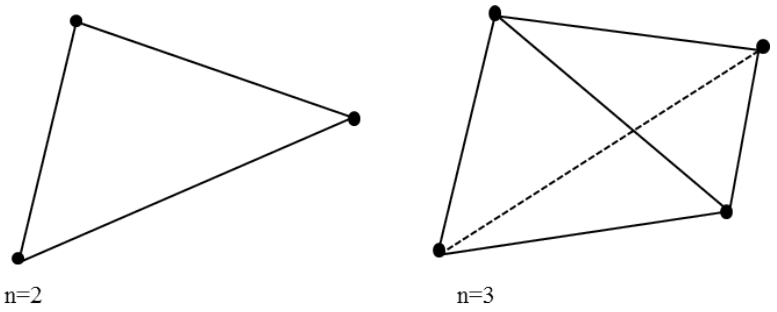

Figure 6.18: Example of a simplex in $\Re^2$ and $\Re^3$

The first simplex method is presented in [SHH62]. Since $n + 1$ values of $f(x)$ would be needed to estimate $\Delta f(x)$ via finite-differencing. The authors made an observation that it needs no more than $n + 1$ evaluations of objective function $f(x)$ to identify a descent. And, because $n+1$ points determine a simplex (a triangle in $\Re^2$, and a tetrahedron in $\Re^3$, as shown in Figure 6.18), new direct search method is developed based on a non-degenerate simplex in $\Re^n$.

A simplex is defined as a set of $n + 1$ points in $\Re^n$. In a non-degenerate simplex, the set of edges adjacent to any vertex forms a basis for the space $\Re^n$. That is, any point in the space R^n can be constructed by taking a combinations of the edged adjacent to any given vertex of the simplex.

In first simplex method algorithm [SHH62], a non-degenerate simplex is firstly initialized. Then, values of objective function are evaluated for $n+1$ vertices. Afterwards, vertices are classified based on their corresponding function values. The best vertex corresponds to the lowest function value. And, the worst vertex is associated with the highest function value.

To find a new minimizer of objective function, the algorithm use a operation called *reflection*. First, the worst vertex is reflected through the centroid point of the opposite face as depicted in Figure 6.19. The new vertex is then evaluated and compared to the others. If the reflected vertex is still the worst vertex, then the next worst vertex of the original simplex is chosen for a new reflection. The procedure is repeated again. The main goal is to either replace the best vertex or assure that the current best vertex is a good candidate for the minimizer of the objective function $f(x)$.

One can be sure that when a reflected vertex produces a strict decrease on objective function compared to the current best vertex, a new candidate for function minimizer is obtained.

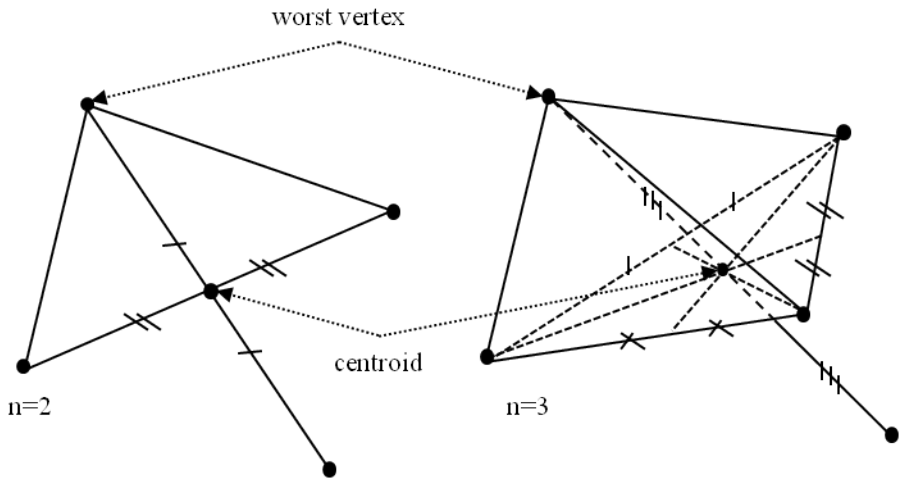

Figure 6.19: Reflection of worst vertex on the centroid of the opposite face in $\Re^2$ and $\Re^3$

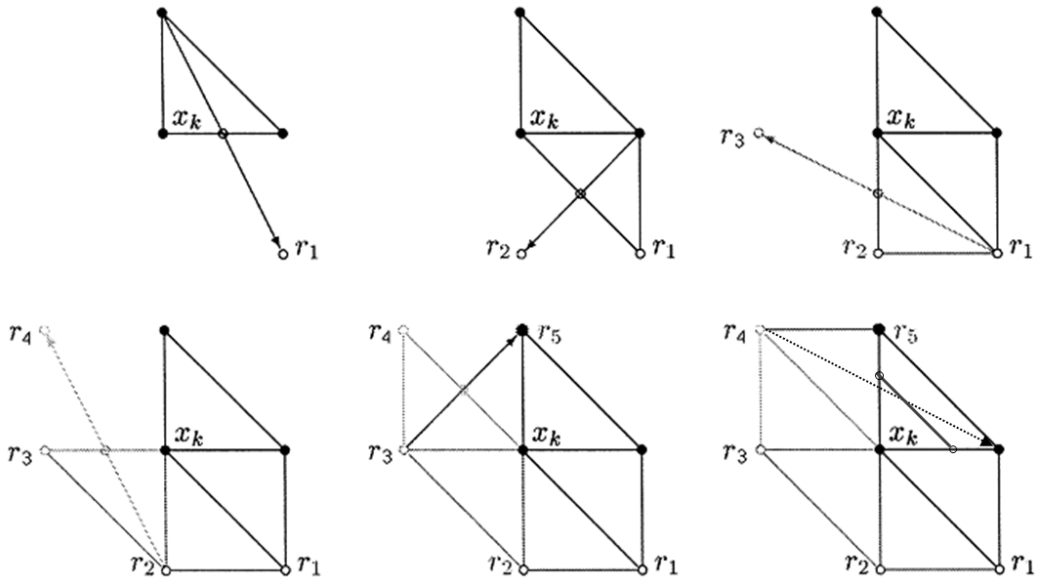

Figure 6.20: Reflections (r_1, r_2, r_3, r_4 and r_5) fail to replace best vertex x_k and the edges adjacent to x_k is reduced to continue the algorithm

However, it happens that after a number of reflection, the simplex becomes the initial one as illustrated in Figure 6.20. In this case, these reflections r_1, r_2, r_3, r_4 and r_5 have failed to replace the best vertex x_k. For this, the algorithm continues by reducing the edges adjacent to the best vertex as depicted in the bottom-right of Figure 6.20.

Nelder-Mead simplex method for optimization, presented in [NM65], is an important development of simplex methods. Authors designed new algorithm to accelerate the search for a minimizer of the objective function by adding operations other than reflection such as contraction and expansion. Different from reflection operation which preserves the original shape of the simplex, additional operations are constructed to accelerate the search by deforming the simplex as shown in Figure 6.21. In an iteration, the algorithm would try all of the three possible moves: reflection, contraction and expansion on every vertex of the initial simplex to find a new best one. If none

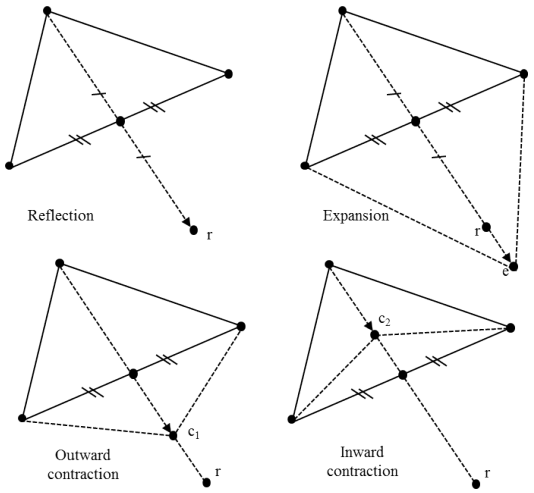

Figure 6.21: Reflection, expansion, and contraction moves of Nedlder-Mead simplex method

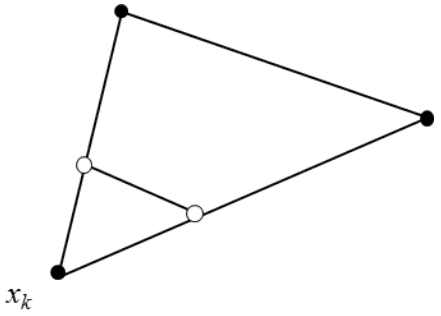

Figure 6.22: Reducing of the simplex when reflection, expansion, contraction moves fail to replace the best vertex x_k

of these moves succeeds, then a new simplex is generated by reducing the edges adjacent to the best vertex x_k as shown in Figure 6.22. The detailed description of Nelder-Mead simplex method is given in Appendix C.

For a constrained problem, a variant of Nelder-mead simplex algorithm which copes with bounds of variables is mentioned in [LLR04]. The bounded variables can leave the domain after either the reflection or the expansion operation. The problem is handled by projection

$$\begin{cases} \text{if } (x_k < x_k^{min}), \text{then } x_k = x_k^{min} \\ \text{if } (x_k > x_k^{max}), \text{then } x_k = x_k^{max} \end{cases} \tag{6.59}$$

Where x_i^{min} and x_i^{max} are lower and upper bounds of the function variable. Authors have also pointed out that by projecting on bounds, the simplex may collapse into subspace of the saturated variables. To tackle the problem, a convergence test based on re-initialization at the point of convergence is then required.

Due to its heuristic characteristic, Nelder-Mead simplex method lacks an explicit mathematical convergence proof. In [LRWW98], convergence results are obtained for strictly convex functions in dimensions 1 and 2. The Nelder-Mead simplex method is proved to converge to a minimizer in 1-dimensional problem. And various limited convergence results is obtained for dimension 2. In [Mck98], the method failed to converge to the minimizer

from a specific starting point for a family of strictly convex functions in two dimensions. Some efforts of improving the original Nelder-Mead method are found in [LLR04, Tse99, PCB02, NT02]. In these studies, strategies for solving the defects of original Nelder-Mead simplex method include using a *sufficient decrease* condition for acceptance of a new vertex rather than a *simple decrease* and restarting if the simplex becomes ill-conditioned, e.g., degenerated into subspace.

On the other hand, with its simple implementation and effectiveness in most of practical applications, Nelder-Mead simplex became the most known direct search method. In december 2014, Google Scholar displayed more than 1200 papers published in 2014 that referred to the Nelder–Mead method for a wide variety of applications. The algorithm is often found in optimization numerical software packages [LTT00b], e.g., in function *fminsearch* in Matlab optimization toolbox. In general, what is noted about the method is that it tends to be able to satisfy optimization purpose by producing a rapid initial declines of function value in comparison to other direct search methods [Wri10].

6.3.2.3 Adaptive methods

The algorithms of this class accelerate the search by adapting search directions to the curvature of the objective function obtained during the course of the search. Underlying algorithms are Rosenbrock's and Powell's methods [Pow64, Ros60].

Rosenbrock's method conducts a series of stages. For each stage, a set of search directions is defined. These search directions are updated from stage to stage to make use of the information acquired about the objective function.

For the initial stage of Rosenbrock's method, the search directions are defined with the coordinate directions. In Figure 6.23 [VB], an example of Rosenbock's algorithm is illustrated for 2-dimensional minimization problem. The algorithm search for improvement of the objective function in axis x_1, x_2 until it finds at least one successful (points 5, 7, 9, 11, 13 for direction of axis x_1 and points 2, 4, 6, 8, 10 for direction of axis x_2) and one unsuccessful step in each of the search directions (the points 3 for axis x_1 and 12 for axis x_2). Then, the stage terminates. If there exists new point that gives the lowest of function value (point 13), then it is used as initial point for the next stage.

For the next stage, instead of using the same search directions, new search directions is generated by rotating the current ones to adapt with the function curvature. For that, the connection between the initial point of the previous stage (point 1) and the new best point (point 13) is taken as a new search direction, with its orthogonal direction, they constitute new set of search directions (defined by points: 1, 13, 18). It should be noted that search directions are orthogonal and therefore they are linearly independent. The search continues and converges to the minimizer of the objective function.

Powell's method uses function modeling as the basic approach. The main intention is to be able to use conjugate directions to accelerate the convergence if the method is applied to a convex quadratic function. Like Rosenbock's

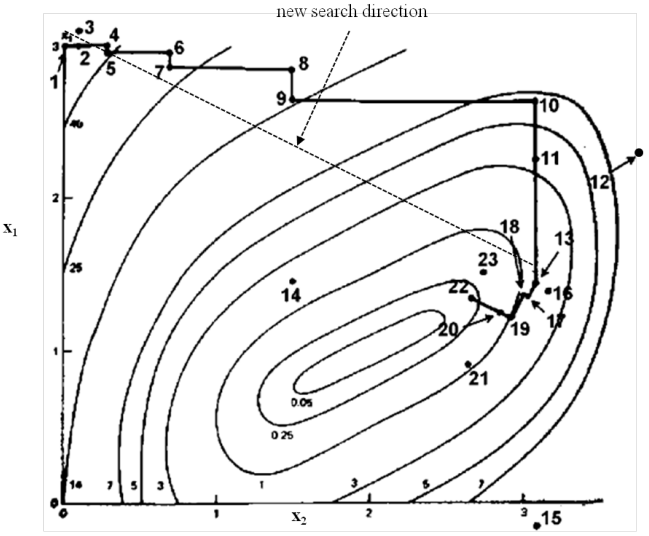

Figure 6.23: Rosenbock's method in problem of dimension 2

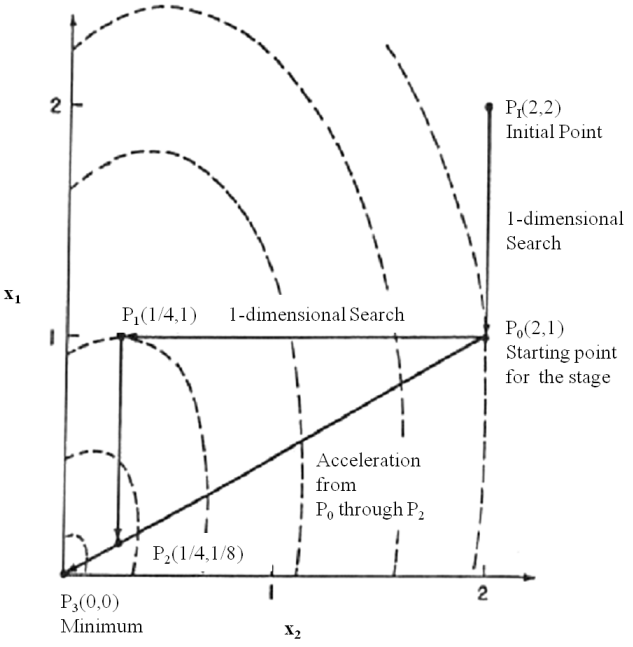

Figure 6.24: Powell's method in problem of dimension 2

method, Powell's method also follows stages. Each stage consists of $n + 1$ 1-dimensional searches which are to find the exact minimizer of a quadratic model of the objective function computed for each search direction. The first n searches are executed along each of linearly independent directions (from P_0 to P_1 and from P_1 to P_2 as illustrated in Figure 6.24 [FPTV92]). The last search $(n + 1)$ is conducted along the direction connecting the point obtained at the end of the first n searches (point P_2 in the example) and the starting point (P_0). The search is accelerated finding new minimum point of the stage (point P_3). At the end of the stage, one of the first n directions is replaced by the last one (connection P_0P_2). In a new stage, the newly found minimum point of the previous stage is used as the starting point, and the whole steps are repeated to search for the minimizer of the objective function.

6.3.3 Adopted optimization strategy

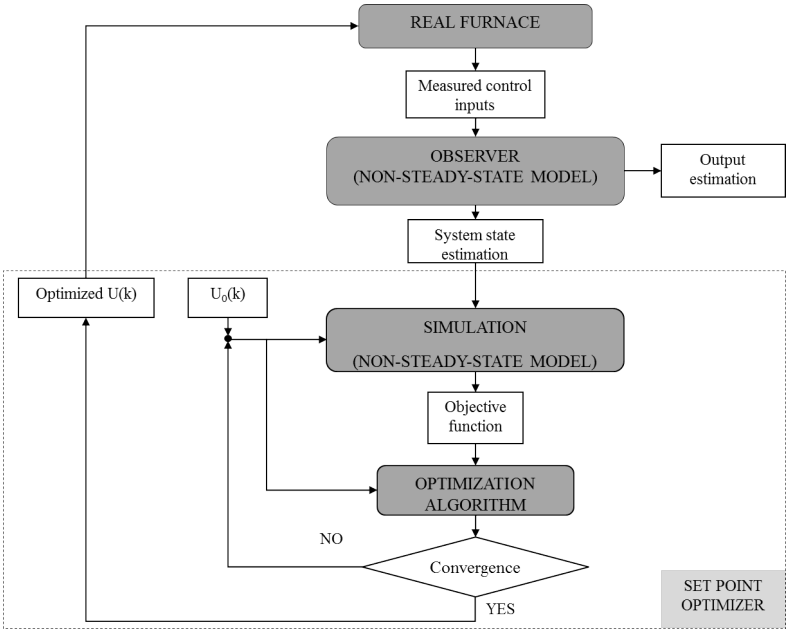

Figure 6.25: Simulation-based optimization of the MPC strategy

In the design of MPC, the choice of optimization algorithm depends on the nature of objective function. For the considered reheating furnace, the optimization problem has been formulated as an unconstrained minimization problem i.e.,

$$\min_{U(k)} J_o(U(k)), \tag{6.60}$$

where $U(k) \in \Re^n$, $n = N_c \times M$, is the control input sequence (zone temperature set-points).

As illustrated in Figure 6.25, the optimization by MPC strategy for the reheating furnace is also a simulation-based method as for the existing method. The part of *setpoint optimizer* in the system includes mainly furnace simulation block, and optimization algorithm block. The furnace simulation calculates the objective function $J_o(k)$ via the non-steady-state furnace model. As formulated in the forgoing sections, J_o contains objective and penalty terms concerning accuracy of final slab temperature, energy consumption and control constraints. To calculate these terms, for sample time k, the simulation of furnace predicts the final slab temperature profiles and furnace energy consumption over the prediction horizon $[k, k+N_p]$. This simulation is executed repetitively from given control input sequence $U(k)$ and the estimations of system state (current temperature profiles of slabs, zone temperatures, etc.). Control input sequence is varied according to the chosen optimization algorithm. Once the control input sequence is optimized according to a given tolerance, it is applied to the real furnace.

On the other hand, a furnace observer takes into account the measured heating power supplied to the furnace zones in order to estimate current temperature profiles of all slabs inside the furnace. Also, this observer estimates the system outputs which are the final temperature profile of the slabs.

$J_o(U(k))$ is a nonlinear function with the following properties:

- Calculation of $J_o(U(k))$ is very time-consuming.
- Exact first partial derivatives of $J_o(U(k))$ cannot be calculated.
- Numerical approximation of the gradient of $J_o(U(k))$ is impractical and also time-consuming since it requires evaluations of $J_o(U(k))$ at different points.
- Values of $J_o(U(k))$ are noisy due to numerical calculation of heat radiation which can be slightly different from one simulation to another.

In application for considered reheating furnace, a highly accurate solution is not practically desired because of high cost simulation. On the other hand, it may be impossible because of uncertainties and errors in the model or data. Actually, what we want is improvement of reheating furnace rather than optimization. Therefore, the crucial issue is finding a better answer quickly for each iteration; asymptotic convergence properties are not relevant.

Among the presented direct search methods, except simplex search methods, the convergence property of algorithms is assured. However, while simplex search methods only need no more than $n+1$ evaluations of objective function to complete the search for improvement of an iterate, the other methods require from $2n$ to 2^n evaluations to do the same thing [LTT00a].

Consider the requirement of the application and characteristic of the available optimization algorithms, we choose Nelder-Mead simplex method to solve the minimization problem in the MPC strategy for level 2 of the reheating furnace. For that, we expect to obtain good control input sequence in a limited number of function evaluations.

The design of Nelder-Mead simplex method for reheating furnace case consists of determining the initial simplex and executing a series of reflection, expansion, contraction, shrink operation. For sample time k, the initial simplex can be constructed from the solution of time $k-1$, i.e., $U_0(k) = U(k-1)$. As in [LLR04], a simplex of size a can be initialized at P_0 based on the following rule

$$U_i = U_0 + \rho e_i + \sum_{j=1, j\neq i}^{n} (q e_j) \text{ for } i = 1, .., n, \tag{6.61}$$

where e_i is a unit base vector and ρ, q determined by

$$\begin{aligned} \rho &= \frac{a}{n\sqrt{2}} \left(\sqrt{n+1} + n - 1\right) \\ q &= \frac{a}{n\sqrt{2}} \left(\sqrt{n+1} - 1\right). \end{aligned} \tag{6.62}$$

The size of the simplex a is taken between 2% and 10% of the smallest domain dimension. This latter parameter is in fact determined by the difference between upper and lower limits of control inputs. For the considered furnace, the differences between upper and lower limits of zone temperature set-points are $90°\text{C}, 155°\text{C}, 175°\text{C}$ and $270°\text{C}$ for zone soaking, heating 2, heating 1, and preheating, respectively. As a result, we take an average value $a = 10°\text{C}$. Different operations are defined with the coefficients of reflection, expansion, contraction and shrink. In general, reflection coefficient has value 1, expansion coefficient takes value 2, contraction and shrink operations have coefficient of $1/2$. Specific steps of the algorithm is given in the following

- Step 1-Initialization of simplex
  The initial simplex in $n - dimensional$ space is defined by $(n+1)$ points $\{U_0, U_1, ..., U_n\}$ which are calculated in Equations 6.62. The objective function is evaluated at each point of the simplex: $y_i = J_o(U_i)$ for $i = \overline{0, n}$. $y_h = J_o(U_h)$ and $y_l = J_o(U_l)$ represent respectively highest and lowest value of cost function. $\overline{U}$ is centroid point of the simplex, i.e., $\overline{U} = 1/n \sum_{j=0}^{n} U_j$.

- Step 2- Reflection operation
  The highest point U_h is reflected through the centroid point $\overline{U}$ to a point U^* which is determined as

$$U^* = (1+\alpha)\overline{U} - \alpha U_h \tag{6.63}$$

  where $\alpha = 1$ is the reflection coefficient. If the function value at the new point is lower than the current highest point, i.e. $U_l < y^* = J_o(U^*) < U_h$ then U_h is replaced by U^* and the algorithm restarts with new simplex. If $y^* \leq y_l$, go to step 3, if $y^* \geq y_h$, go to step 4.

- Step 3- Expansion operation
  U^* is expanded to U^{**} by a relation

$$U^{**} = \gamma U^* + (1-\gamma)\overline{U} \tag{6.64}$$

  where the expansion coefficient $\gamma = 2$. If $y^{**} < y_l$, U_h is replaced by U^{**} and the algorithm restarts. If $y^{**} > y_l$, U_h is replaced by U^* before restarting.

- Step 4- Contraction operation
  U_h is replaced by U^*. Contracted point U^{**} is determined by

$$U^{**} = \beta U_h + (1-\beta)\overline{U}. \tag{6.65}$$

  Where β is contraction coefficient. If $y^{**} \leq y_h$ then P_h is replaced by U^{**} and the algorithm restarts. On the contrary, if the highest point U_h cannot be replaced with any of reflection, expansion, or contraction operations. Shrink operation is executed in order to generate a new simplex, i.e.

$$U_i = \vartheta[U_i + U_l] \text{ for } i = 1, n \tag{6.66}$$

  where shrink coefficient $\vartheta = 1/2$. If the convergence tolerance is satisfied then stop, otherwise go to step 1 and the algorithm restarts

The convergence criterion used to halt the procedure is calculated as $\sqrt{1/(n+1)\sum_{i=1}^{n}(y_i - \overline{y})^2}$. That is, the algorithm terminates the iteration when the simplex become sufficiently small.

6.4 Numerical result

In order to validate the adopted MPC strategy on the considered reheating furnace, we simulate the furnace operation by constructing a furnace simulator from the non-steady-state model presented in Chapter 3. The data used to validate the simulator is provided from the database of the considered furnace. From the scheduling table, the information of the heated slabs, fuel composition, etc., and the zone power set-points calculated by the level 1 controllers, the furnace simulator calculates instantaneous temperature profile of slabs, exact position of each slab, the zone temperatures, and also the final slab temperature profiles as depicted in Figure 6.26. At each sample time, the set-points optimizer calculates the optimal zone temperature set-points based on the current position, temperature of slabs, and the current zone temperatures. In addition, to compare MPC strategy and the existing

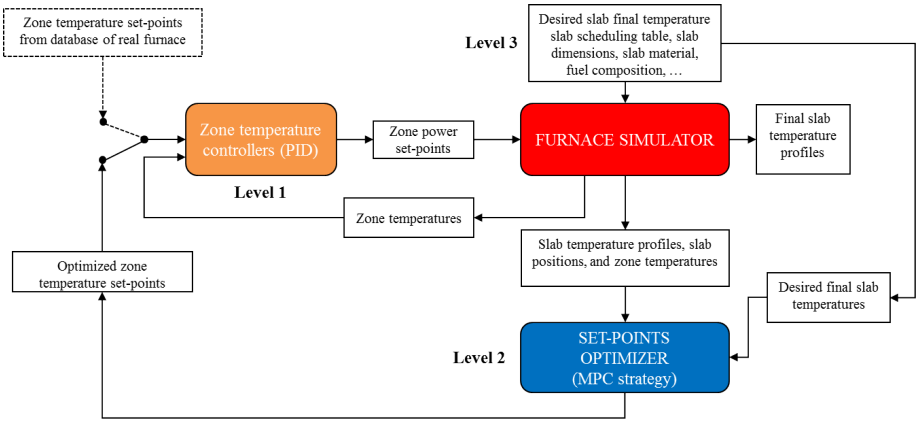

Figure 6.26: Software-based simulation environment

control method of level 2, instead of using directly final slab temperature profiles of the furnace database, we use the values calculated by the simulator provided with the zone temperature set-points from furnace database. By doing so, we can avoid the error of simulator (comparing to the real furnace) in the comparison between MPC strategy and the existing method.

This section presents firstly the simulation environment, operational scenario of the furnace, and the validation of the simulator against the industrial data. Specific tuned parameters of MPC strategy will then be given. Consequently, the results of MPC strategy will be analyzed and compared to the results of the existing method.

6.4.1 Simulation conditions

The whole simulation is coded in Intel Fortran 90 to facilitate the programming task because the non-steady-state model is originally developed in Fortran 90. The structure of simulation program is outlined in Figure 6.26. One may ask why the level 1 controllers are simulated as PID instead of distributed MPC which has been studied and implemented. It is because the available data used in the simulation was recuperated when the furnace was still running with PID controllers. In addition, from the point of view of level 2, control of level 1 is considered transparent. Thus, the choice of controller for level 1 will not affect controller level 2. The simulation

environment captures the whole cascaded control system described in Section 6.1 (cf. Figure 6.2). The main difference is that the observer is not required because all the system variables are available from the furnace simulator itself.

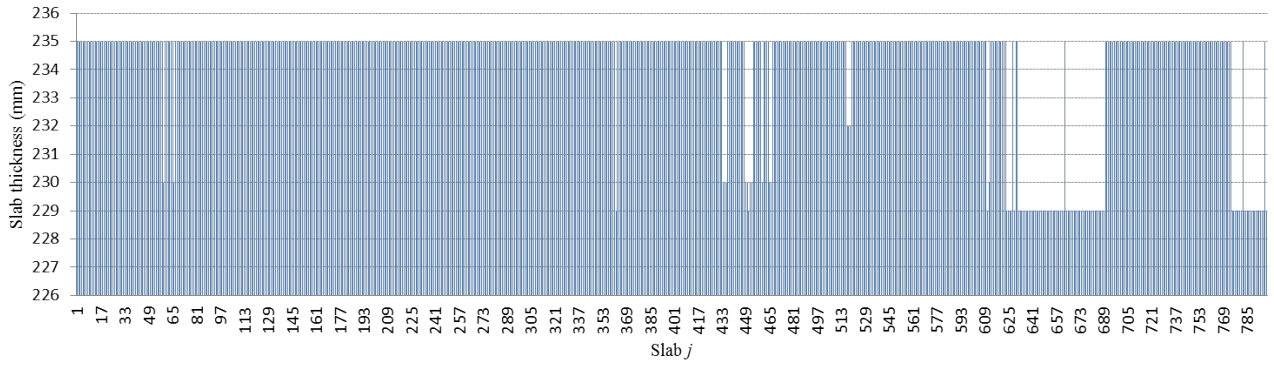

Figure 6.27: Thickness of slabs

The simulation environment has been installed on a standard PC with a 2.5GHz CPU and 3 GB of RAM. On this machine, to simulate 1 h of furnace operation the non-steady-state model needs 5 s. This is considerably more time-consuming than steady-state mode which is around 10^5 times faster. We have chosen the prediction horizon of Δt_P=2 h and sample time Δt_s=5 min according to the average reheating time of slabs and time interval of loading and discharging events. Thus, predictions of slab temperatures over the prediction horizon will need 10 s. On the other hand, we want the simulation environment to be implementable for real-time operation. Therefore, the optimization of MPC controller has to deliver the optimized zone temperature set-points in less than a period of one sample. As a result, the maximum number of function evaluations we have is $l_{max} = \Delta t_s \times 60/10 = 30$. The heating scenario uses database of three-day furnace operation in which around 900 slabs are heated and

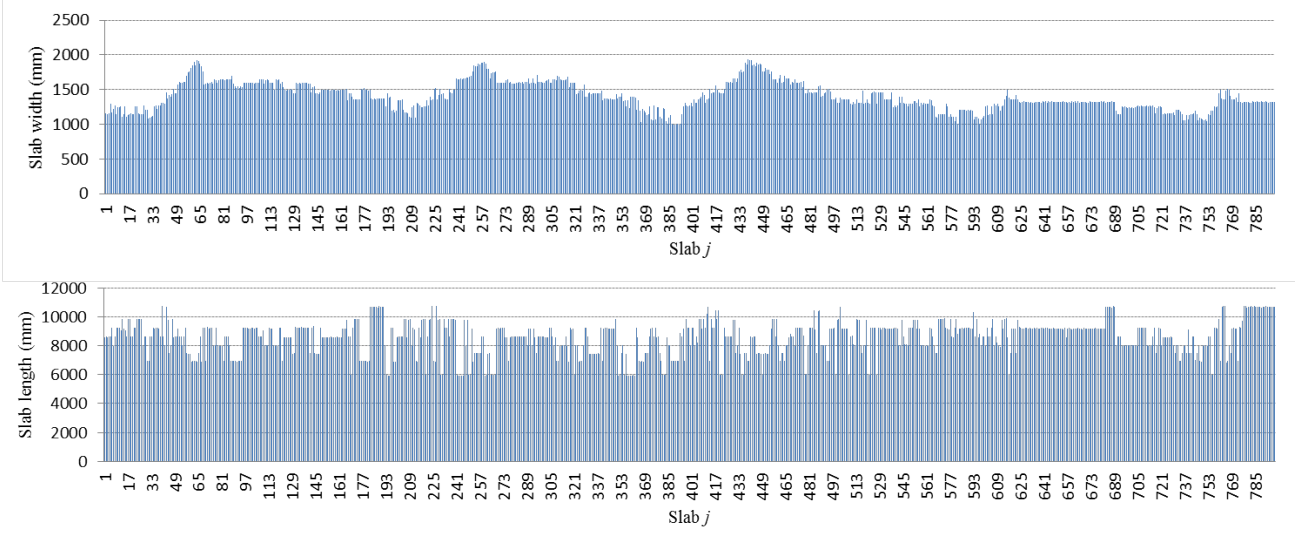

Figure 6.28: Width and length of slabs

withdrawn from the furnace. The data has been chosen to capture transient operational situations such as large

variations of desired final slab temperatures, varying slab dimensions, different reheating times, production short halt of 30 min to long stop of 2 h.

Figure 6.27 shows the thicknesses D_j of the slabs. The thicknesses vary between 229 mm and 235 mm and most common value is 235 mm, this variation is not considerable because, as mentioned previously, the considered furnace is dedicated to the heating of slab with thickness of 235 mm. The width and length of the slabs are spread over the range of 0.99 m to 1.93 m and 5.90 m to 10.75 m, respectively as shown in Figure 6.28. Despite highly regular thicknesses, the slab width and length vary greatly. Thus, the transient effects on the heating process are unavoidable.

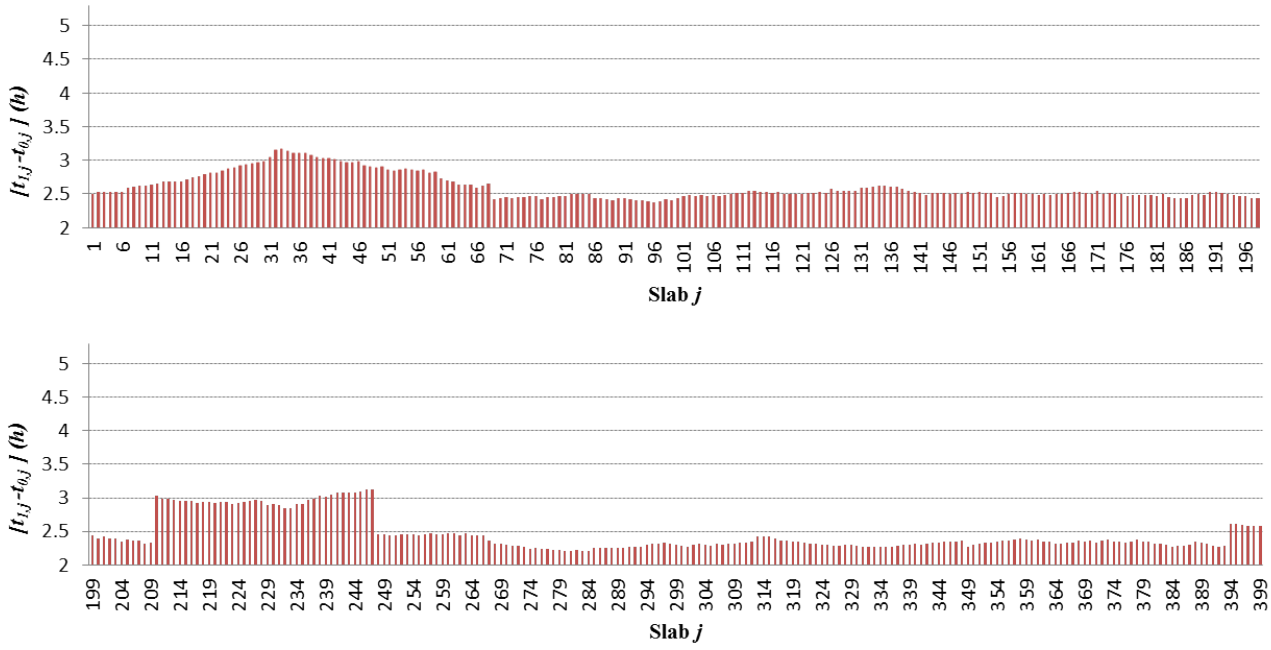

Figure 6.29: Residence time of slabs 1 to 399

The reheating times of slabs $t_{j,1} - t_{j,0}$, or in other word, the residence times of the slabs inside the furnace are given in Figures 6.29 and 6.30. The average reheating time is around 2.5 h. Besides, we can see slabs such as the slabs $j = 210$ to $j = 247$ of which residence times are increased to around 3 h. This is because the furnace has been stopped for 30 mn when it contained slabs $j = 210$ to $j = 247$. In a longer stop, reheating time can be dramatically increased to 4 h and 5 h as the case of slabs $j = 600$ to $j = 641$. These long residence times may lead to overheating and unwanted damage of the material quality.

The usual position-time diagram of representative slabs are shown in Figure 6.31. The lines represents the position of each slab in function of time. The position of a slab is the distance from its center to the entry of the furnace. Due to the long stop at time 46.3 h, the slabs $j = 600$ to $j = 606$ remained at the same position for about 2 h.

For the considered reheating scenario, the temperature set-point of zone preheating, heating 1, heating 2 and soaking are limited respectively between $\tilde{T}_{z,min}$=[830, 1085, 1200, 1210]°C and $\tilde{T}_{z,max}$=[1100 , 1250, 1355, 1300]

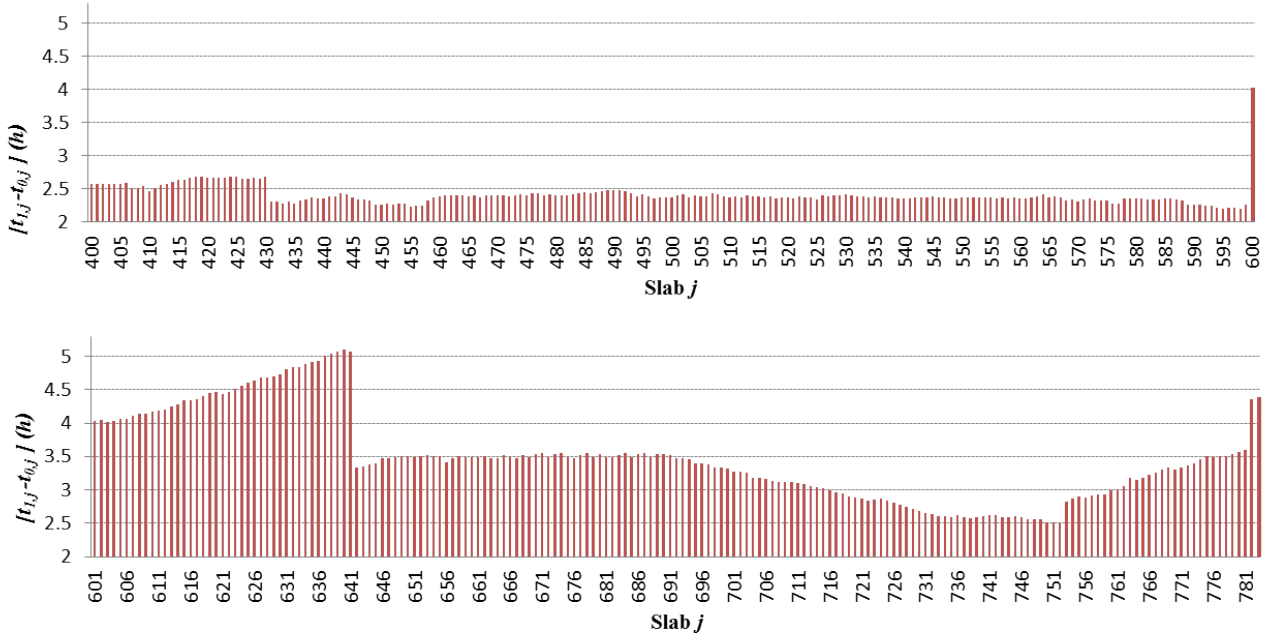

Figure 6.30: Residence time of slabs 400 to 783

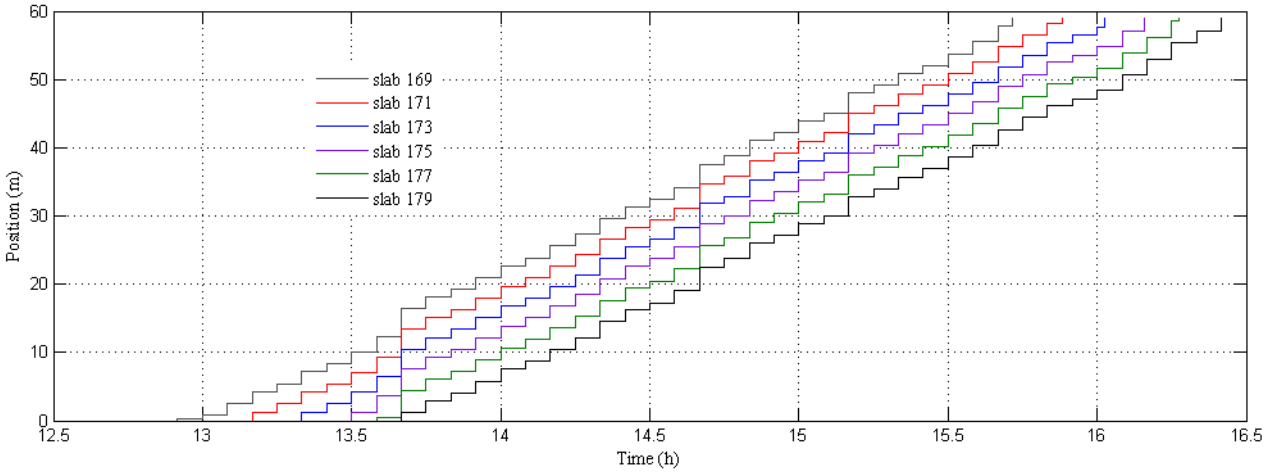

Figure 6.31: Position-time diagram of slabs 179 to 189

°C. The rate of change of zone temperature set-points are also bounded with upper value $\dot{\tilde{T}}_{z,max}$=9°C$/min$ and lower value $\dot{\tilde{T}}_{z,min}$=$-$4.5°C/min.

The desired final slab temperatures are presented in Figures 6.33 and 6.34. For all the slabs, the desired temperature profile $\tilde{T}_{j,end}(y_j)$ is homogeneous with respect to y_j. And, the bounds on the final temperature profile are determined by $T_{j,end,max} = \tilde{T}_{j,end} + 20$°C and $T_{j,end,min} = \tilde{T}_{j,end} - 20$°C. In the Figures 6.33 and 6.34, these bounds are indicated by two limits of the grey bars. In practice, these bounds could be relaxed according to operating scenario, $T_{j,end,max}$ in particular because over-heating is less restricted than under-heating which causes rejections of slabs in hot rolling mills. Generally, we can see that desired temperatures of slabs are highly

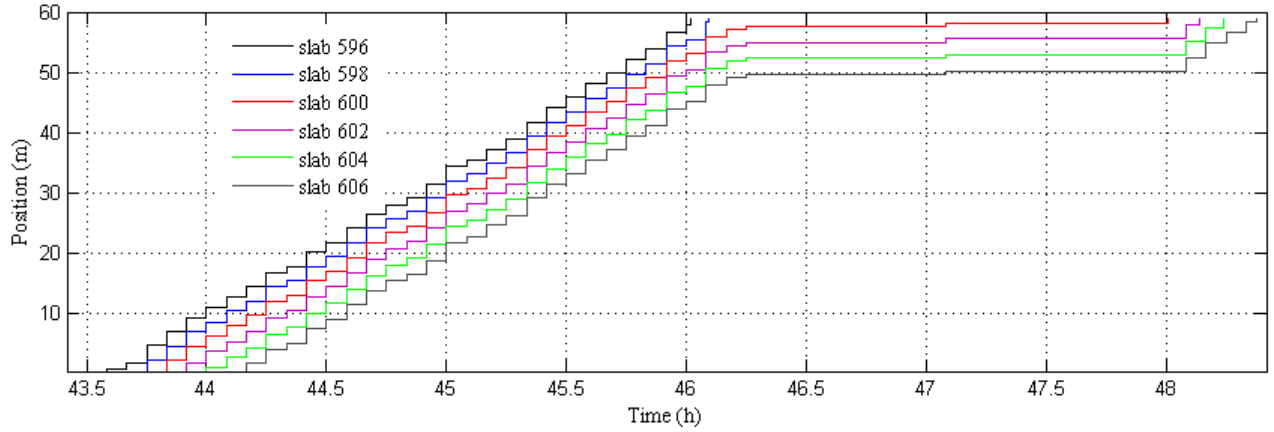

Figure 6.32: Position-time diagram of slabs 506 to 606

irregular with exception of slabs $j = 618$ to $j = 687$. The irregularity between desired final slab temperatures will make the control task more difficult to achieve high temperature accuracy. Temperature range is from 1170°C, as of slabs $j = 30, l = 31$, to 1230°C as of slabs $j = 57, j = 58$. Due to slow thermal inertia of the furnace, it would be desirable that consecutive slabs have as much overlap between the allowed final temperature ranges $[T_{j,end,min}, T_{j,end,max}]$ as possible. We see that most of slabs have some required temperature overlaps with their neighboring ones, with exception of some slabs like $j = 31$ and $j = 32$ or slab $j = 295$. In fact, the arrangement of slabs according to their desired final temperature is one of important task of level 3, which, however, is not in the scope of this dissertation. The upper bound of slab temperature is $T_{j,max} = 1350$°C, and the temperature gradient of slabs should remain below $\Delta T_{j,max} = 350$°C. Maximum inhomogeneity of final slab temperature profile $\Delta_{j,max,end}$ is dependent of slab thickness. Since the considered furnace heats most slabs that are 235 mm thick, $\Delta_{j,max,end}$ is fixed at value 100°C. The Nelder-Mead simplex algorithm is used as optimization method. For initialization of simplex algorithm, we use a simplex with size $a = 10$°C. The initial control inputs U_0 of the simulation are imposed with the averaged values of upper and lower limits, i.e., $U_0 = (\tilde{T}_{z,max} + \tilde{T}_{z,min})/2$=[965, 1167.5, 1277.5, 1255]°C. Prediction horizon $\Delta t_P = 120$ min which represents average residence time of a slab inside burner zones. The sampling period is $\Delta t_s = 5$ min which represents the average discharging time (or loading time) interval between 2 successive slabs. The control horizon is $\Delta t_C = 5$ min, which means that $N_C = 1$. About the parameter tuning of the objective function, weighting factors on error of final slab temperatures and final slab temperature gradients are $\alpha_1 = 70$, $\alpha_2 = 10$. Others weighting factors concerning energy consumption are $\alpha_2 = \alpha_3 = \alpha_5 = 10$. Weighting factor on penalty terms μ is imposed with great value, here, $\mu = 1000$.

For the simulation, all slabs have initial temperature close to ambient temperature which is fixed at 20°C. At time t=0 h, the furnace is assumed to have inside it 31 (average number) fictive slabs with standard dimensions of (700 x1860 x235) mm and standard time interval between discharging events of 5 min. These fictive slabs are given with a certain temperature profile according to its position inside the furnace. The slabs of real system are subsequently entered the simulator for the rest of the simulation.

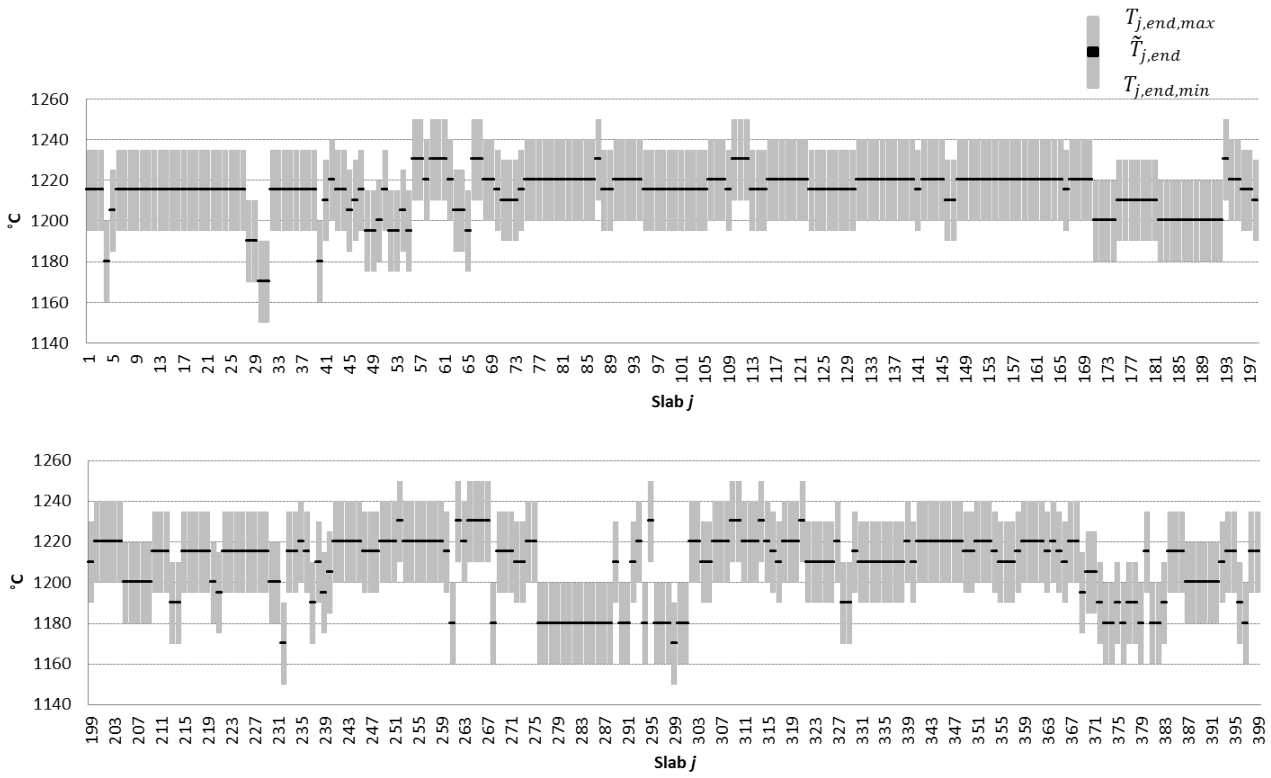

Figure 6.33: Desired final temperature of slabs 1 to 399

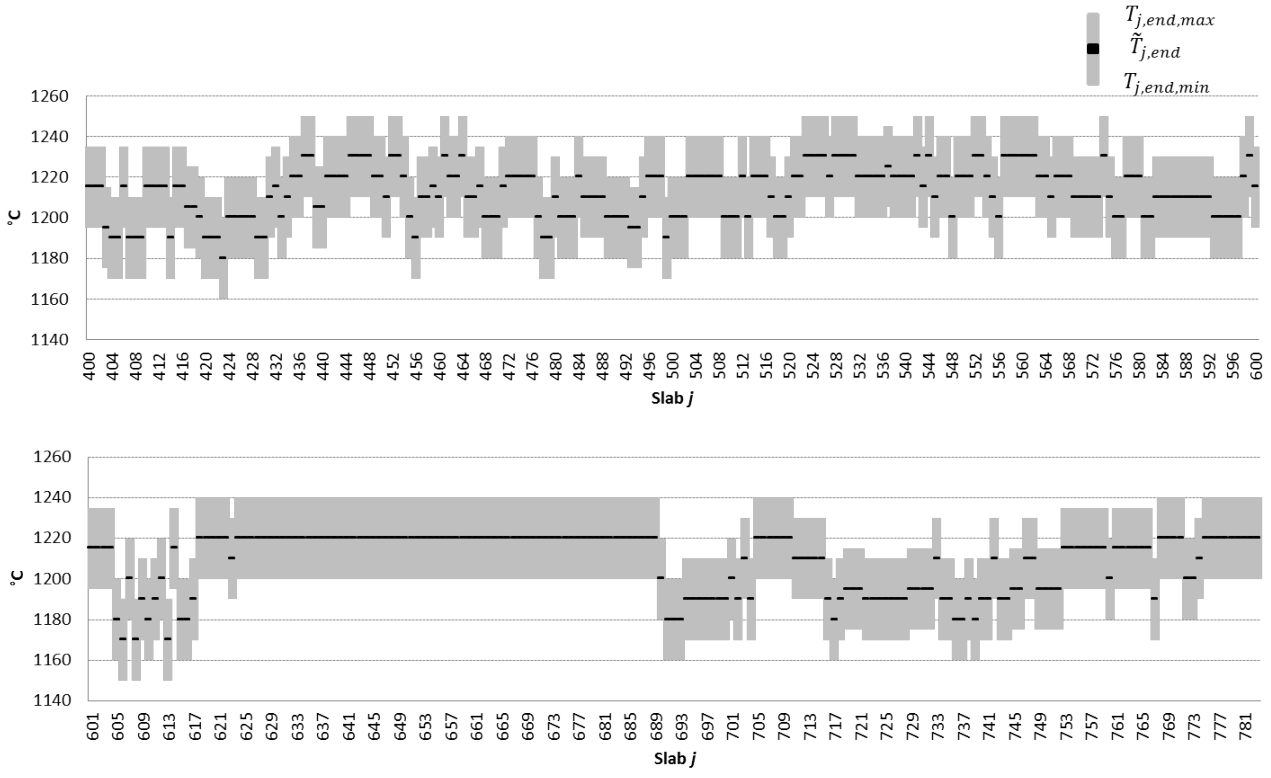

Figure 6.34: Desired final temperature of slabs 400 to 783

Before integrating the MPC controller in the simulation, we have to validate the simulator with the given operating scenario of the furnace. The data of zone temperature set-points calculated with existing controllers is applied to the simulator. Then, the obtained results are compared with data of the real system. The averaged final temperature of slabs are shown in Figure 6.35. We see that the simulator have successfully reproduced the final temperature of slabs. The histogram of temperature errors are given in Figure 6.36. The average error is 1.5 °C with a standard deviation of 9 °C.

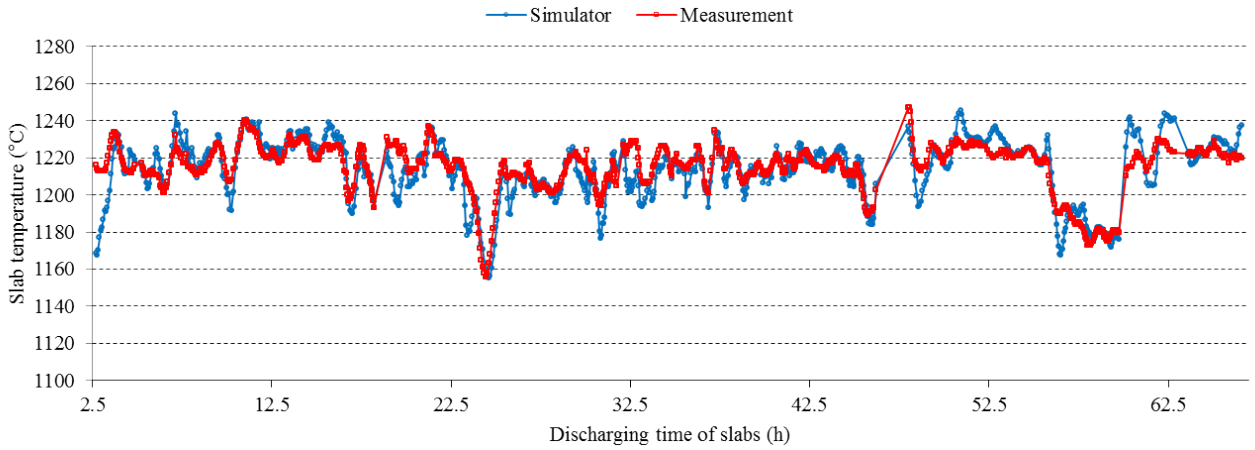

Figure 6.35: Averaged final slab temperatures

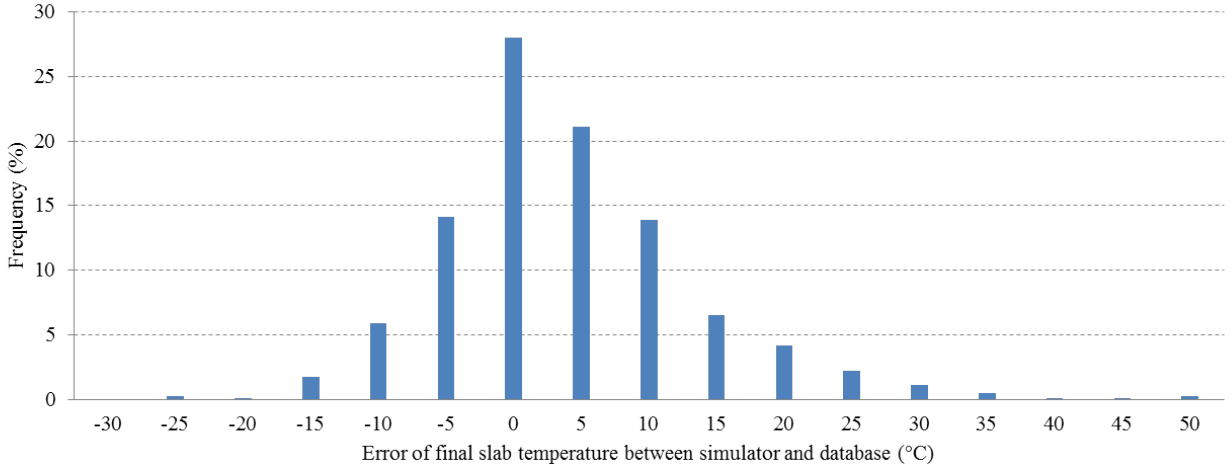

Figure 6.36: Averaged final slab temperatures

On the other hand, we should also look at the consumed power calculated by the simulator in comparison to the measurement of the real system. In Figure 6.37, we can see that the simulator has been able to reproduce the dynamic of the measurement although some static errors are present. As given in Table 6.5, the average consumed power calculated by the simulator is 97.5 MW which is close to average measurement value of 99 MW.

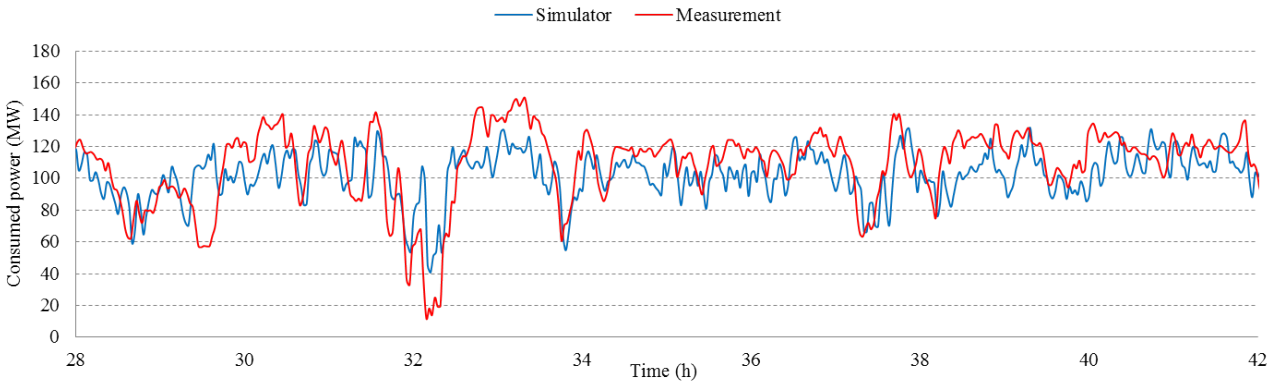

Figure 6.37: Consumed power of the furnace

	Average consumed power (MW)				
	Soaking	Heating 2	Heating 1	Preheating	Total
Simulator	10	28.91	42.27	16.35	97.53
Measurement	10.97	28.06	43.12	16.89	99.04

Table 6.5: Average consumed power of the furnace

6.4.2 Simulation results

Zone temperature set-points calculated by the MPC controller are shown in Figures 6.38, 6.39, 6.40 and 6.41 in comparison with values designed by the existing controller for the time interval [0 h, 64 h]. The dashed lines show the upper and lower bounds of the temperature set-points. The red solid curve shows the set-points calculated by MPC controller, whereas the blue solid curve shows the set-points given by the existing controller. The black solid curve is the simulated zone temperatures following set-points of MPC controller given by the PID controllers of level 1. We see generally the set-points optimized by MPC controller respect their bounds.

The temperature of the soaking zone is given in Figure 6.38. We see that set-points calculated by MPC are lower than those given by the existing controller. The PID controllers are, in most of time, able to follow set-points provided by MPC controllers. Due to upper limits of zone powers PID may sometimes fail to attain the set-points like at the intervals [2 h, 3 h] and [25 h, 26 h].

For zone heating 2, temperature set-points calculated by MPC are generally higher than values given by the existing controller as shown in Figure 6.39. The level 1 controllers succeed to follow the temperature set-points. In zone heating 1, the set-points given by MPC are mostly lower than that of the existing controller, especially for the time intervals [2 h, 5 h] and [50 h, 60 h]. The PID controllers have no problem tracking set-points provided by MPC controllers.

Unlike other zones, we see clearly that PID controllers of preheating zone fail to follow the set-points given by the MPC controller which are a lot lower than values calculated by the existing controller. In fact, the temperature of this zone is not only affected by its fuel flow which has lower bound greater than zero, but also temperature of other zones through the flow of exhaust gas. Therefore, low temperature set-points are not followed by PID

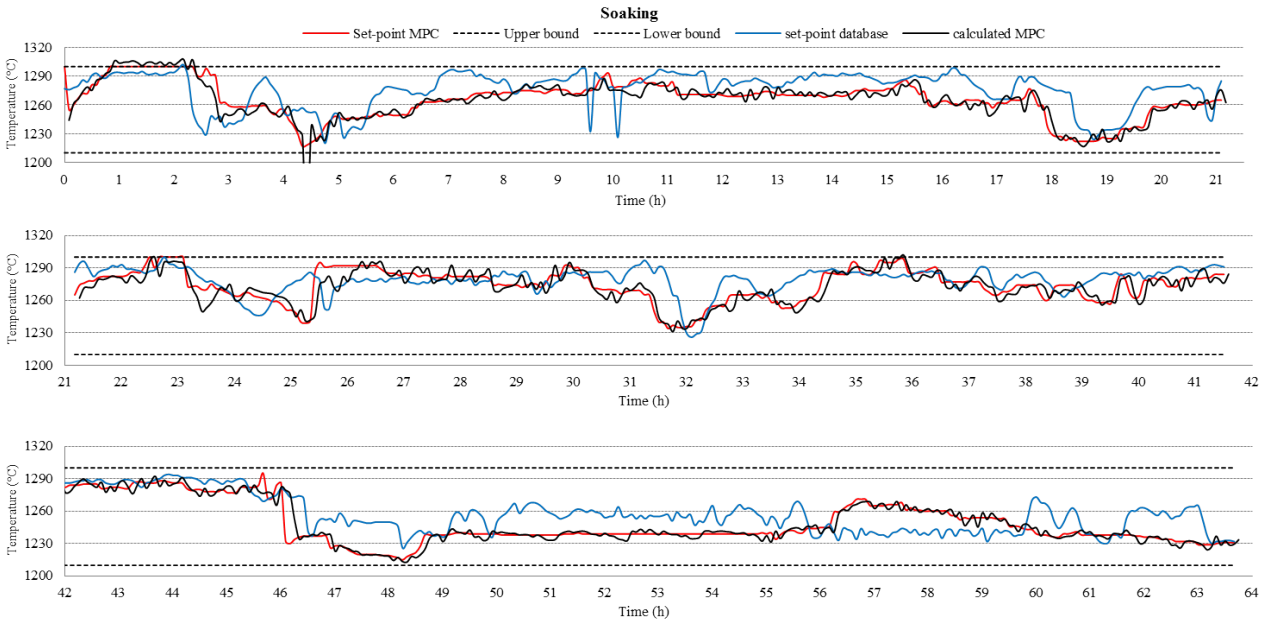

Figure 6.38: Temperature set-point of soaking zone calculated by MPC controller compared to that of the existing controller

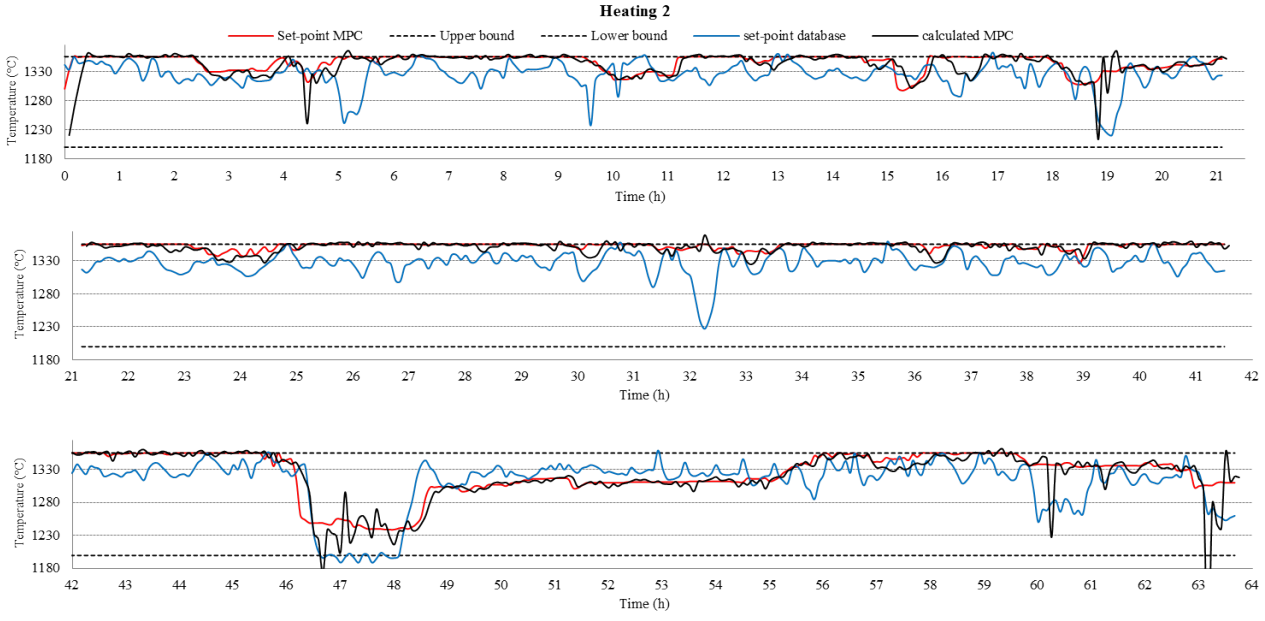

Figure 6.39: Temperature set-point of zone heating 2 calculated by MPC controller compared to that of the existing controller

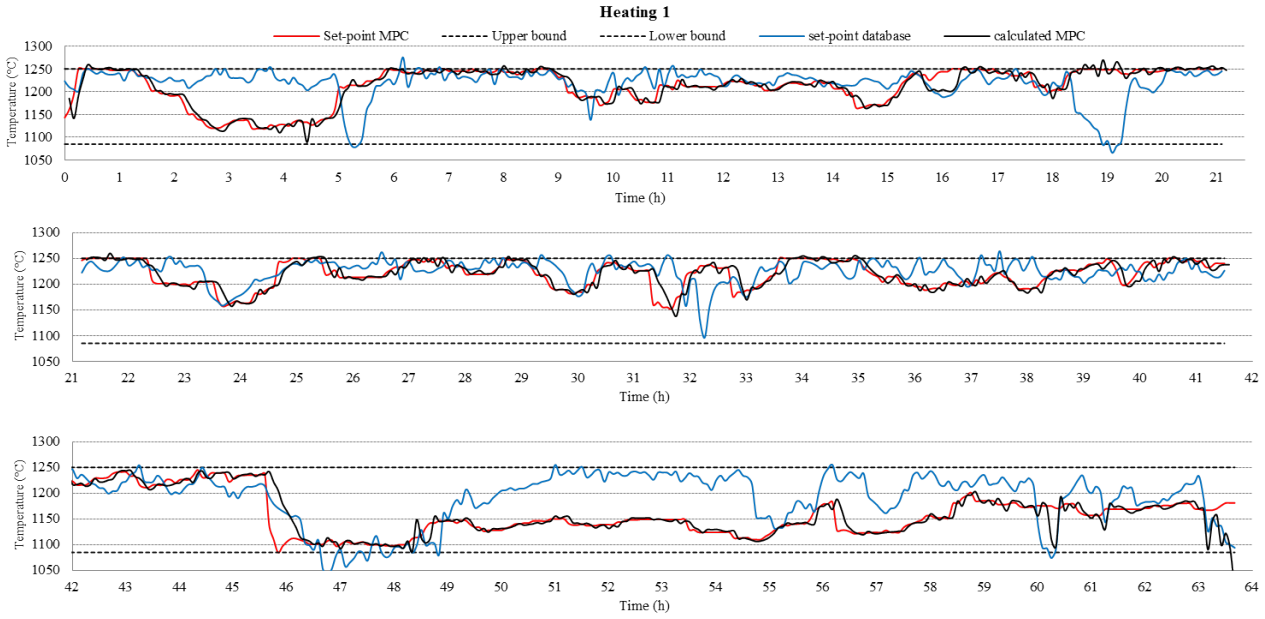

Figure 6.40: Temperature set-point of zone heating 1 calculated by MPC controller compared to that of the existing controller

controllers. The optimization in MPC strategy has provided low set-points because we use fixed lower constraint (810°C) while it is actually the time-varying black solid curve.

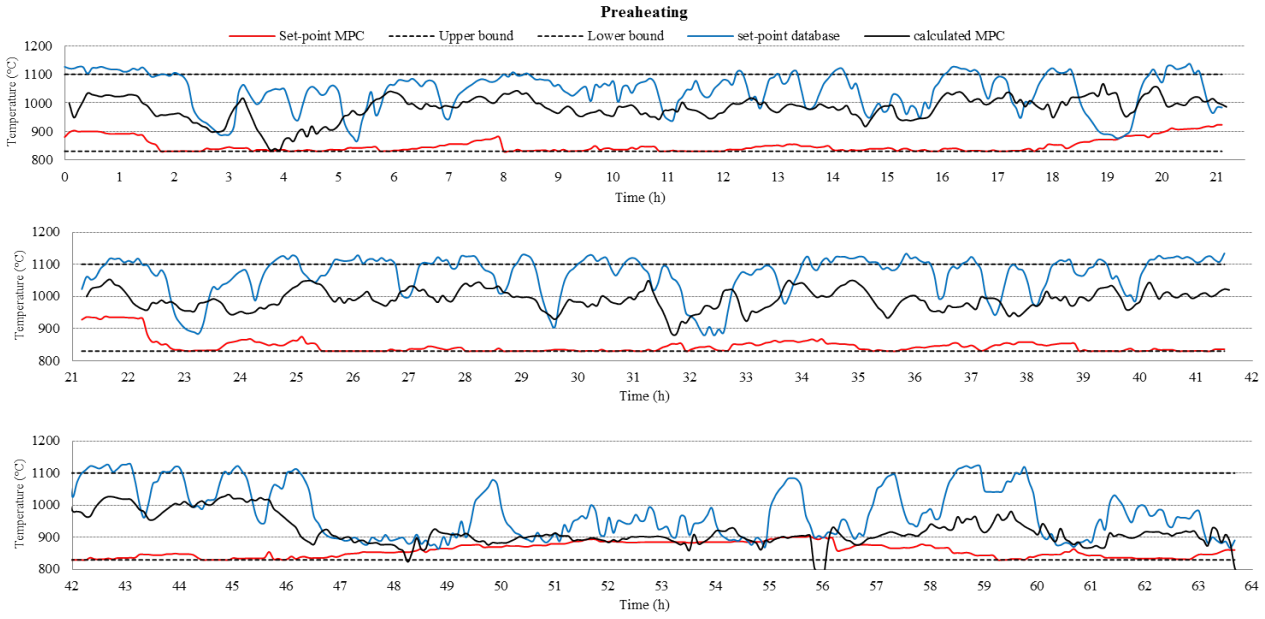

Figure 6.41: Temperature set-point of zone preheating calculated by MPC controller compared to that of the existing controller

The average zone temperature set-points are given in Table 6.6. In average, the MPC controller has redistributed the zone temperatures towards the exit of the furnace, especially at the zone heating 2. By increasing the average temperature set-points of zone heating 2, MPC controller is able to decrease that of other zones, especially in the front zones: preheating and heating 1. Now, we should be interested in the heating results of MPC controller.

	Average zone temperature set-points (°C)			
	Soaking	Heating 2	Heating 1	Preheating
MPC controller	1262	1340	1196	851
Existing controller	1270	1321	1211	1028

Table 6.6: Average zone temperature set-points

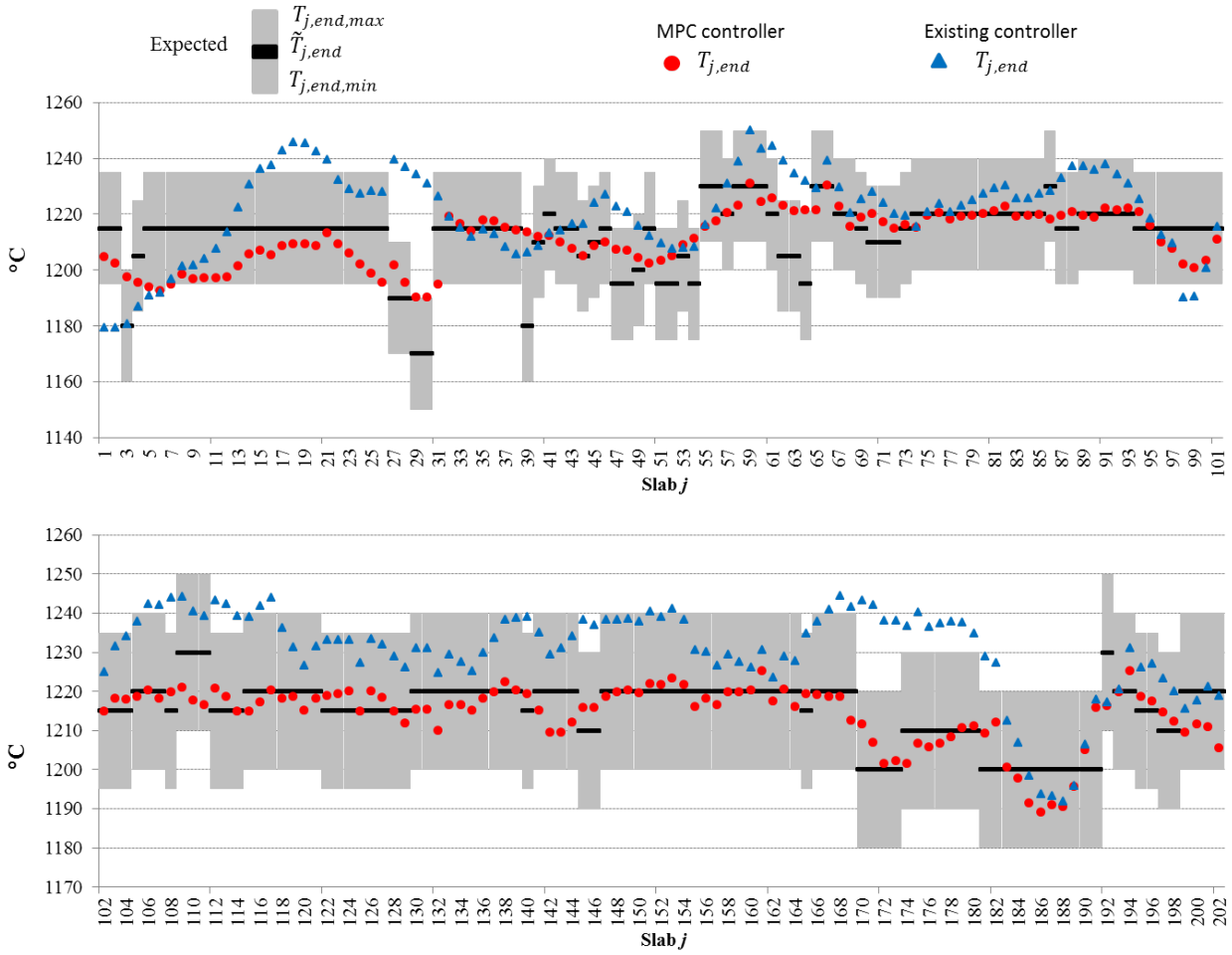

Figure 6.42: Final temperature of slabs 1 to 202

The average final slab temperatures $mean_{y_j} T_{j,end}(y_j)$ are shown in Figures 6.42, 6.43, 6.44 and 6.45. The final slab temperatures simulated with MPC controller are represented by the red dotted curve and the blue one represents the temperatures simulated with the exiting controller. Most of the temperatures provided by MPC controllers stay inside of the expected bounds with exceptions of slabs of which desired final temperatures are particularly different with those of neighboring slabs such as slabs $j = 269$ to $j = 275$ or slabs $j = 384$ to $j = 386$ in Figure 6.43.

On the other hand, existing controller often overheats the slabs like for the slabs $j = 102$ to $j = 182$ in Figure 6.42 or the slabs $j = 630$ to $j = 695$ in Figure 6.45. This tendency is the preventive strategy adopted in the existing controller to avoid slab under-heating and slab rejections in the hot rolling mill. However, it may become

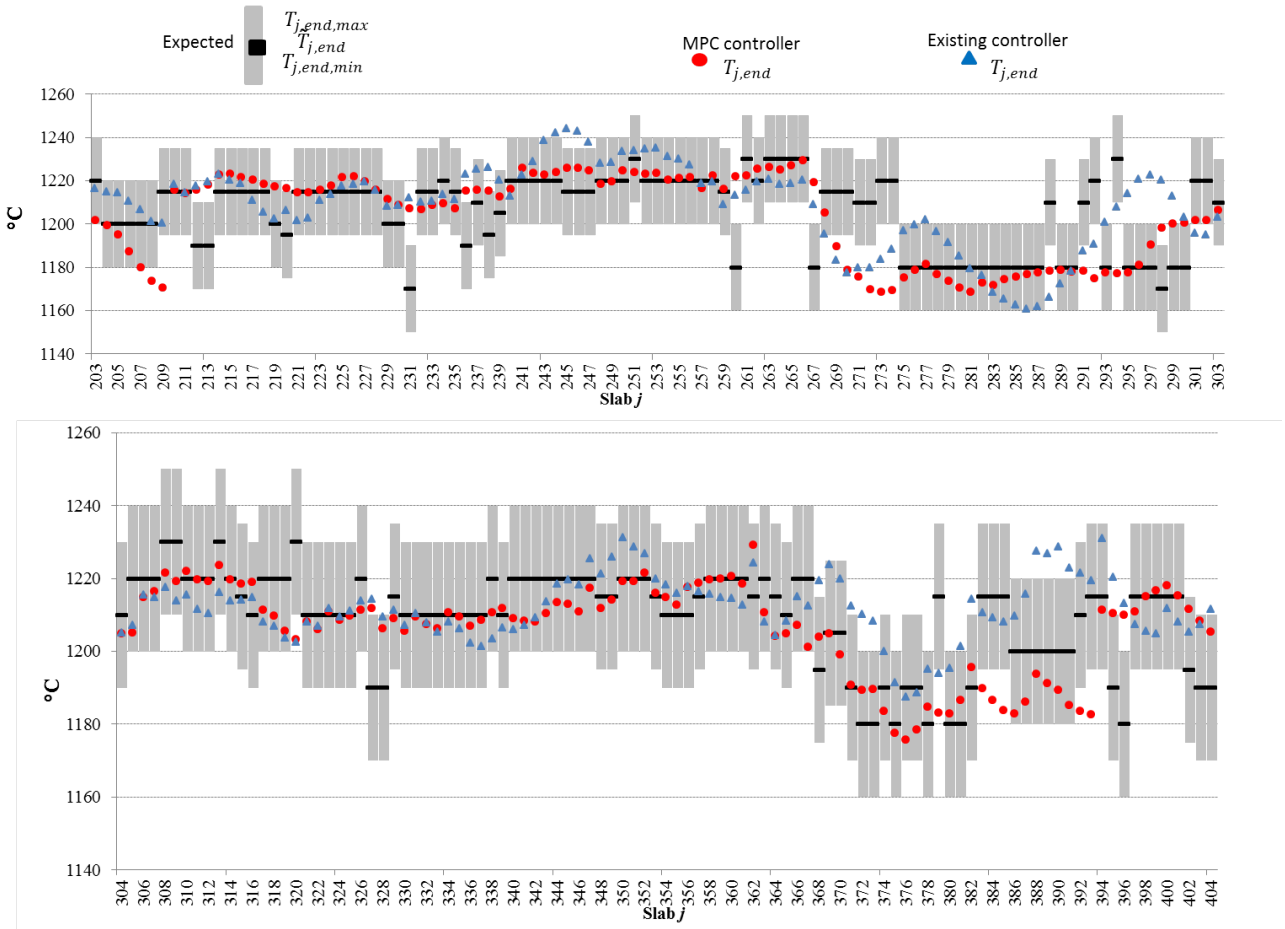

Figure 6.43: Final temperature of slabs 203 to 404

excessive and conservative as for slabs $j = 755$ to $j = 766$ in Figure 6.45, in these cases energy is wasted. MPC controller, on the contrary, gives less slab overheating and less deviations from the desired temperatures, as example of slabs $j = 75$ to $j = 95$ in Figure 6.42 or slabs $j = 752$ to $j = 782$ in Figure 6.45.

The slabs that have high residence time compared to the average due to a long furnace stop, like the slabs $j = 600$ to $j = 700$ when a stop of 2 h occurred (c.f. Figure 6.30), are often largely overheated with the existing controller as shown in Figure 6.45. The MPC controller, in contrast, is able to reduce the overheating caused by long stop.

More specified statistics are given in Figure 6.46, we have the histograms of final slab temperature errors (difference between calculated and desired values) obtained with MPC strategy and with the existing controller. On average, we see that both methods slightly overheated slabs. And the overheating is reduced from 13.7°C with the existing controller to 7.4°C with MPC controller as specified in Table 6.7. This is a crucial improvement which may contribute to the reduction of overall energy consumption of the furnace. In addition, standard deviation of temperature errors are reduced from 10°C to 8°C. This indicates that MPC strategy is more accurate that the existing controller. To measure the success of the heating, we consider the bounds of temperature errors which is the range of -20°C to 20°C. For MPC strategy, 92% of slabs is successfully heated while the existing controller only obtained the results of 75%. More success on final slab temperature leads to less slab rejections at the downstream process, e.g., the hot rolling mills. In conclusion, compared to the existing controller, the MPC

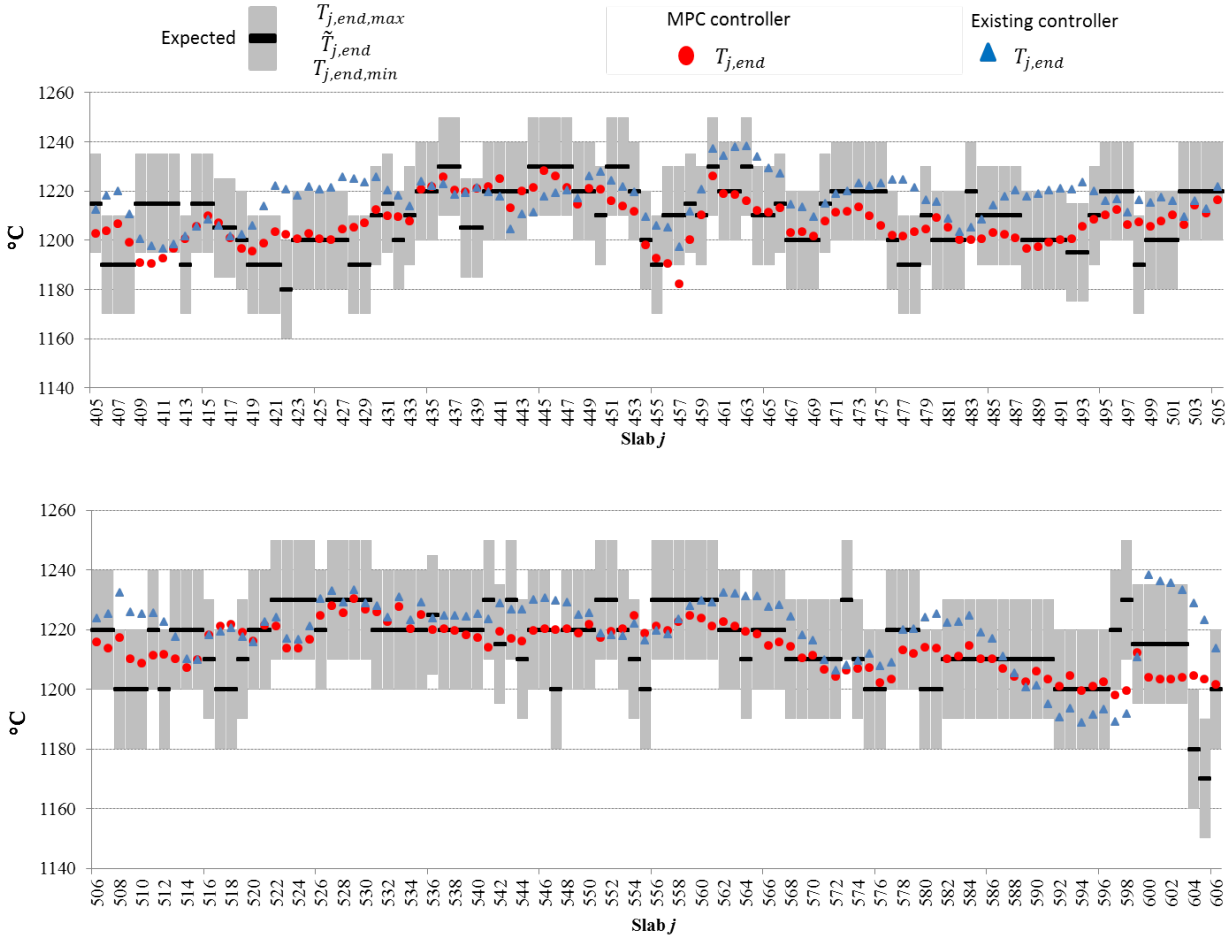

Figure 6.44: Final temperature of slabs 405 to 606

strategy has shown more accurate control over the final slab temperatures, and consequently less slab overheating which may contribute to consumption reduction.

	Average (°C)	standard deviation(°C)	±20°C(%)
MPC	7.4	8	92
Existing	13.7	10	75

Table 6.7: Deviation of final slab temperature from the desired value

Homogeneity of slabs is reflected with the final temperature gradients inside slabs as shown in Figure 6.47. The red curve represented final slab temperature gradients simulated with MPC controller, the blue one are those simulated with the existing controller. In general, MPC controller gives lower temperature gradients with exceptions as of slabs $j = 700$ to $j = 747$. Some slabs have particularly more homogeneous temperature with MPC controller than with existing controller such as slabs $j = 40$ to $j = 95$ and slabs $j = 200$ to $j = 238$. The slabs $j = 600$ to $j = 700$ are affected by a long stop of 2 h, therefore, their final temperature gradients are much smaller than other slabs in the results of both of controllers. The improvement of MPC controller is confirmed

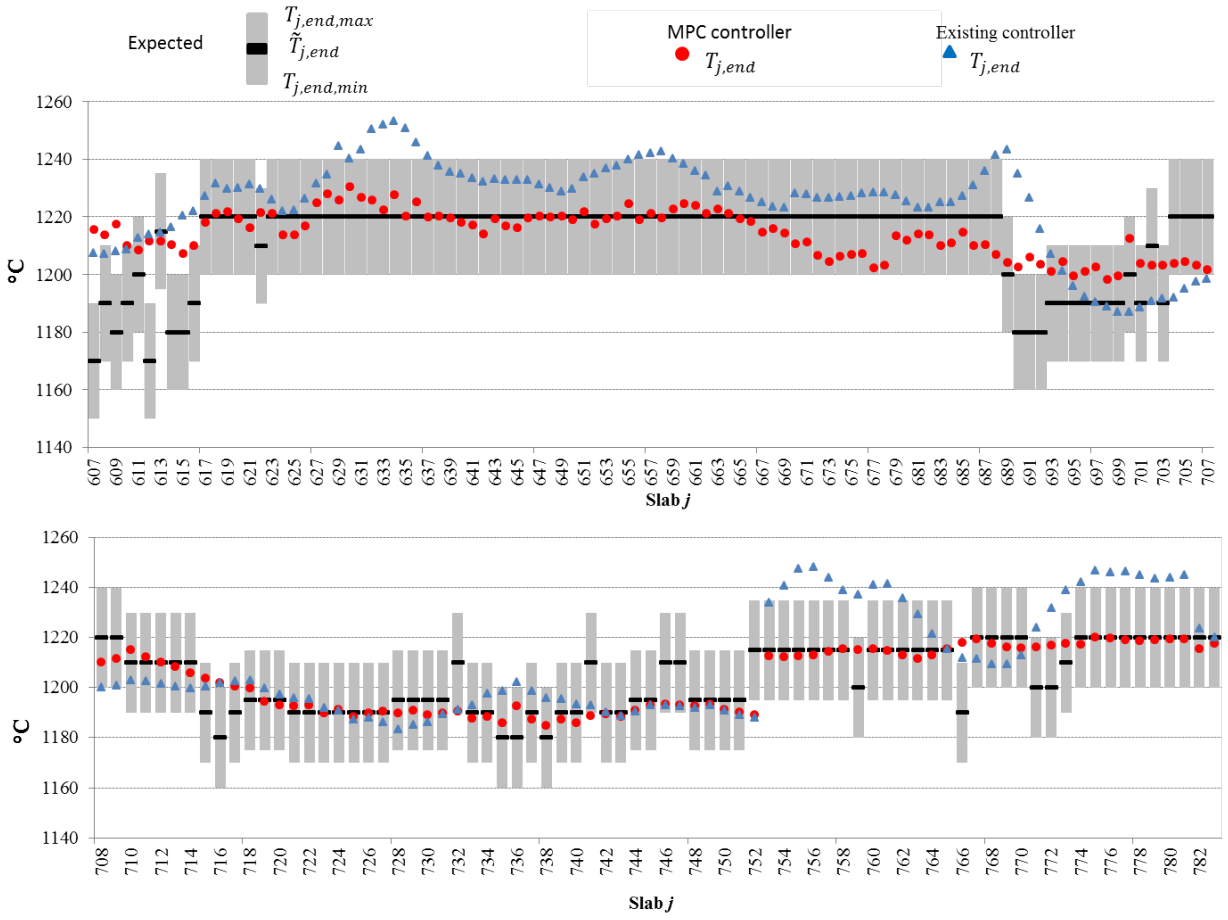

Figure 6.45: Final temperature of slabs 607 to 783

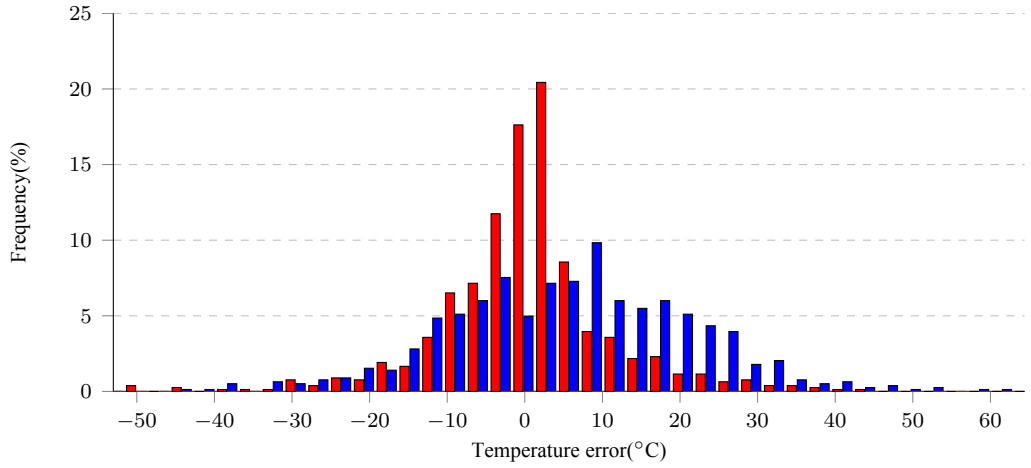

Figure 6.46: Histogram of error of final slab temperature: MPC controller; Existing controller

with Figure 6.48, and Table 6.8. Average gradient is reduced from 73 °C to 68 °C and 3 more percent of slabs is the theoretical upper limit of final temperature gradient which is fixed at 100°C.

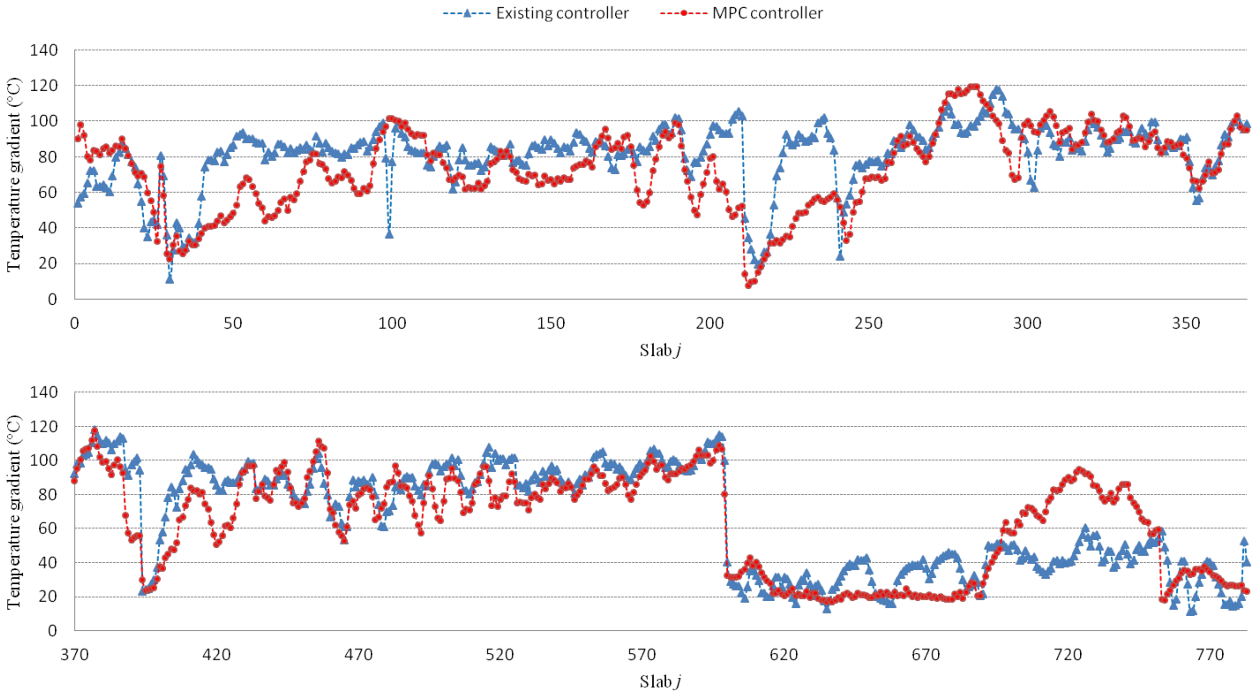

Figure 6.47: Final temperature gradient of slabs

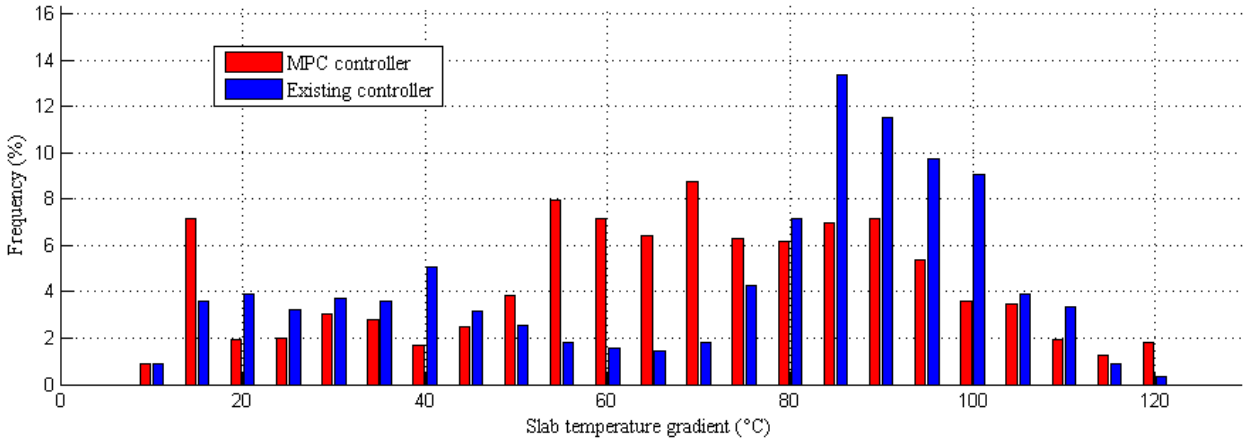

Figure 6.48: Histogram of final slab temperature gradients

	Average (°C)	standard deviation(°C)	< 100°C(%)
MPC	68	26	93
Existing	73	26	90

Table 6.8: Inhomogeneity of final slab temperatures

The temperature evolution of some representative slabs, which have typical residence time of 2.5 h, are shown in Figures 6.49, 6.51, and 6.53. The red solid thick line represents the average temperature of the slab $mean_{y_j} T_j(y_j, t)$

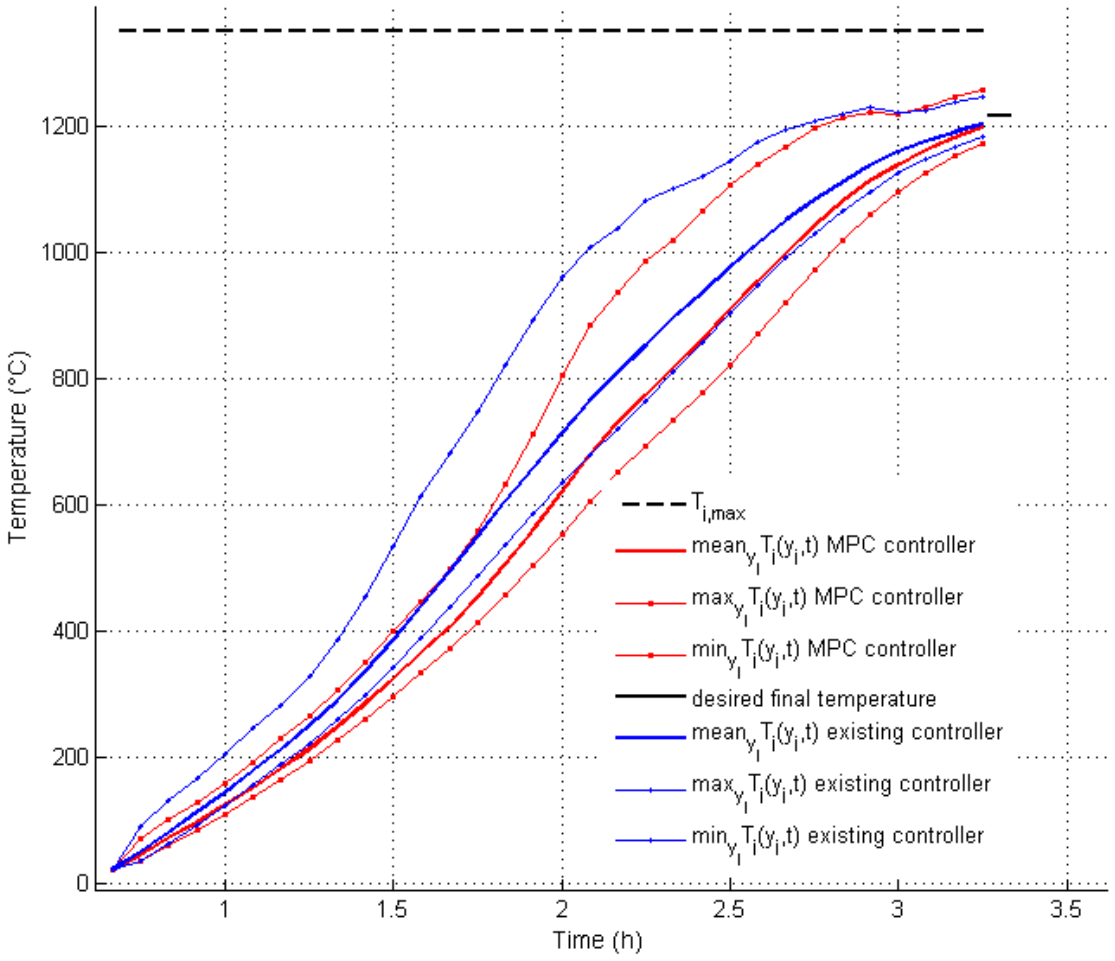

Figure 6.49: Temperature trajectory of slab $j = 9$

simulated with MPC controller, whereas the blue one is that given by existing controller. The corresponding temperature gradients are given in Figures 6.50, 6.52, and 6.54. The thin lines with markers are the maximum and minimum temperature of slabs. Generally, we see that the temperature trajectories respect the upper limit and the final temperature gradients satisfy the required upper bounds. The temperature gradients hardly exceed the upper bound.

The average slab temperatures given by MPC controller are higher than that simulated with the existing controller. Similarly, the maximum temperature of slab given by existing controller is also higher than that simulated with the MPC controller. Temperature gradients of the considered slabs given by existing controller increase rapidly since slabs enter the furnace. This is one of the causes leading to energy inefficiency. On the contrary, with MPC controller, the gradients evolve more slowly compared to those simulated with the existing controller. This tendency will lead to reduction of energy consumption.

For the slabs that are affected by the furnace stops as slabs $j = 600$, $j = 601$ and $j = 641$, their representative

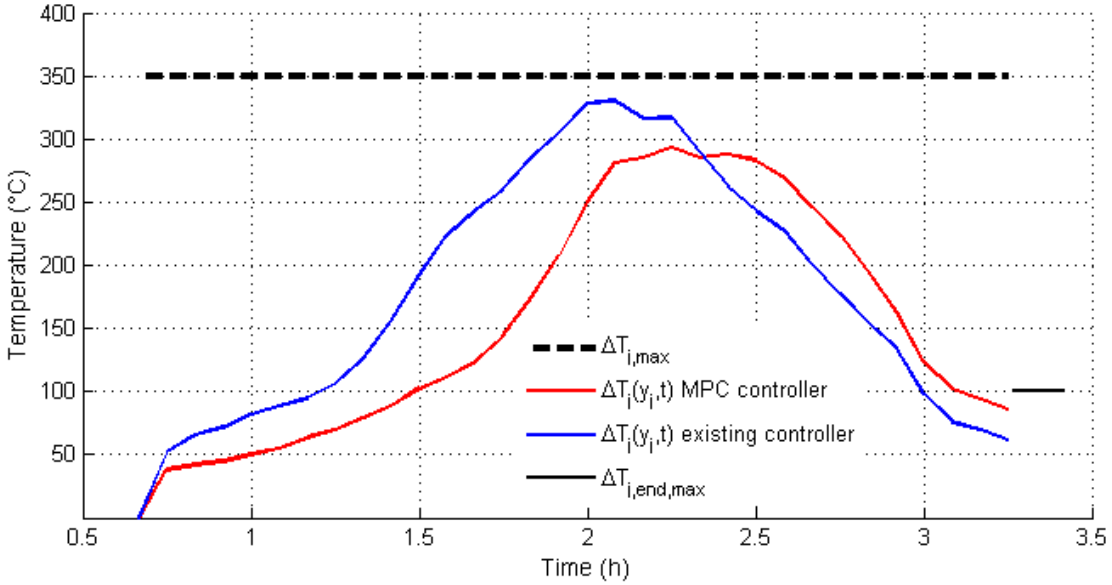

Figure 6.50: Temperature gradient of slab $j = 9$

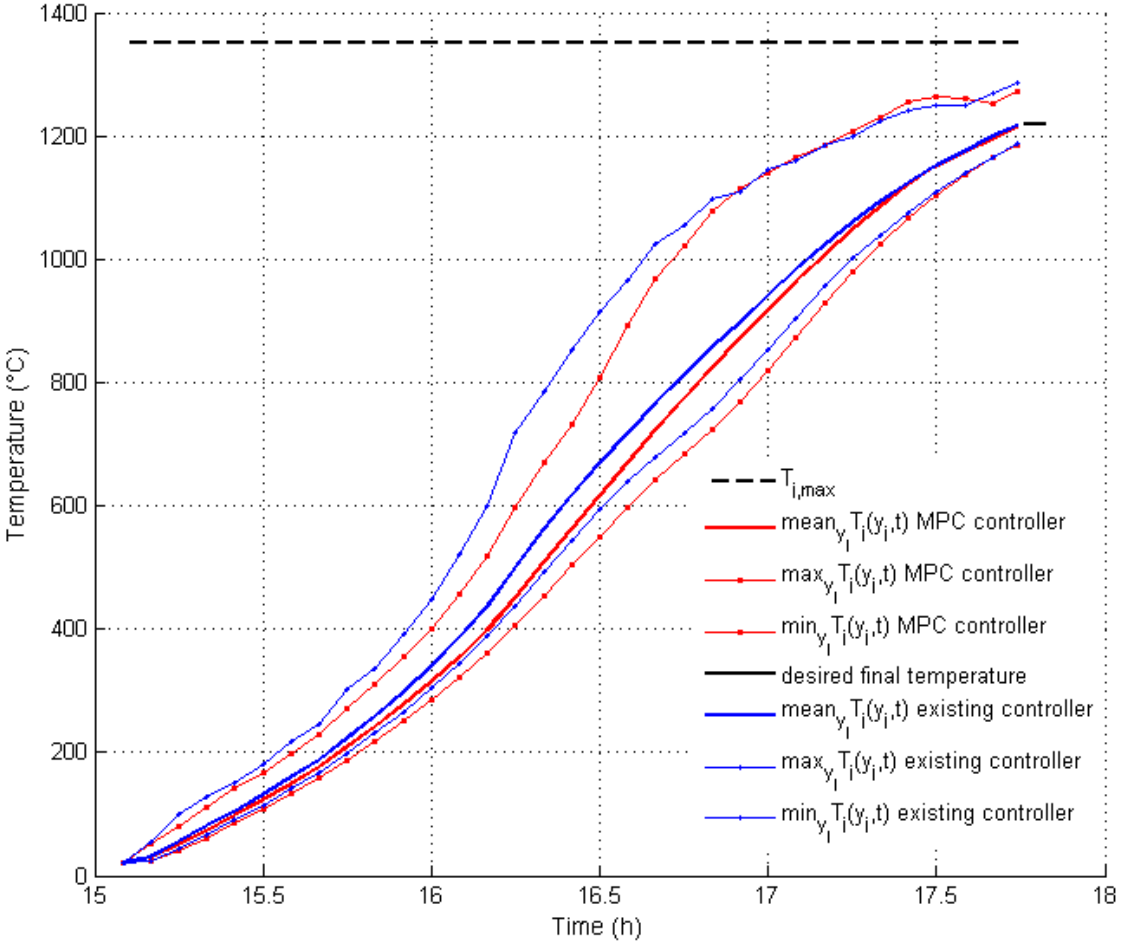

Figure 6.51: Temperature trajectory of slab $j = 200$

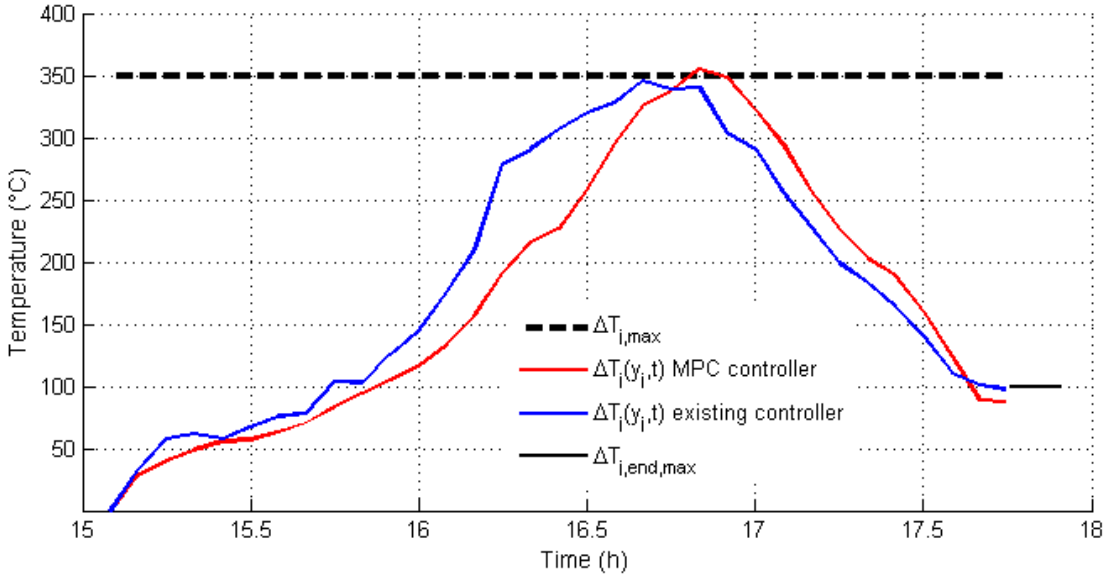

Figure 6.52: Temperature gradient of slab $j = 200$

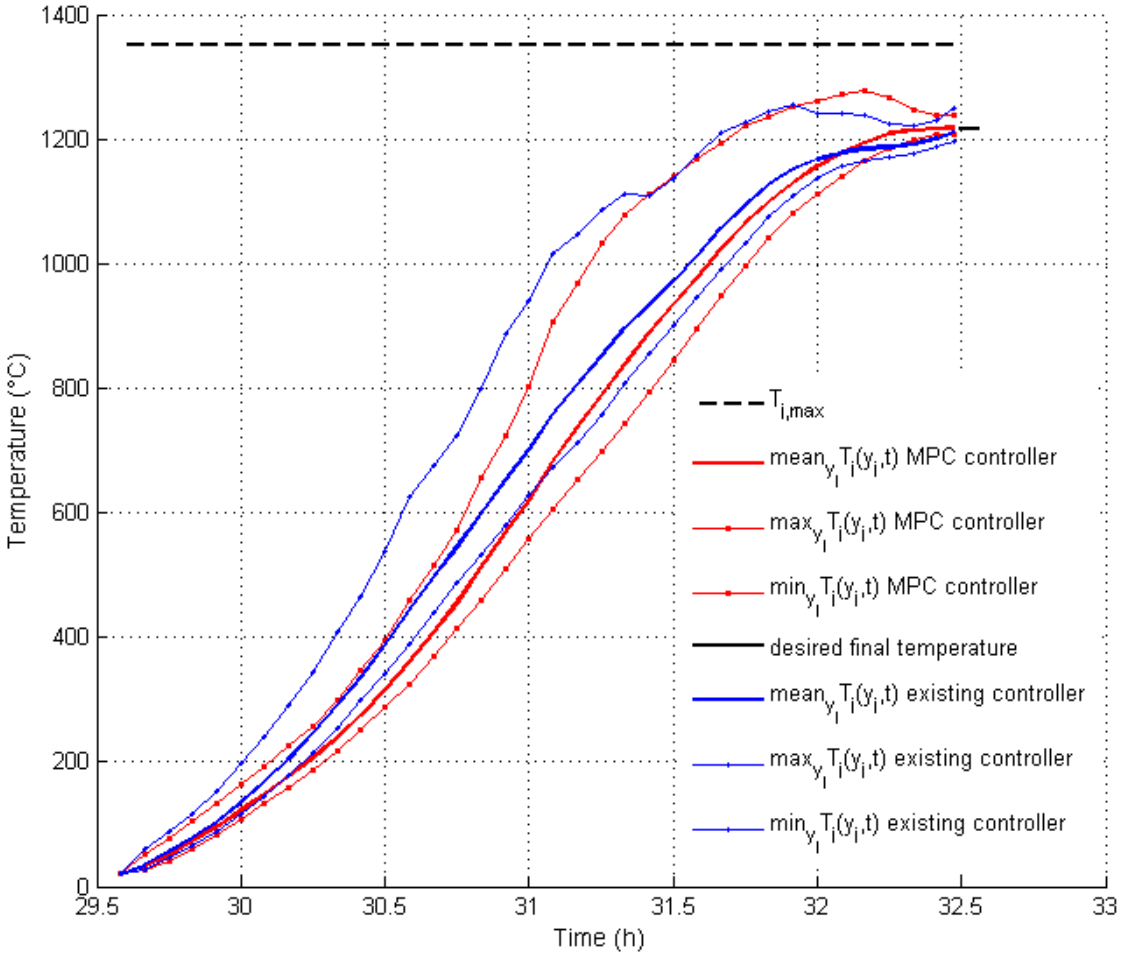

Figure 6.53: Temperature trajectory of slab $j = 400$

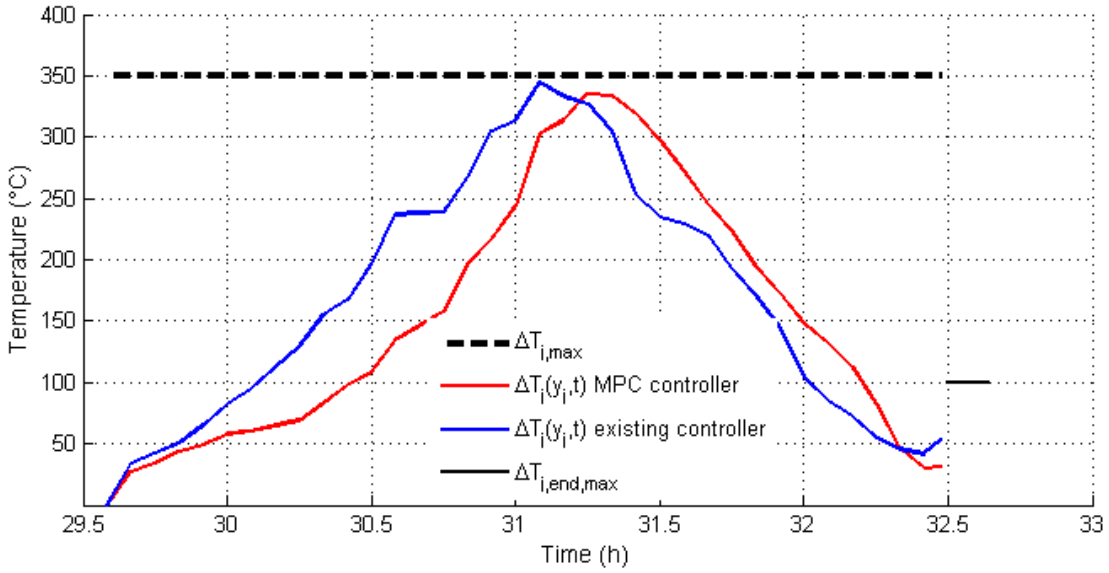

Figure 6.54: Temperature gradient of slab $j = 400$

temperature trajectories are shown in Figures 6.55, 6.57, and 6.59. We see clearly the effect of the stops at the end of the trajectories of slabs $j = 600$ and $j = 601$ where slab temperatures are highly homogeneous for both of controllers as shown in Figures 6.56 and 6.58. On the other hand, with slab $j = 641$ which is actually located at the entry of the furnace when the furnace stop happened, the temperature trajectory his slowly evolved at the beginning as shown with temperature gradients of Figure 6.60. Generally, the MPC controller gives lower final temperature gradients than simulated with the existing controller.

Finally, a general observation for the whole simulation of 900 slabs is given in Figure 6.61 and Figure 6.62. These figures show the average temperature trajectory and the temperature gradient of 900 slabs. With the given information of these figures, we can confirm that the MPC controller heat the slabs more gradually than the existing controller. The average temperature of slabs and the corresponding gradient is reduced in general and therefore energy can be saved.

The energy consumption of the furnace zones simulated with two controllers are shown in Figure 6.63. The red columns represent results of MPC controller and the blue ones are results of the existing controller. For two heating zones the MPC controller requires more fuel power than the existing controller. On the other hand, power of other zons, especially preheating zone, are higher when simulated with existing controller than with the MPC controller. MPC controller requires nearly no supply of fuel for the preheating zone which means that preheating zone is heated mainly by the exhaust gas. This can also explain why the MPC controller consumes for in the zone heating 1, since the exhaust gas heating the preheating zone come directly from the zone heating 1. Total fuel consumption of the furnace is shown in Figure 6.64 for the representative time interval [33h 42h]. We see that with MPC controller the furnace fuel consumption is more smooth, and in general lower than that simulated with the existing controller. The specific energy consumption, i.e, consumed energy per ton of steel heated is given in Table 6.9. The provided numerical values are normalized with a nominal value of specific energy consumption relating to the considered furnace. For the whole simulation of three-day furnace operation, the MPC controller

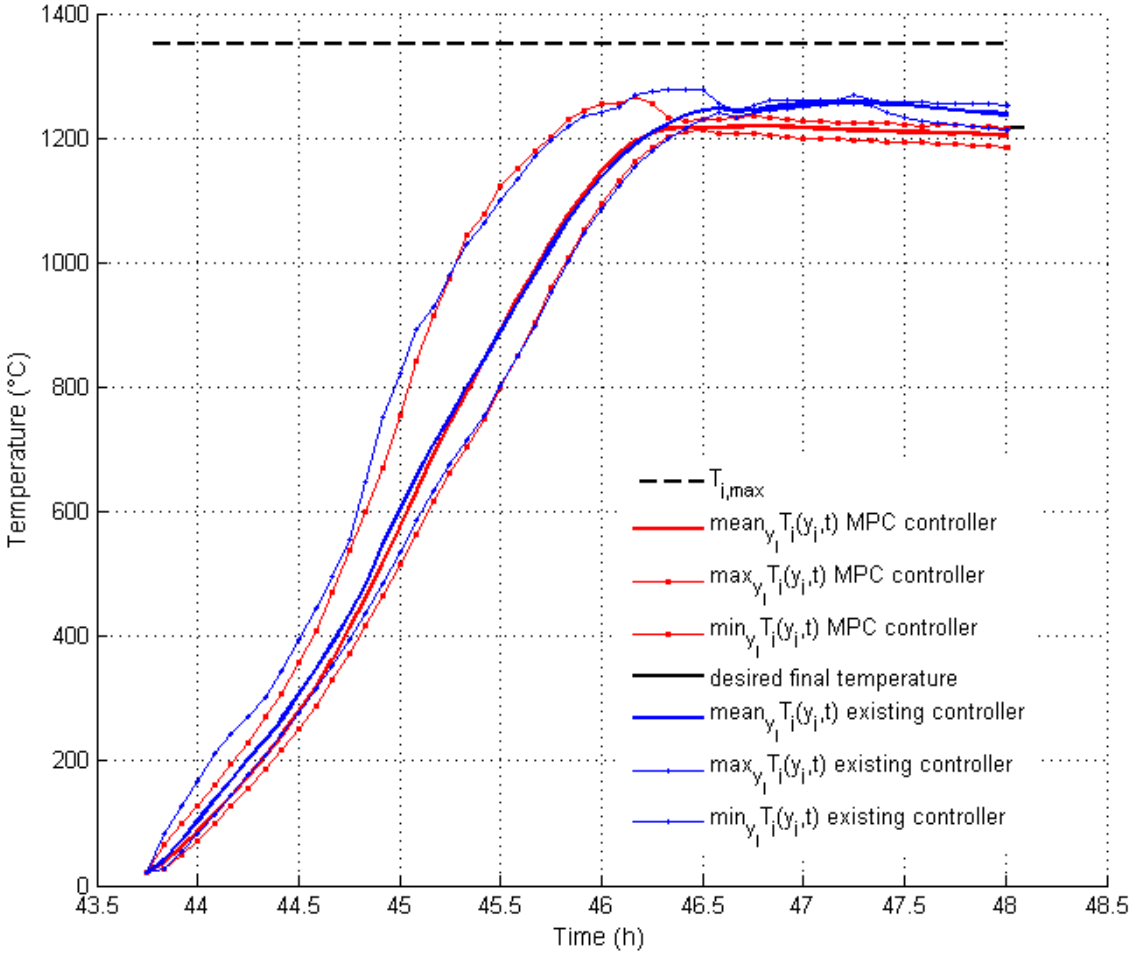

Figure 6.55: Temperature trajectory of slab $j = 600$

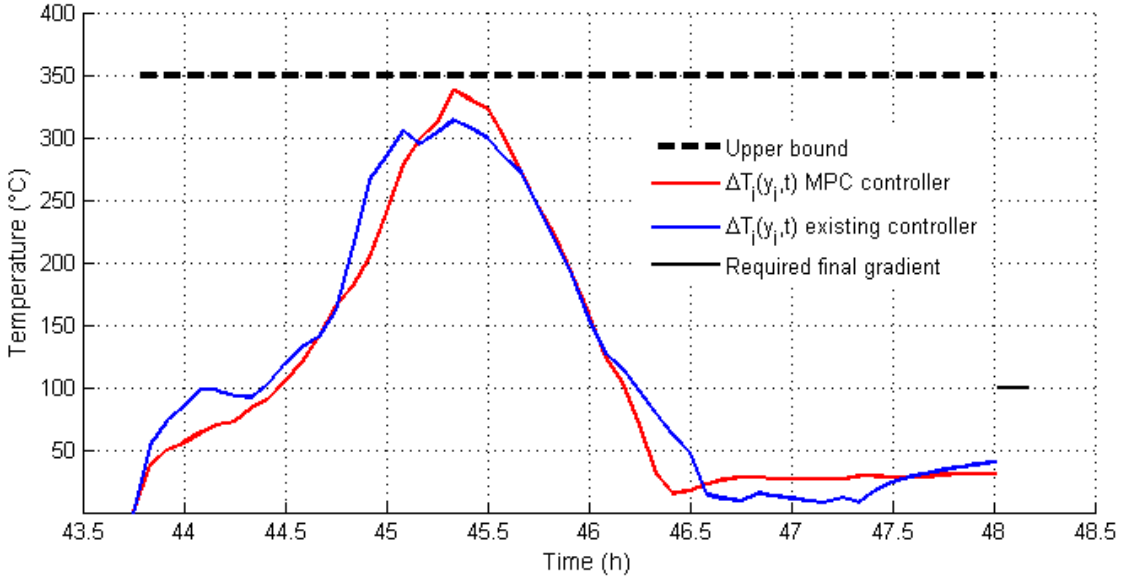

Figure 6.56: Temperature gradient of slab $j = 600$

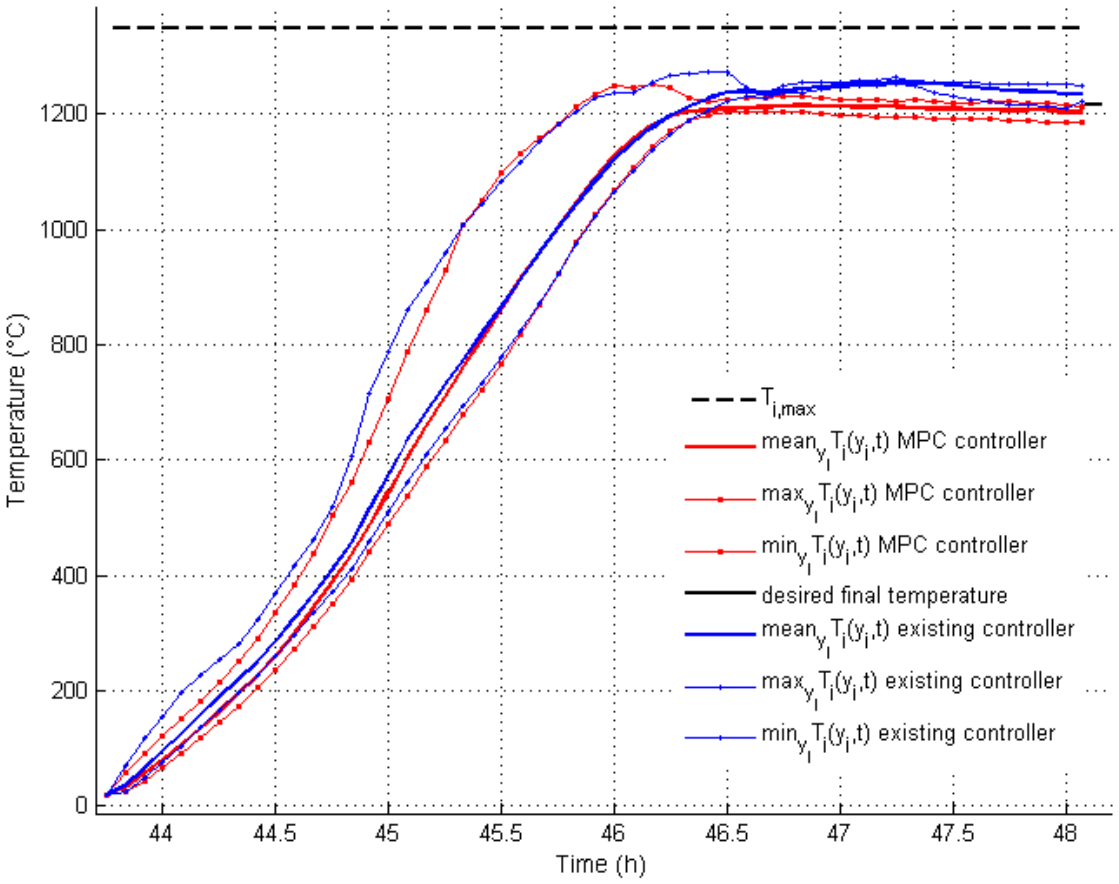

Figure 6.57: Temperature trajectory of slab $j = 601$

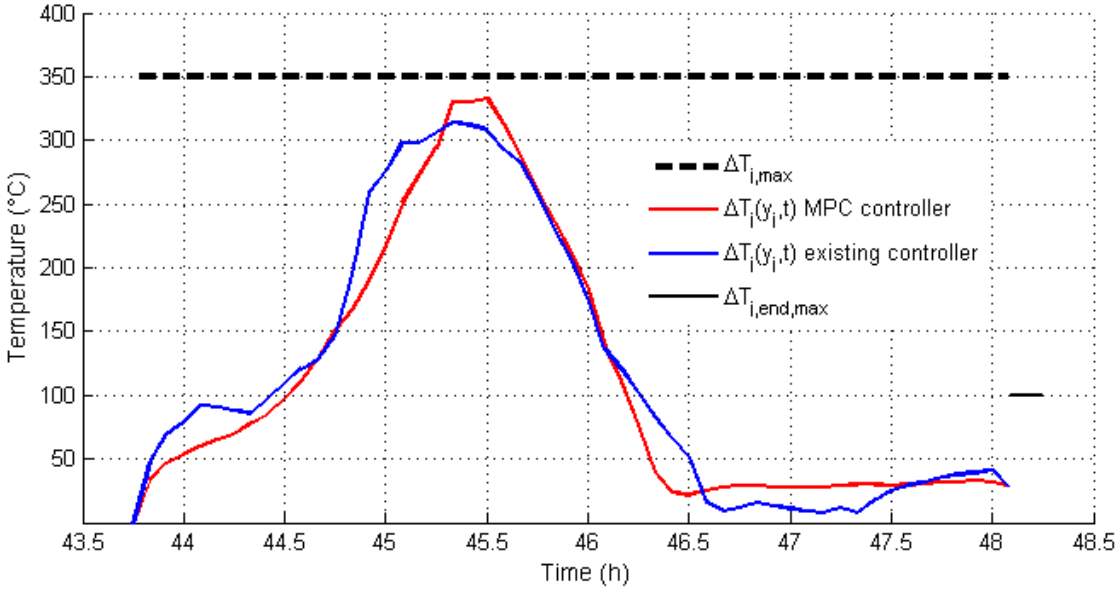

Figure 6.58: Temperature gradient of slab $j = 601$

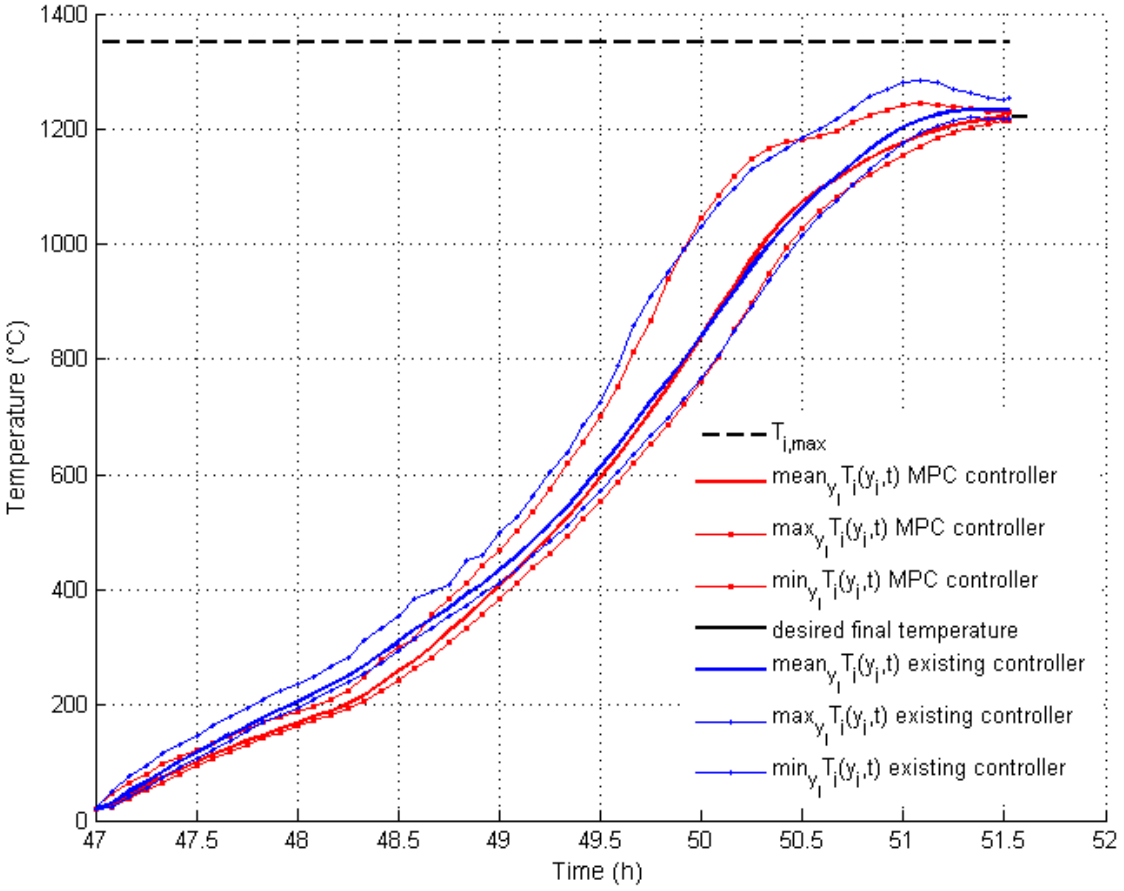

Figure 6.59: Temperature trajectory of slab $j = 641$

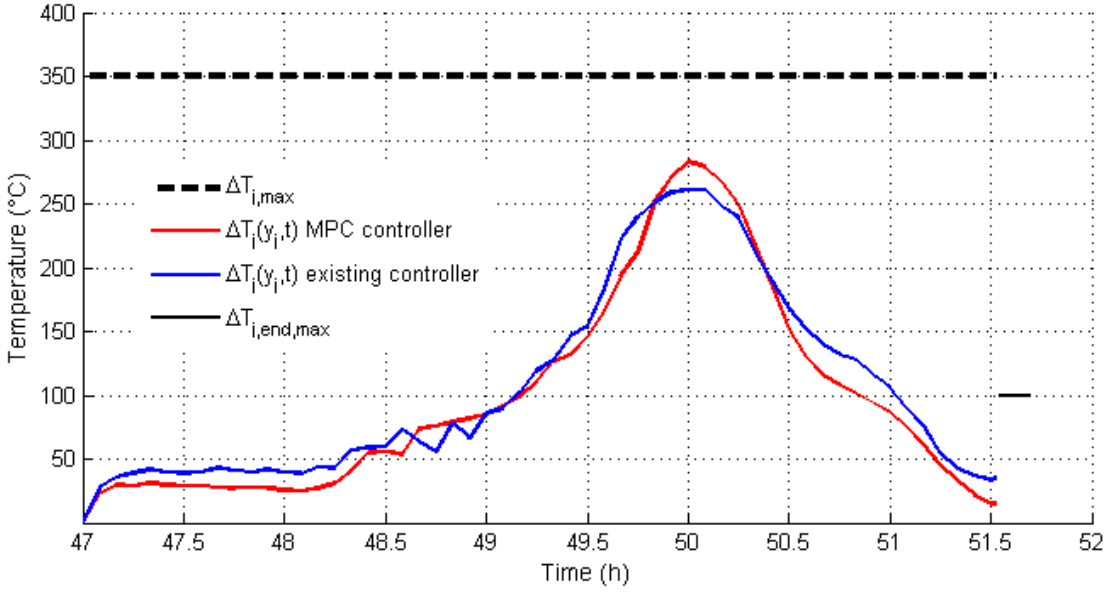

Figure 6.60: Temperature gradient of slab $j = 641$

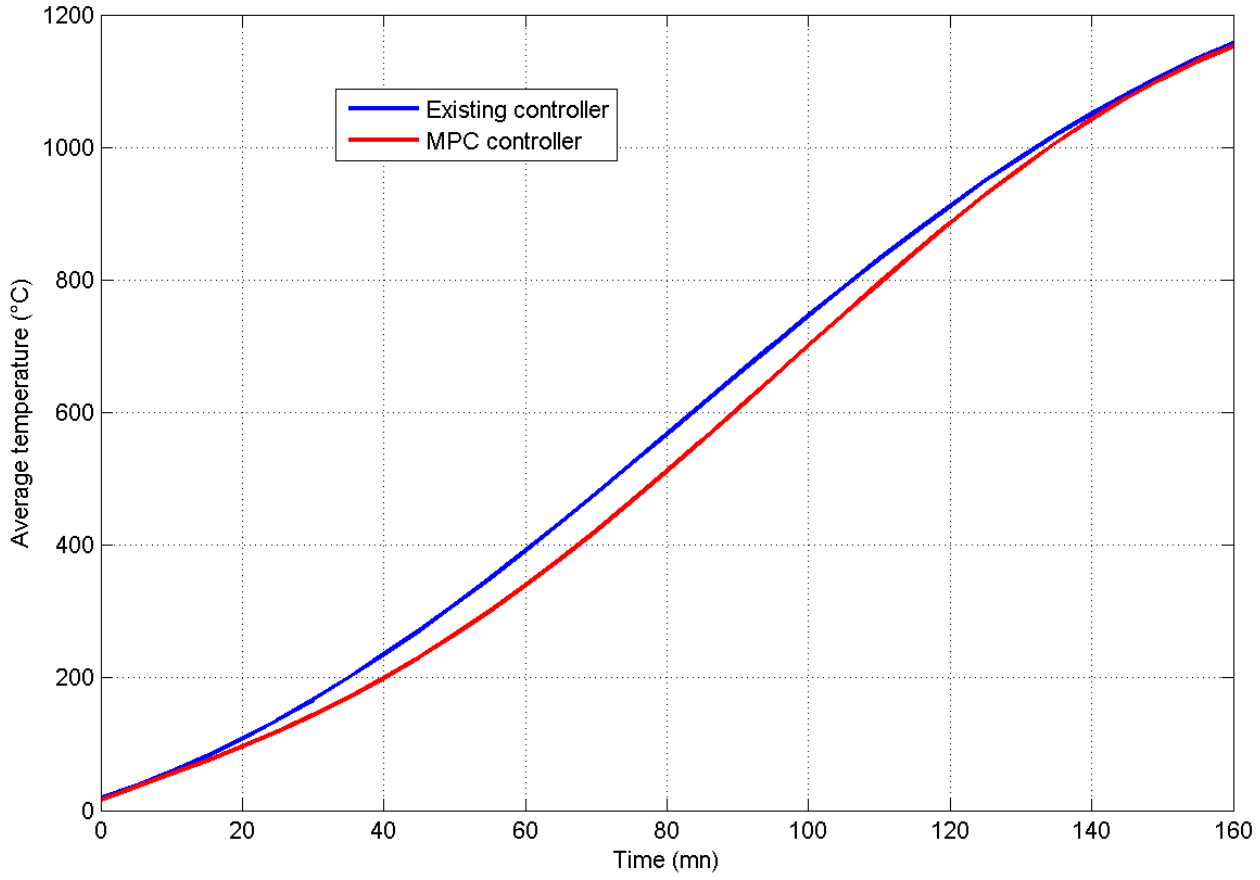

Figure 6.61: Average temperature trajectory of 900 slabs

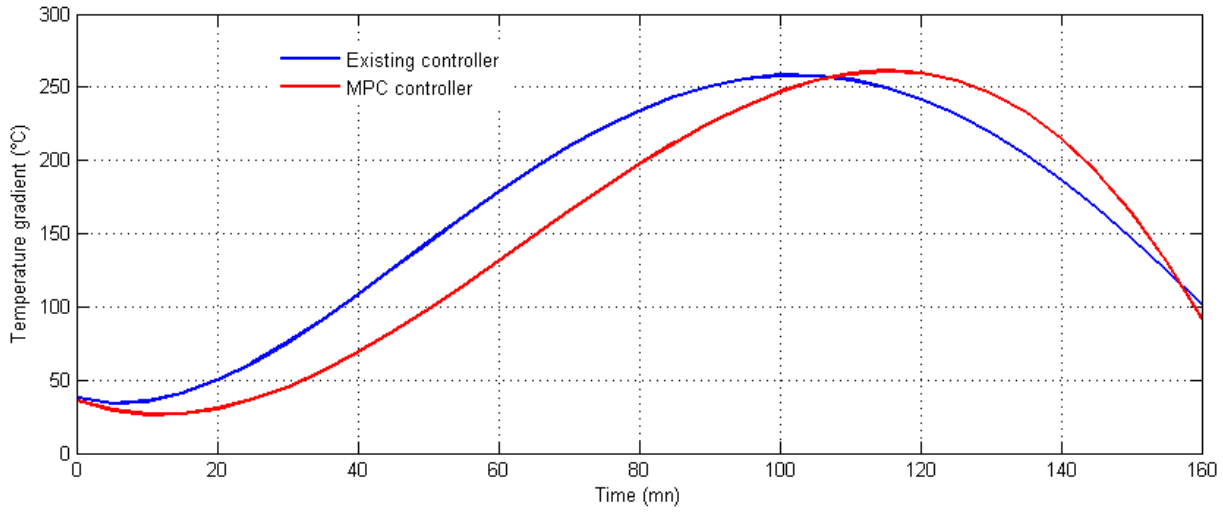

Figure 6.62: Average temperature gradient of 900 slabs

reduces from 4.6 % to 5.6% of energy consumption compared to the existing controller. In total, the energy saving is 5.2%

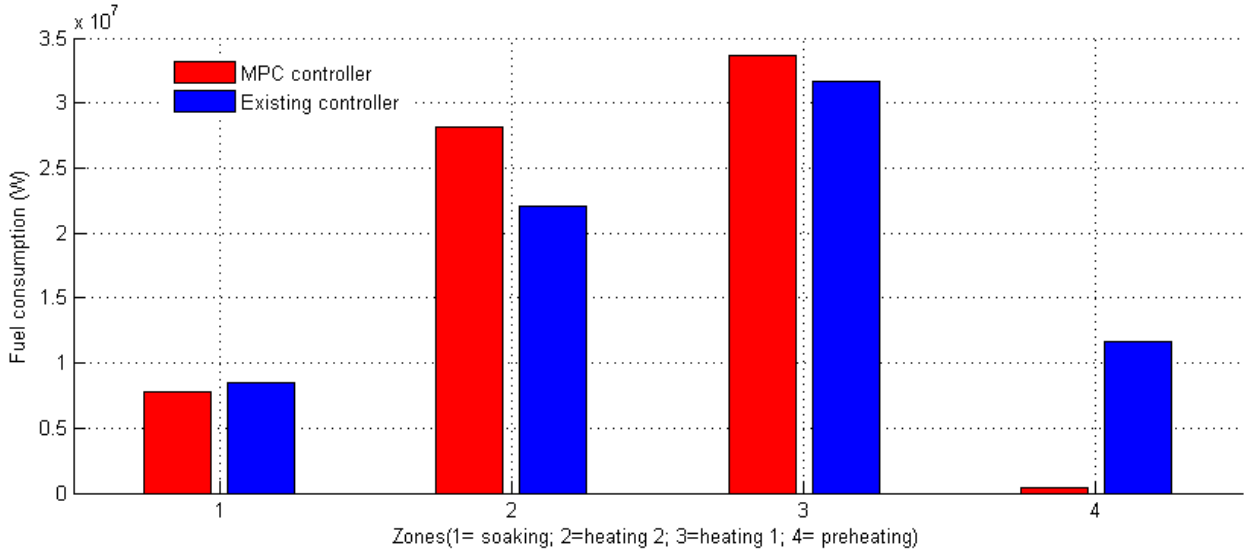

Figure 6.63: Fuel consumption of controllable zones

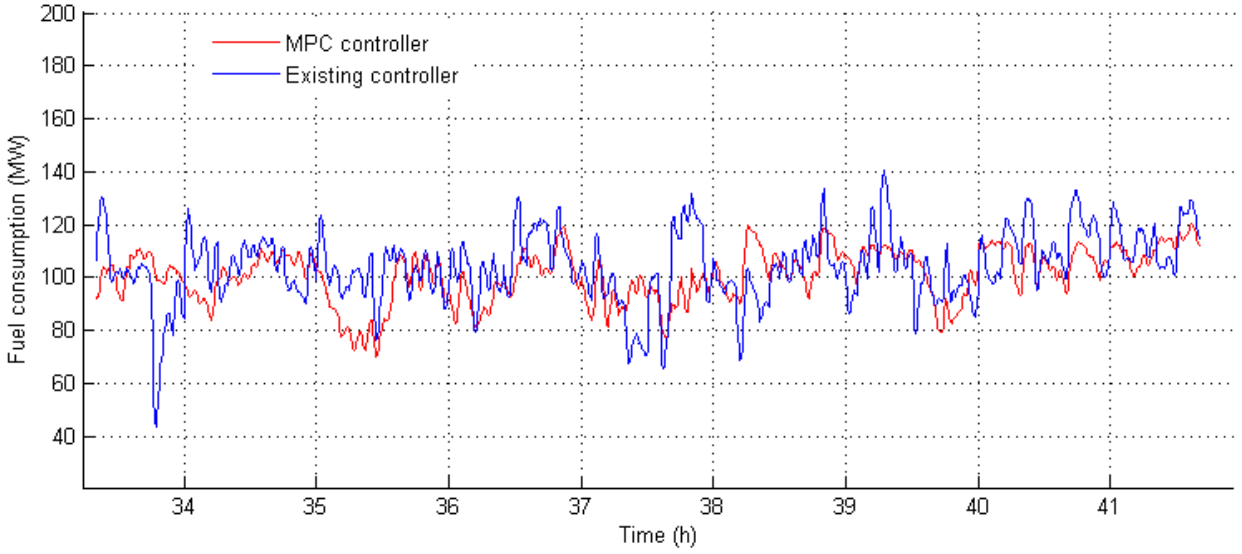

Figure 6.64: Total fuel consumption of the furnace

	MPC	Existing controller	Reduction (%)
Day 1	0.9852	1.0444	5.5
Day 2	1.0148	1.0667	4.5
Day 3	0.9111	0.963	5.6
Average	0.9704	1.0222	5.2

Table 6.9: Normalized specific energy consumption

6.5 Conclusions

The MPC strategy is designed for the level 2 of reheating furnace. The problem is the control of slab temperature so that it attains the desired final value. Firstly, the solution consists in formulating an optimization problem which is subject to furnace constraints. The cost function of the optimization problem takes into account the error of final slab temperature, temperature gradient inside slabs, energy consumption of the furnace, and general objectives such as reduction of furnace temperature level and slab temperature level. The furnace constraints concern the limits on zone temperatures and their rate of changes. Constraints on temperature of slabs are maximum temperature inside slabs, maximum temperature gradient of slabs, bounds on final slab temperature and final temperature gradients. These constraints can be accounted as penalty terms.

Secondly, the dynamic optimization required by MPC strategy is based on the simulation of furnace by a computer program in which the non-steady-state model is implemented. This model demands considerable computational resources. Therefore, gradient-based optimization methods are not those of interest because they need either explicit or approximation of derivatives of the cost functions. On the contrary, direct search methods need only the evaluations of cost function. The Nelder-Mead simplex method is well-known for its simple implementation, flexibility with constraints, here proved to be efficient for the design of MPC level 2 of reheating furnace. The simulation results on industrial scenarios show that energy consumption can be reduced up to about 5%, with higher homogeneities and accuracy of final slab temperatures.

In the next chapter, we reconsider the scheduling of slabs in order to boost the furnace productivity while assuring the heating requirement of slabs. The MPC controller proposed in the present chapter will be the basic of the controller constructed for scheduling task. Beside the criterions on accurate final slab temperatures, low temperature gradient and energy efficiency, the maximum throughput rate of furnace is the new criterion which is equally expressed by finding the minimum time interval between slab discharging events. A new control input added to the optimization is the average time interval between slab discharging events which is lower bounded according to the requirement of the hot rolling mill.

Part III

Improvement and Perspective

Chapter 7

Boosting of productivity

Slab temperature control may involve control of furnace temperature or control of slab velocity or both. In part II, we have focused on the problem of optimizing furnace temperature set-point in level 2 and control them in level 1 with a *predefined slab scheduling*. That is, the sequencing of slabs is known in advance with a table providing the time interval between loading and discharging events of consecutive slabs as shown in Figure 7.1. For example, we consider two consecutive slabs labeled *14P144* and *14P145*. Then, the predefined schedule imposes that the slab *14P144* will be loaded into the furnace 2 minutes after the slab *14P145*. Similarly, the slab *14P144* will be discharged from the furnace 4 minutes after the slab *14P145*.

In practice, when downstream process, e.g., hot rolling mill does not constitute a bottleneck of the production chain, the priority of reheating furnace is maximizing the throughput rate, and obviously, at the same time assuring the requirement of heating quality. Therefore, we will retake the solution previously constructed in chapter 6 and adapt it to the problem of maximizing throughput rate. This chapter will start with overview on published work on scheduling problem of reheating furnace. Afterwards, the problem statement clarifies the objectives and constraints of furnace control concerning slab scheduling. Then, the formulation of the problem is given. And finally, simulation results based on industrial data are presented and discussed.

Slab label	**Discharging gap (min)**	**Loading gap (min)**	Spatial interval (mm)	Thickness (mm)	Width (mm)	Length (mm)	Initial temperature (°C)	Desired final temperature (°C)
14P144	**5**	**4**	106	235	1159	8596	20	1215
14P145	**4**	**2**	122	235	1144	8606	20	1215
14P146	**4**	**4**	108	235	1157	8572	20	1215
14P147	**5**	**5**	115	235	1293	8622	20	1180
14P148	**4**	**4**	111	235	1170	8618	20	1205
14P149	**4**	**4**	112	235	1215	9243	20	1215
14P150	**5**	**2**	120	235	1266	7997	20	1215
14P151	**4**	**3**	108	235	1144	8634	20	1215
14P152	**4**	**3**	113	235	1247	9243	20	1215

Figure 7.1: Predefined scheduling table of slabs

7.1 Literature review

In published works for reheating furnace control, there are generally two classes of control: *production scheduling* and *furnace temperature control*. The production scheduling may ignore the effect of furnace temperature control and the non-uniformity of distributed heat power on the productivity of the furnace. Whereas, works on furnace temperature control often consider that production scheduling is predefined.

In [TLRY01], a review on scheduling method for integrated steel plant is provided. The authors emphasized on the direct path of slab from continuous caster to hot rolling mill, where reheating furnaces are not present or only used to slightly heat a small amount of slabs. This is hot-charging problem because at the furnace entrance, slabs are already at high temperature. Another review is given in [DF01] where production planning and scheduling of integrated steel plants are solved by mathematical methods.

In [LCG98], the production scheduling task of two identical reheating furnaces in a hot strip mill is determined by the tabular search heuristic. In this method, each slab is attributed with a desired forwarding velocity, and each furnace is run at the desired velocity of the slowest slab it contains.

In [PSV92], a heuristic scheduling system for a single slab reheating furnace and a rolling mill is presented. The system uses a modified greedy algorithm with small decision trees and total enumeration. An objective function that accounts for the empty space in the furnace, the discharging times of slabs and the deviations from ideal values of slab residence times is used.

In [ACK95], an algorithm for production scheduling of a reheating furnace and a hot rolling mill is designed based on a heuristic model. A cost function taking into account reheating times and fuel consumption and scheduling constraints are considered in the sequencing problem. The problem is solved by enumeration and a modified and pruning procedure. The constraints keep the problem size in a reasonable range. The utilized heuristic model of furnace requires that the residence time of slabs in each furnace zone is lower bounded.

In [BML07], the scheduling task of a reheating furnace is developed. A furnace model is used to calculate the temperature gradient and mean error of final slab temperature. An objective function tacking into account the later quantities is minimized with a genetic algorithm.

In [JKZ05], a reheating furnace for billets is studied. The influence of predefined space between billets on furnace productivity is analyzed. The study showed that an optimum space between billets exists, i.e., maximal furnace productivity can be achieved while assuring the heating requirements. For the considered furnace, the products to be heated are steel slabs. Spaces between slabs are predefined and do not significantly influence the furnace productivity because these spaces are much smaller than slab's width and length. Therefore, this space between slabs will not be included in the productivity study of the considered furnace.

In the past, the optimum furnace zone temperatures and forwarding velocity for certain kinds of products are often tabulated for steady-state operation as presented in [Led86a]. In [KMP94], a nonlinear PI-type furnace controller is proposed that uses only the slab velocity as a control input. In the control of a small reheating furnace with

two zones, the nonlinear model predictive strategy utilizing zone temperatures and forwarding speed as control input is presented. The constraints of upstream and downstream are not taken into account.

Another nonlinear predictive control scheme for furnace control is presented in [BBG08]. The control inputs are the fuel flows of furnace zones and the conveying speed. The developed method firstly finds the steady-state conditions that maximize the production profit (productivity, energy consumption, heating quality). Then a linearized model around the steady-state point is used to construct a dynamic optimization. For this later problem, the objective function accounts for the furnace constraints, and the deviation of the system from the steady-state optimum (optimized speed and fuel flows) found by the static optimization. The obtained simulation results showed that the fuel consumption can be reduced and throughput rate of furnace is increased. This is because the system can respond more quickly to changes in material flows and delays.

Recently, a method for slab scheduling is developed in [SK13] to find the minimum reheating time of slabs. A switched analytic model of reheating furnace is able to compute enthalpy absorbed by slabs. Based on this model, a quadratic programming problem is formulated taking into account furnace constraints. The optimal solution is the smallest reheating times needed to attribute sufficient heat for slabs. An example scenario in which three furnaces operate in parallel obtained significant results like: higher accuracy of final slab temperature, lower temperature gradient and increased throughput rate.

In summary, most of the published works utilized either furnace temperature or slab velocity as control inputs. Only few combine these two strategies. We want to contribute to the filling in of this gap by utilizing the MPC method. In this chapter, based on the numerical non-steady-state model presented in chapter 3, we design MPC strategy to improve at the same time furnace productivity, heating quality and energy consumption. The control inputs are time gaps between discharging and loading events of slabs, and the furnace zone temperature set-points. In the following, constraints and objective of scheduling task in reheating furnace are clarified.

7.2 Problem statement

Boosting productivity of the furnace and optimizing furnace temperature control cannot be done as separated tasks. This is because maximum reheating power of a furnace is in fact a dynamic quantity, which depends, at each sampling time, on furnace load and the current furnace thermal state (slab, walls and gas temperature, etc.). As a result, the optimization of furnace scheduling must take into account the heating requirement, furnace constraints as well as productivity objective and constraints of hot rolling mill, i.e.,

- The temperature trajectories of slabs are upper bounded by $T_{j,max}$.
- Final slab temperature profiles are limited with $T_{j,end,max}$ and $T_{j,end,min}$.
- Trajectory of temperature gradient of slabs are upper bounded by $\Delta T_{j,max}$ so that slab deformations are avoided.

- Final slab temperature gradient should be as small as possible and stay below a certain required value $\Delta T_{j,end,max}$.
- Maximizing the throughput rate of processed steel slabs. This implies that forwarding velocity should be as high as possible, or time intervals between consecutive discharging events should be as small as possible.
- Furnace is limited with certain productivity due to constraint of hot strip mill. Therefore, the time interval between discharging events are limited with a lower bound $\delta t_{sch,min}$.
- Slabs are often arranged into montage according to its material properties, with different montage the cylinders of hot rolling mill need to be changed; so a space between these sets of slabs is imposed giving time for necessary replacements. Therefore, the time interval between these montage will also be imposed meaning not dependent on δt_{sch}.
- Stoppages of furnace can be scheduled or unscheduled events, and they are unavoidable and need to be included in the simulation.

7.3 Problem formulation

The sequencing of slabs through the furnace is determined with a table as example shown in Figure 7.1, in which time intervals between consecutive loading and discharging events are predefined. Therefore, the smaller these time intervals are the more throughput rate of the furnace is increased. Because the furnace can only support certain level of slab load, the discharging and loading are usually synchronized, meaning if one slab is discharged, another slab at the furnace entry would be loaded. As a result, the average time interval between discharging and loading of consecutive slabs are in general equal. Consequently, we can use either the time intervals of slab discharging events or those of slab loading events as extra control inputs along with zone temperature set-points. Considering the complexity of the non-steady-state model that we want to use for the optimization of MPC controller, it would be preferable that the number of input variable has a low dimension to reduce number of furnace simulation needed. Therefore, we should use only an average value of the time intervals as the extra control input. More specifically, for each furnace simulation, we can redefine the slab sequencing by replacing the current values of time interval in table of Figure 7.1 with an value provided by the optimization algorithm.

Because the slab scheduling is reconsidered each sampling time, reheating time of a slab can change constantly. As a result, if we use a fixed prediction horizon for the optimization problem, the number of slabs discharged of the furnace can change from one furnace simulation to another provided with the different time intervals. For example, if the time interval of discharging is small then more slab will be discharged during the prediction horizon period and inversely, if the time intervals are long then there will be less slabs discharged in the same period of time. In the formulation objective functions, number of slabs discharged is a crucial quantity which affects the calculated results. We shall want the number of discharged slabs during the prediction horizon to be constant. Thus, MPC strategy will be applied with a variable prediction horizon. The following variables are considered for scheduling optimization tasks:

- Average time interval of discharging events $\delta t_{sch}(k)$. And obviously, $\delta t_{sch}(k) \geq 0$
- Lower bound of time interval of discharging events $\delta t_{sch,min}(k)$
- Sampling time of level 2: Δt_s
- Number of slabs considered in the prediction horizon is a constant number N_s, whereas the prediction horizon $[k, k+N_P(k)]$ defined with $N_P(k) = \lceil N_s \times \delta t_{sch}(k)/\Delta t_s \rceil$ is time-varying dependent of $\delta t_{sch}(k)$.
- Fixed control horizon $[k, k + N_C - 1]$ with $N_C \leq N_P(k)$
- Control input $u(k)$ is now defined by the zone temperature set-points and the average time interval of discharging events, i.e.,

$$u(k) = [\delta t_{sch}(k), T_{z,1}(k), T_{z,2}(k), ..., T_{z,M}(k)] \tag{7.1a}$$
$$= [u_1(k), u_2(k), ..., u_{M+1}(k)]. \tag{7.1b}$$

- Control input sequence $U(k) = \{u(k),\ \forall t \in [k, k+N_C-1]\}$. It shall be noted that $u(k) = u(N_C),\ \forall k \in [N_C, N_P]$.
- Loading and discharging events of a slab j correspond to: $k_{j,0}$ and $k_{j,1}$, respectively, with

$$k_{j,0} = int(t_{j,0}/\Delta t_s) \tag{7.2a}$$
$$k_{j,1} = int(t_{j,1}/\Delta t_s). \tag{7.2b}$$

In comparison to the MPC method proposed in Chapter 6, the difference of MPC constructed in this chapter dedicated to the boosting of productivity is that we use one more variable concerning slab sequencing $\delta t_{sch}(k)$ and the prediction horizon N_P is a time-varying parameter.

The objective and constraint concerning throughput rate are added along with the heating objectives and constraints that have been formulated in Chapter 6. Especially, the weighting coefficients used in the objective functions are chosen by the same approach defining the priority of each slab according to its position inside the furnace.

The requirement of final slab temperatures are formulated as the sum of weighted final temperature errors of slabs that are discharged during the prediction horizon, i.e.,

$$J_1\left(U(k)\right) = \sum_{i=1}^{N_P(k)} w_1(k+i) \times |T_s(k+i) - \tilde{T}_s(k+i)|, \tag{7.3a}$$
$$= \sum_{j \in S_1(k)} w_1(k_{j,1}) \times |T_s(k_{j,1}) - \tilde{T}_s(k_{j,1})|, \quad k_{j,1} \in [k, k+N_P(k)]. \tag{7.3b}$$

And, objective of homogeneity of slab is the sum of weighted temperature gradients as

$$J_2(U(k)) = \sum_{i=1}^{N_P(k)} w_2(k+i) \times g(k+i) \tag{7.4a}$$

$$= \sum_{j \in S_1(k)} w_2(k_{j,1}) \times g(k_{j,1}), \qquad k_{j,1} \in [k, k+N_P(k)]. \tag{7.4b}$$

The criterions on energy consumption is expressed as

$$J_3(U(k)) = \sum_{i=1}^{N_P(k)} w_3(k+i) \sum_{j=1}^{M} P_j(k+i). \tag{7.5}$$

And, general objective function of reducing furnace temperature is formulated as

$$J_4(U(k)) = \sum_{i=1}^{N_C} w_4(k+i) \sum_{j=2}^{M+1} u_j(k+i). \tag{7.6}$$

The general objective to reduce slab temperature trajectory is written as

$$J_5(U(k)) = \sum_{i=1}^{N_P(k)} [x_1'(k+i) \times w_5(k+i) \times x_1(k+i)]. \tag{7.7}$$

In particular, objective of maximum throughput rate is translated into minimum average time interval between discharging events. That is, the criterion on scheduling objective can be given as

$$J_6(U(k)) = \sum_{i=1}^{N_C} w_6(k+i) \times u_1(k+i) \tag{7.8}$$

where $w_6(k)$ is a time-dependent coefficient defined with

$$w_6(k+i) = \frac{N_C - i + 1}{N_C} \tag{7.9}$$

The constraints of zone temperature is now expressed as

$$J_{c,1}(k) = \sum_{j=1}^{N_P(k)} \sum_{i=2}^{M+1} [\max(0, u_i(k+j) - T_{z,i,max}(k+j)) + |\min(0, u_i(k+j) - T_{z,i,min}(k+j))|] \tag{7.10a}$$

$$J_{c,2}(k) = \sum_{j=1}^{N_P(k)} \sum_{i=2}^{M+1} [\max(0, \dot{u}_i(k+j) - \dot{T}_{z,i,max}(k+j)) + |\min(0, \dot{u}_i(k+j) - \dot{T}_{z,i,min}(k+j))|]. \tag{7.10b}$$

The constraints on final slab temperature gradient is given as

$$J_{c,3}(k) = \sum_{i=1}^{N_P(k)} \max(0, g(k+i) - g_{max}(k+i)) \tag{7.11a}$$

$$= \sum_{j \in S_1(k)} \max\left(0, g(k_{j,1}) - \Delta T_{j,end,max}\right). \tag{7.11b}$$

The upper and lower bound on final slab temperatures are formulated as

$$J_{c,4}(k) = \sum_{i=1}^{N_P(k)} [\max(0, T_s(k+i) - T_{s,max}(k+i)) + |\min(0, T_s(t) - T_{s,min}(k+i))|] \tag{7.12a}$$

$$= \sum_{j \in S_1(k)} \max(0, T_s(k_{j,1}) - T_{s,max}(k_{j,1})) + |\min(0, T_s(k_{j,1}) - T_{s,min}(k_{j,1}))|]. \tag{7.12b}$$

The slab temperature trajectory and temperature gradient are limited by upper bounds T_{max}, ΔT_{max}, respectively.

$$J_{c,5}(k) = \sum_{i=1}^{N_P(k)} [\max(0, [x_1(k+i) - T_{max}(k+i)]'[x_1(k+i) - T_{max}(k+i)])] \tag{7.13}$$

$$J_{c,6}(k) = \sum_{i=1}^{N_P(k)} [\max(0, [x_2(k+i) - \Delta T_{max}(k+i)]'[x_2(k+i) - \Delta T_{max}(k+i)])] \tag{7.14}$$

Important constraint of scheduling task is the minimum time interval between discharging events $\delta t_{sch,min}$, i.e.,

$$J_{c,7}(k) = \sum_{i=1}^{N_C} |\min(0, \delta t_{sch,min}(k+i) - \delta t_{sch,min})|. \tag{7.15}$$

This is the extra constraint added to the heating problem formulated in chapter 6. If the time interval is lower than the limit, the furnace discharge too fast and the downstream process, e.g., will not be able to tolerate it. Therefore, the penalty term $J_{c,7}(k)$ will have a positive value every time the upper limit of time interval is not respected.

The scheduling optimization task is constructed as an unconstrained minimization problem given by

$$\min_{U(k)} J_o(U(k)) = \sum_{i=1}^{6} \alpha_i \times J_i(U(k)) + \mu \sum_{i=1}^{7} J_{c,i}(U(k)) \tag{7.16}$$

The weighting coefficient μ usually has high value to penalize the cost function when constraints are violated. Other coefficients α_i ($i \in [1, 6]$) are chosen according to the importance of the objectives. Most important objectives are success of desired final slab temperatures and low temperature gradients. In second places, there are objectives of maximizing production rate and reducing energy consumption (reducing zone temperature and fuel supply).

7.4 Simulation results

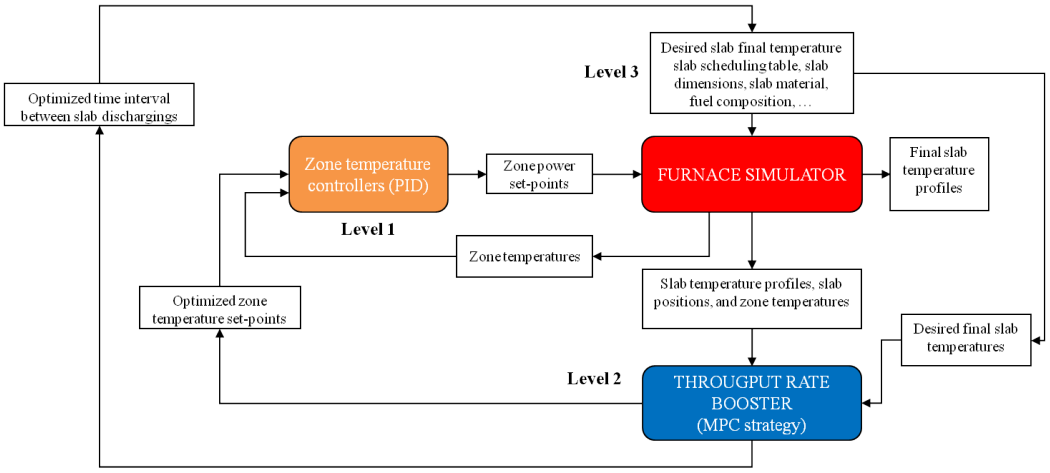

Figure 7.2: Simulation diagram of scheduling optimization

We test the forgoing design of MPC strategy for the scheduling optimization of the three-day furnace operation presented in chapter 6. The simulation scheme is given in Figure 7.2. In this scheme, in addition to optimized set-points of zone temperature, the optimization also delivers the optimal time interval of the discharging events. The latter value will be used to update the slab scheduling. In practice, it is a little difference since scheduling optimization by MPC strategy, instead of directly modifying the slab sequencing, will propose to the furnace operator the new time interval that increases productivity and at the same time satisfies the heating requirements, see Figure 7.3.

The simulation is based on real furnace data. Hot rolling mills can receive slabs which have thickness in the range of 229 mm to 235 mm. We consider a real operating scenario where 900 slabs (20200 tons of steel) have been reheated within 72 h. The average reheating time of a slab is 170 min. The number of slabs taken into account for each sampling time is fixed at 20. This number represents the number of slabs located inside controllable zones. Sampling time is 5 min. Control horizon is limited by complexity of the furnace model, therefore, N_C is simulated with the value of 1. The minimum of average time interval between discharging event is $\delta t_{sch,min} = 2$ min which corresponds to the minimum time necessary for transporting a slab away from the furnace exit door so that the space is available for new discharge.

In Figures 7.4 and 7.5, the results of optimized discharging time interval are shown. Slabs having discharging time interval higher than 10 mn such as slab $j = 240$ and $j = 425$ indicate stops of the furnace. We see that after long scheduled stops as in period from $j = 640$ to $j = 850$ the MPC controller gives lower discharging time interval meaning higher throughput rate.

In term of final slab temperatures, we can compare the results of MPC controller and the existing controller in Figures 7.6 and 7.7. Firstly, MPC controller avoids slab overheating. And in general, MPC controls more accurately the final slab temperatures.

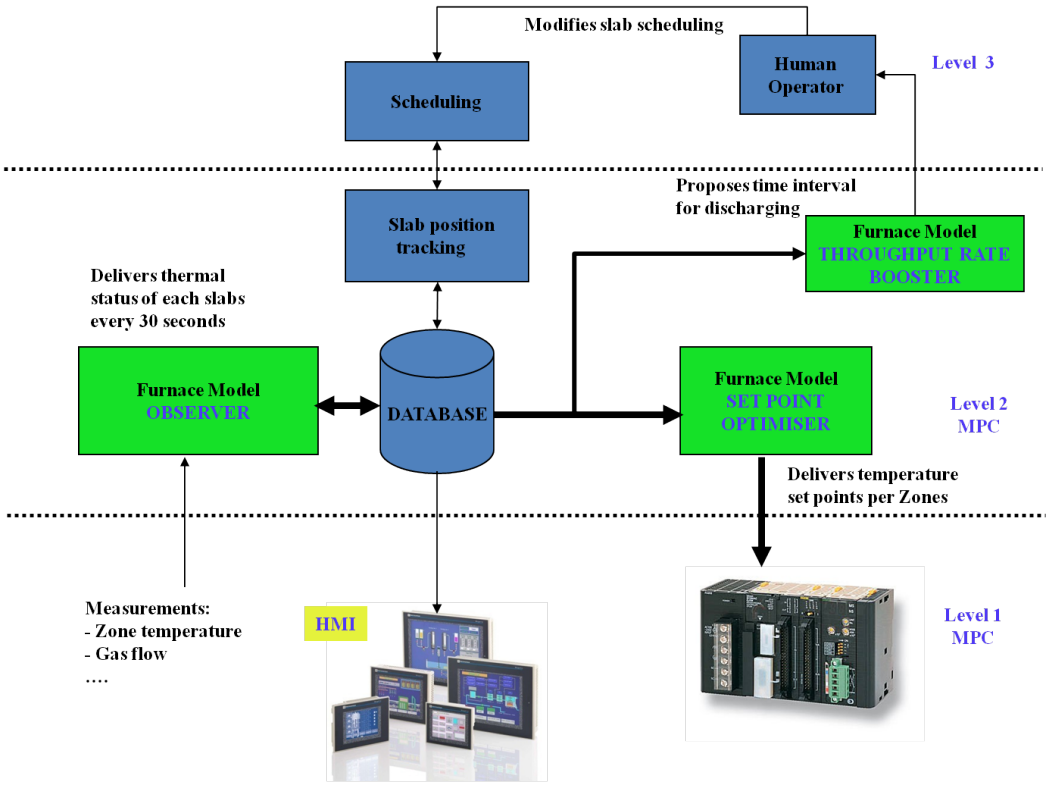

Figure 7.3: Implementation diagram of MPC strategy

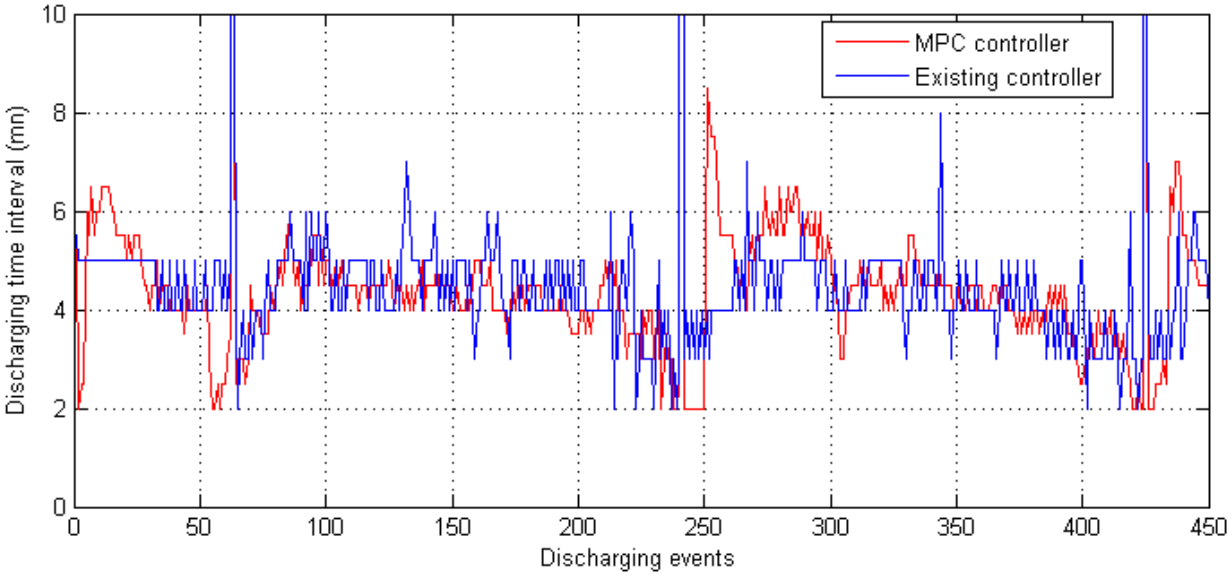

Figure 7.4: Discharging time interval of events $j = 1$ to $j = 450$

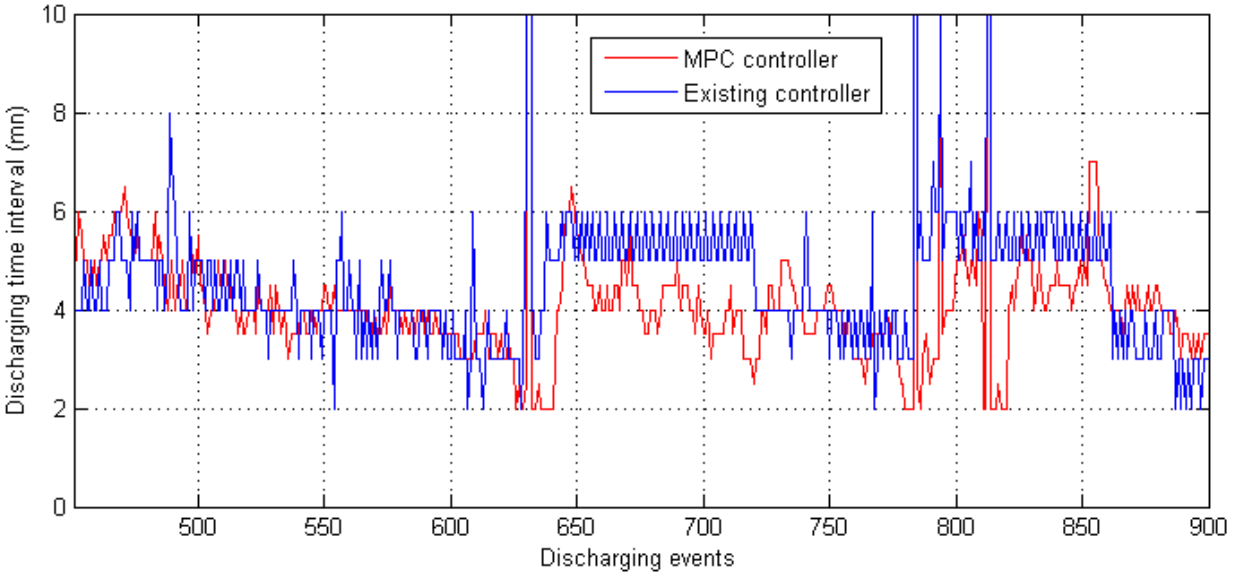

Figure 7.5: Discharging time interval of events $j = 451$ to $j = 900$

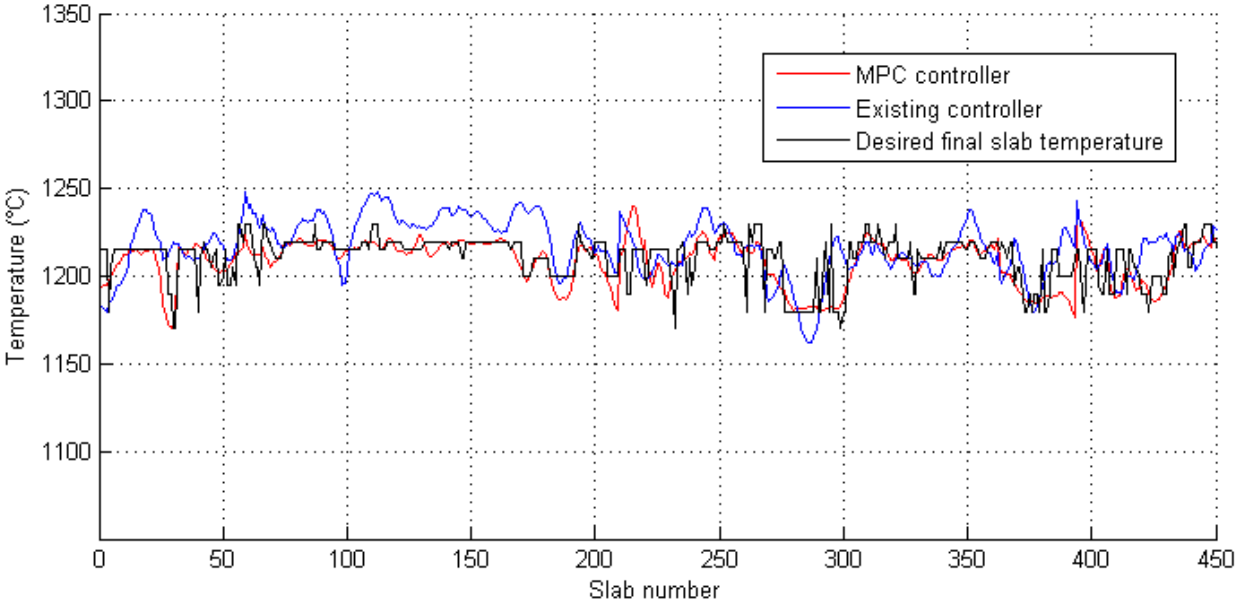

Figure 7.6: Average final temperature of slabs $j = 1$ to $j = 450$

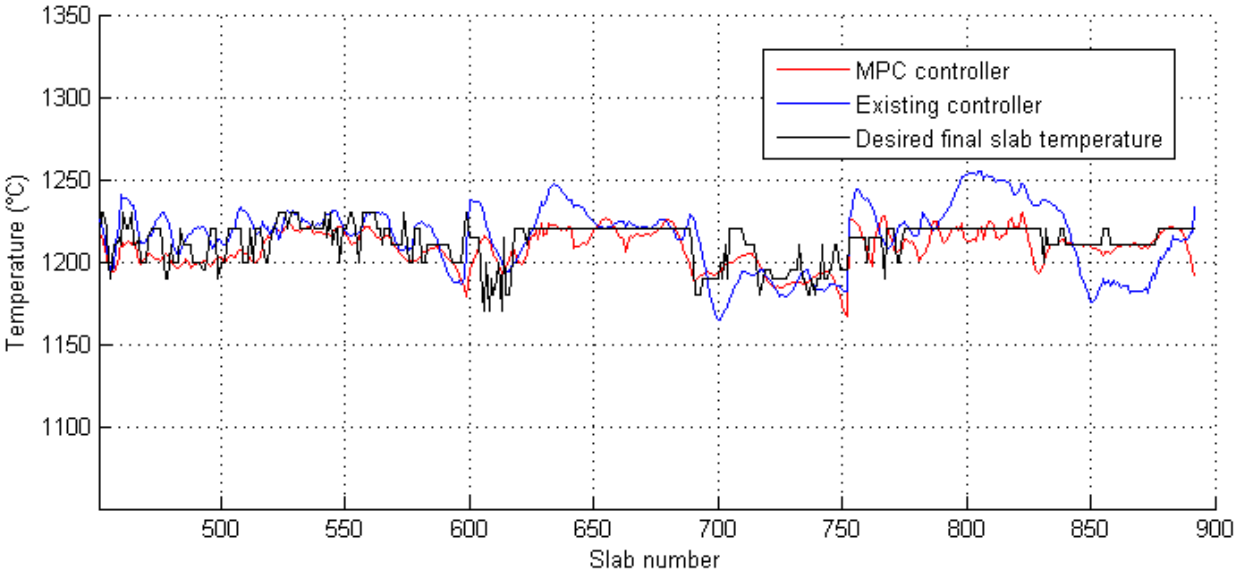

Figure 7.7: Average final temperature of slabs $j = 451$ to $j = 900$

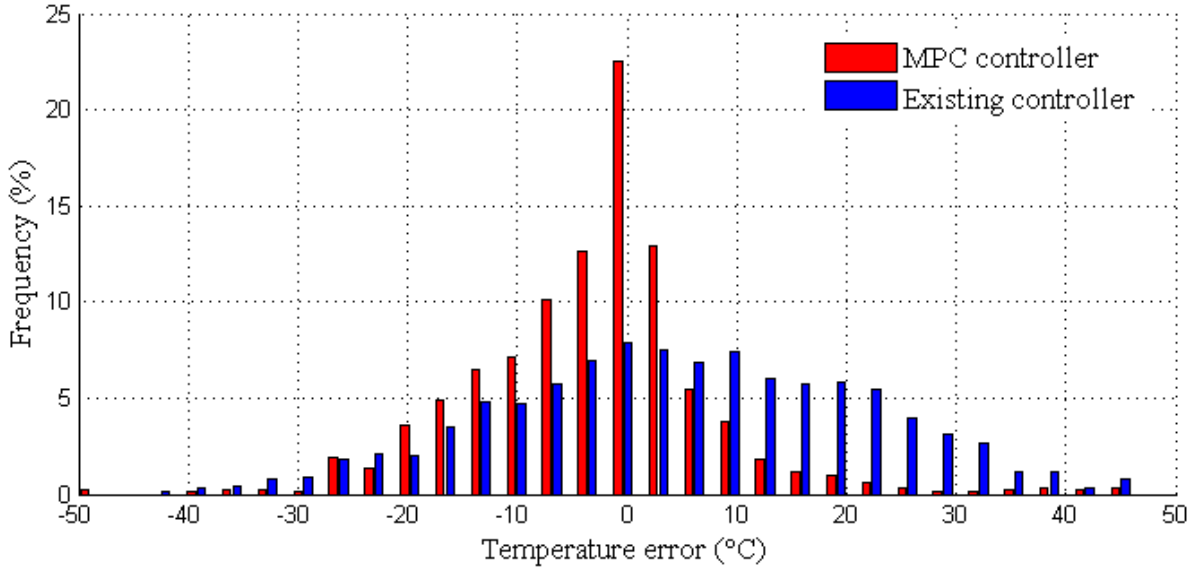

Figure 7.8: Histogram of final slab temperature error

More specifically, Figure 7.8 shows that MPC strategy gives more exact final slab temperature with a higher frequency of final temperature error at 0. The average error of final slab temperature given by MPC is -3 °C with a standard deviation of 10 °C in Table 7.1, which is better than given by the industrial data 5 °C and 17 °C respectively.

Final slab temperature gradient is shown in Figure 7.9. We see that in general the final slab temperature gradients simulated with MPC controller are more regular between slabs compared to the existing controller. The average temperature gradient of slab simulated with MPC controller is reduced of 7 °C with a standard deviation decreased from 22°C to 7 °C as shown in Figure 7.10 and Table 7.1.

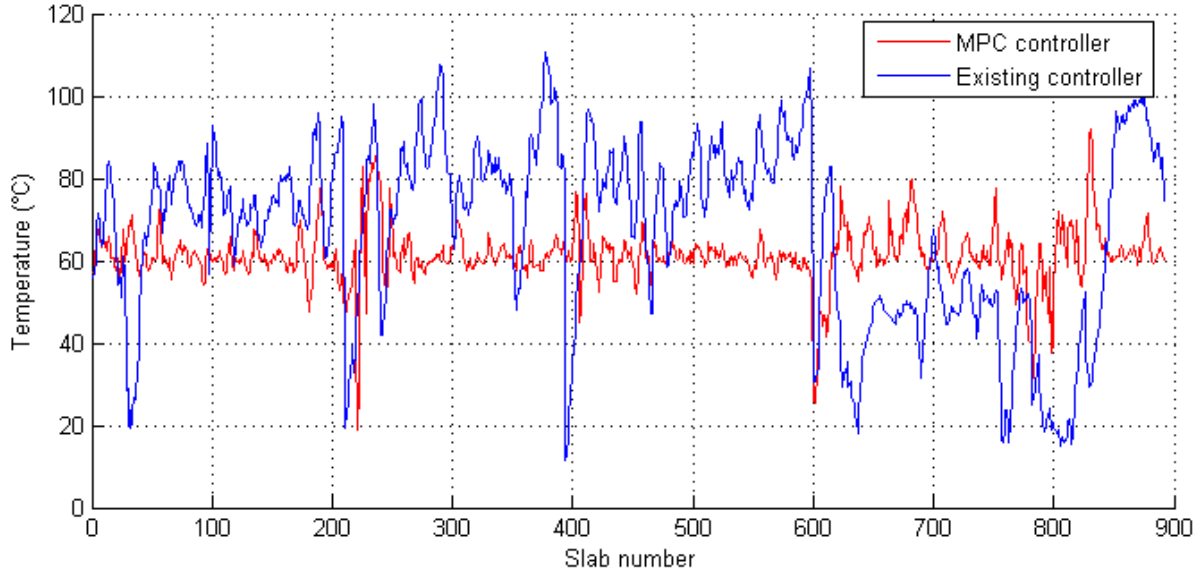

Figure 7.9: Final slab temperature gradient

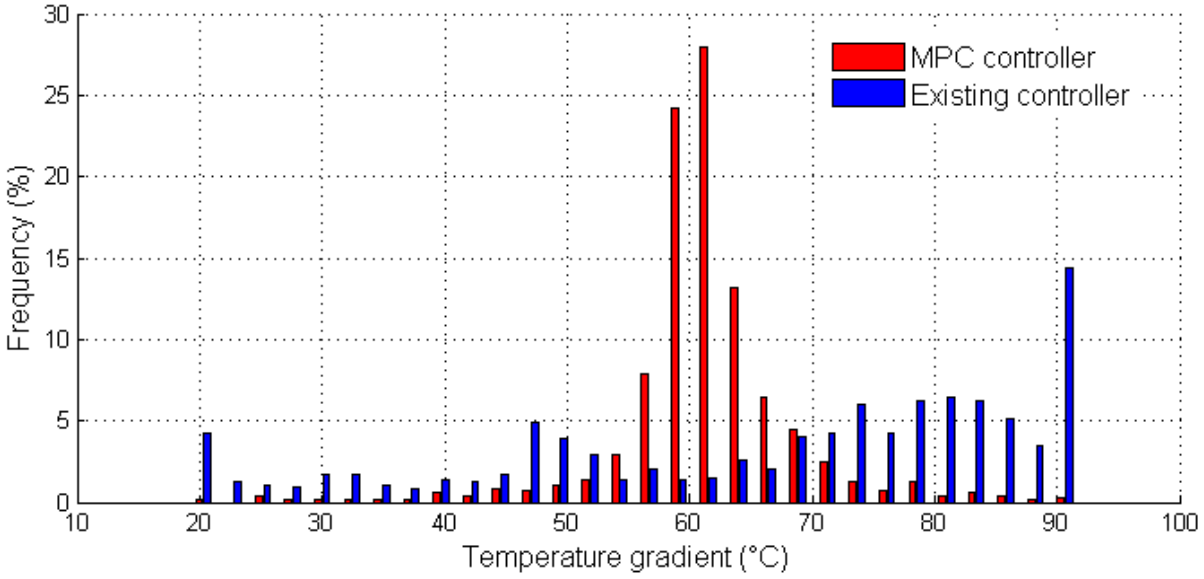

Figure 7.10: Histogram of final temperature gradient

Quantity	MPC	Existing controller
Average of ($T_s(k)$-$\tilde{T}_s(k)$) (°C)	-3	5
Standard deviation of ($T_s(k)$-$\tilde{T}_s(k)$) (°C)	10	17
Average of g(k) (°C)	61	68
Standard deviation of g(k) (°C)	7	22

Table 7.1: Final slab temperature and temperature gradient with MPC strategy and the existing controller

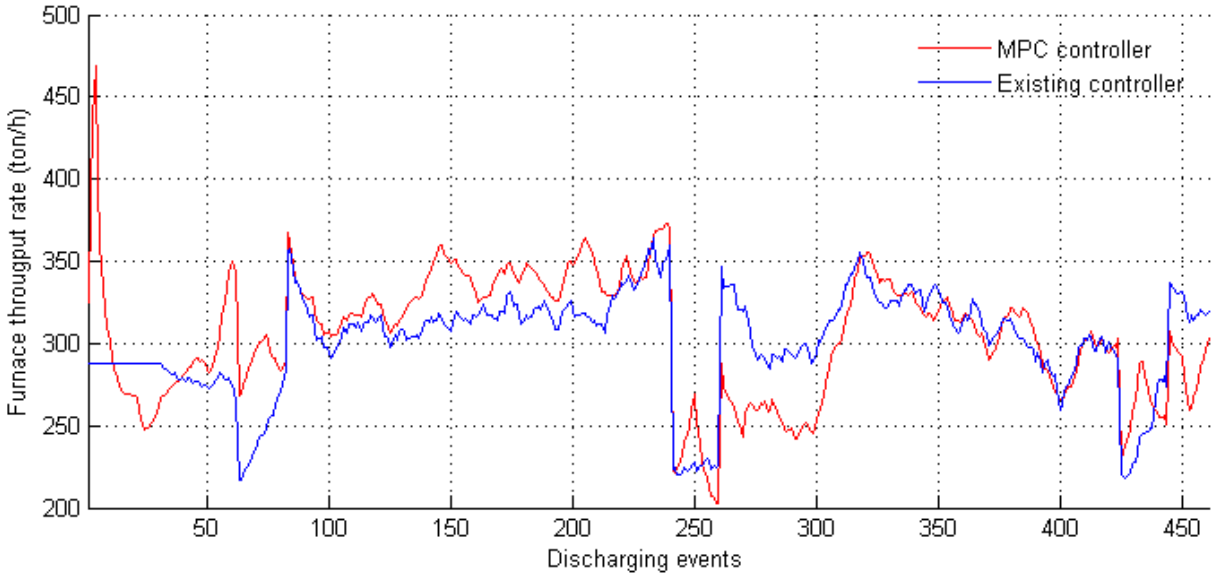

Figure 7.11: Furnace throughput rate measured at discharging events $j = 1$ to $j = 450$

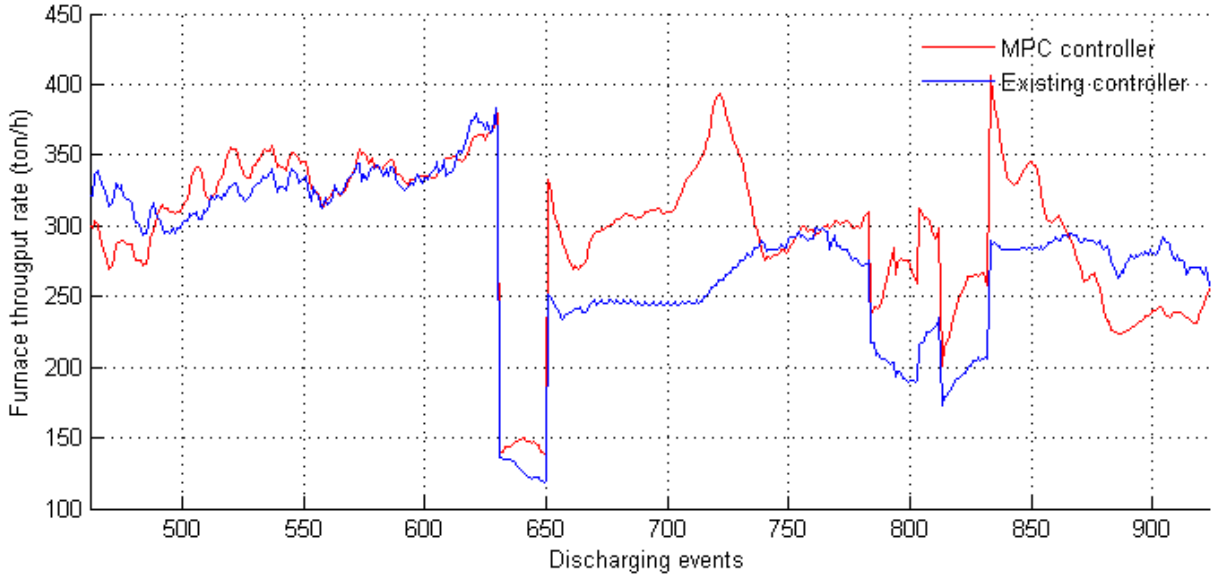

Figure 7.12: Furnace throughput rate measured at discharging events $j = 451$ to $j = 923$

In term of throughput rate, different results are presented in Figures 7.11, 7.12 and Table 7.2. It should be noted that the estimation of throughput rate is synchronized with discharging events and not with time because, at each optimization, MPC controller modifies the scheduling of the slabs. The throughput rate can be increased by 15 t/h with MPC in comparison to the existing controller. The total heating time and average reheating time are also reduced. The price of higher production rate is higher energy consumption which is increased from 1.25 Gj/t to 1.27 Gj/t.

In Figure 7.13, we can see the average temperature of slab number 570 with MPC and with existing controlled in the same time reference. The slab is heated faster with MPC controller and therefore discharged sooner than with

Quantity	MPC	Existing controller
Average throughput rate (t/h)	296	281
Average reheating time (min)	169	179
Total heating time (h)	68	72
Specific energy consumption (Gj/t)	1.27	1.25

Table 7.2: Production rate and energy consumption indicators

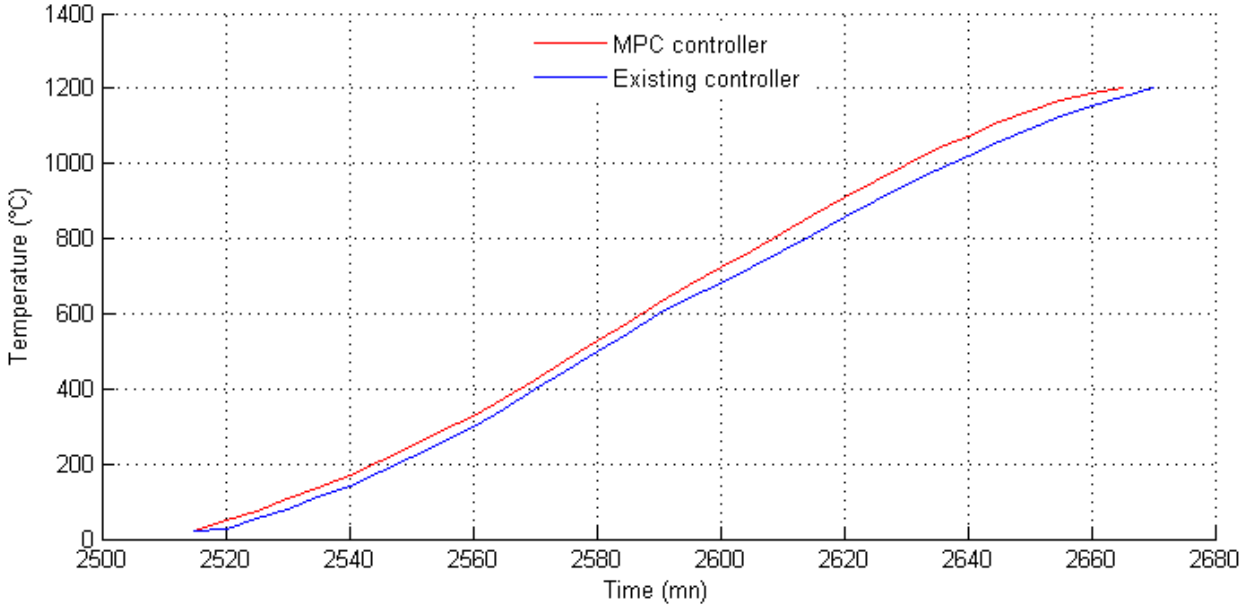

Figure 7.13: Average temperature of slab number 570

the existing controller. The temperature gradient is lower with MPC than with the existing controller as shown in Figure 7.14.

In term of reheating time of slabs, the results obtained with MPC strategy is compared with that of the real data as shown in Figure 7.15. In general, with MPC strategy, reheating times of slab are reduced. For both results of MPC and the real data,when the reheating time of slab is increased suddenly, it means that there is a stop. A short stop of furnace can be about 20 min. And, a long stop is more than 50 min. We can see that when a short stop happens MPC does not reduce reheating time of slabs. However, when a long stop happens, MPC strategy can reduce significantly the reheating time of slabs. This means MPC reacts faster to the stops than the current method. For this reason, the production rate of the furnace is increased as shown in Table 7.2.

7.5 Conclusions

This chapter has widened the furnace control from a predefined scheduling strategy to a variable scheduling strategy where MPC method is utilized to optimize the zone temperature set-points, and at the same time to find the maximum possible throughput rate of the furnace.

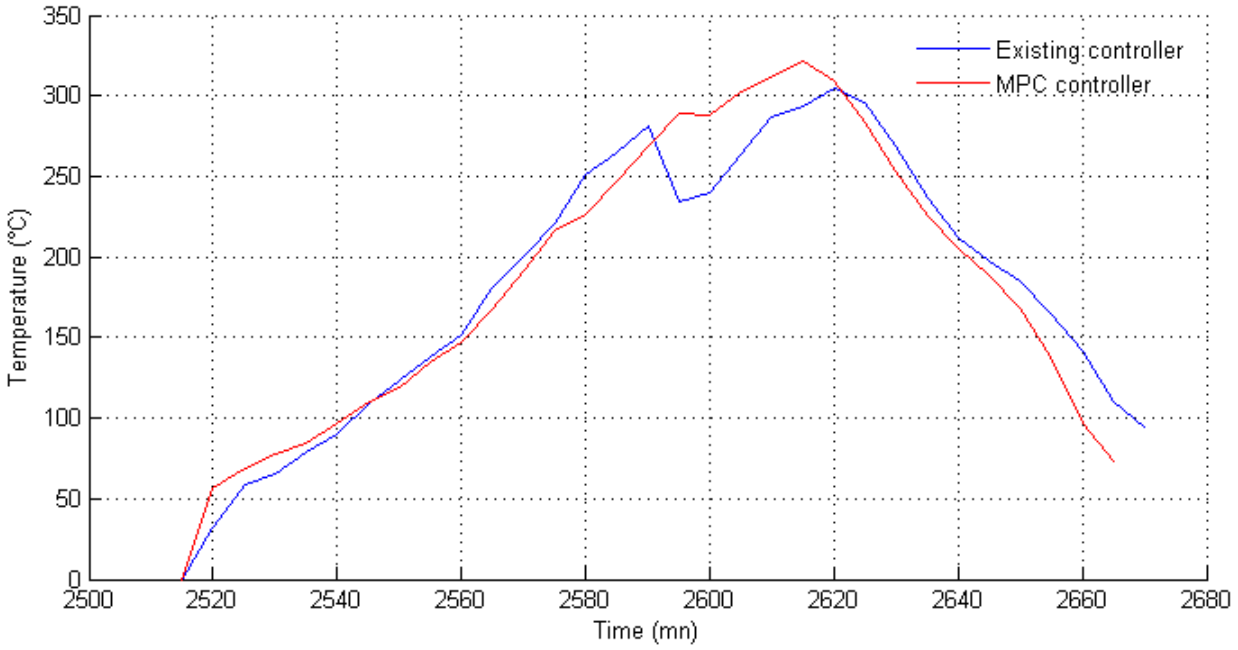

Figure 7.14: Average temperature gradient of slab number 570

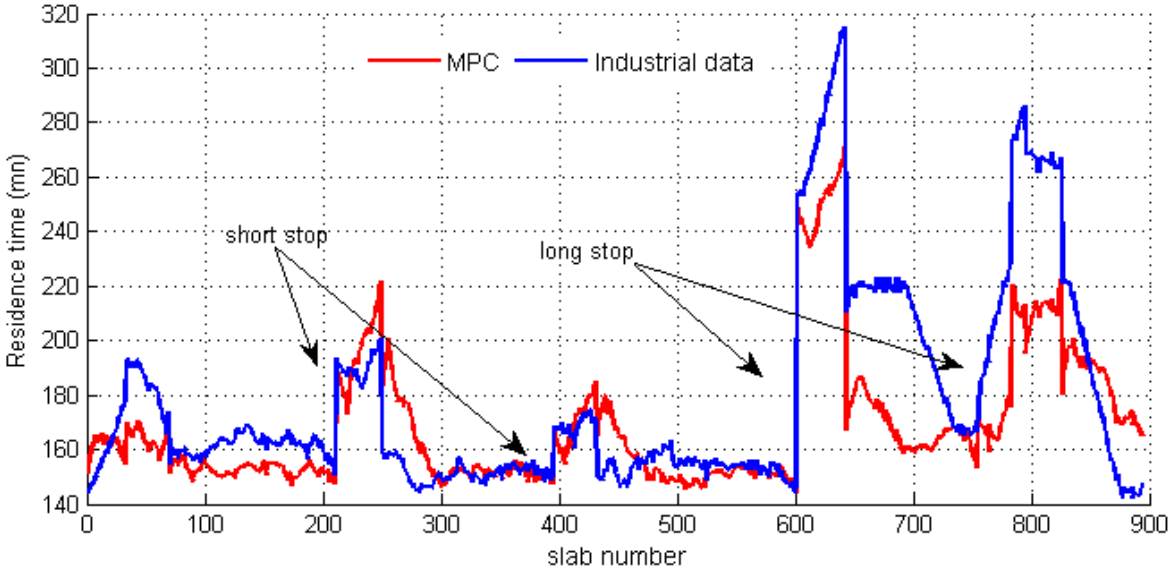

Figure 7.15: Reheating time of slabs

The MPC controller designed for boosting of productivity is basically developed from the solution constructed in the chapter 6. Heating objectives and constraints are the same as those formulated of predefined scheduling. Beside the zone temperature set-points, the time interval of discharging events is added as a control input. The latter variable actually is the direct index of furnace throughput rate since the faster slabs are discharged at the required temperatures the higher furnace productivity is. Therefore, an new objective of control is to minimize the time interval between discharging events. On the other hand, the hot rolling mill is limited with a upper bound. As a result, an additional constraint is represented by a lower bound of the time interval.

When the new control input, the time interval between discharging events, is varied from one furnace simulation to another, the number of discharged slabs over a fixed prediction horizon changes. Therefore, the calculation of objective functions will be affected since they account for final temperature and temperature gradient of every individual discharged slab. As a result, instead of utilizing a fixed prediction horizon, we used a variable horizon changing according to the time interval while the number of discharged slabs over the prediction horizon is constant.

The design is tested with the industrial data of three-day furnace operation. Simulation results show that the furnace throughput rate can be increased up to 15 ton per hour while furnace consumption is also increased of 0.02 Gj/ton. Considering the economical benefit of increased throughput rate compared to the price of an energy unit, it is usually preferable to push the furnace to its maximum capacity. In term of heating requirement, the MPC controller shows to be more accurate over the final slab temperature than the existing controller and especially the slab temperature gradients are reduced significantly.

Chapter 8

Extension to other metallurgical furnaces

Besides slab walking-beam reheating furnaces, other types of process in which steel products are heat treated are also big energy consumers of a steel plant. These processes are often divided into zones which are controlled at different level of temperatures. The application of proposed distributed MPC algorithm for control of zone temperatures may lead to reduction of energy consumption and improvement of product quality. In this chapter, we present the furnaces for continuous annealing, continuous galvanizing and the hot blast stoves in which we want to use the MPC strategy to improve heating quality and reduce energy consumption.

8.1 Furnaces in continuous annealing and galvanizing process

8.1.1 Continuous annealing

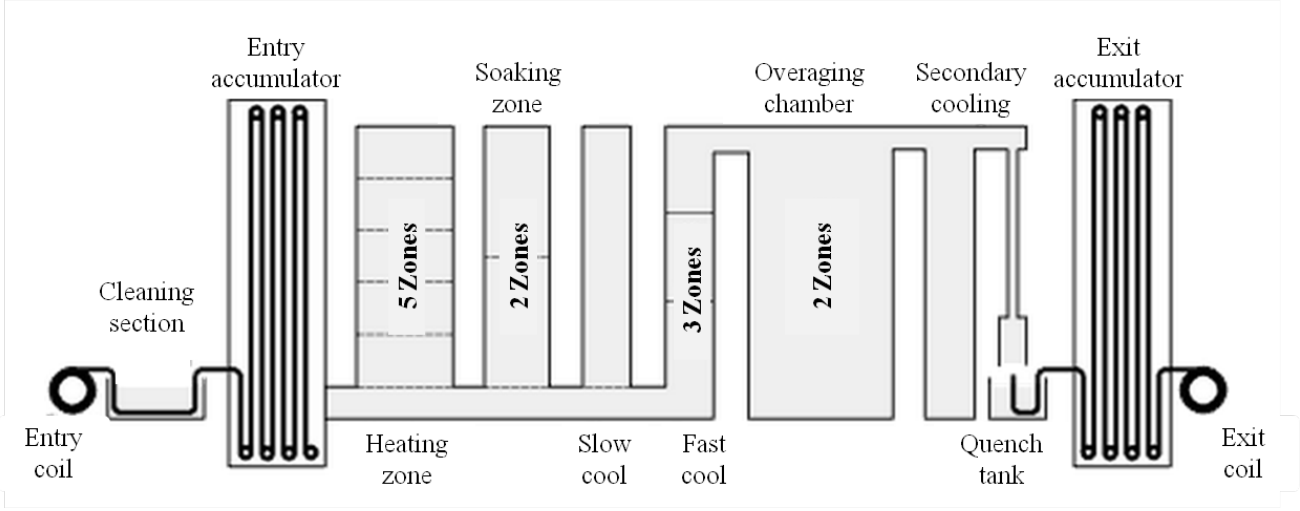

Figure 8.1: Continuous annealing process

Continuous annealing is the process of heating rolled strip in a sequence of zones to a predefined temperature according to steel grade and strip dimensions. The final result of the process is an increased of ductility and elimination of strains in the strip that lead to failure in subsequent processes.

Figure 8.1 shows the steps of continuous annealing process. Because the total processed strip can be several kilometers long, the furnace is vertical, withe the strip making several traverses within each zone. Firstly, the strip goes through cleaning section which is dedicated to removal of all dirt, grease, mill scale, and other oxides form on the surface of steel. Accumulators act as buffers between static steel coils (at entry and exit) and the continuous strip running through the furnace. As the entry coil is emptied and is replaced with a new full coil and both strips welded together, the entry accumulator unwinds to provide continuous strip. Similarly, the exit accumulator winds up while the exit coil is full and is unloaded at the exit end.

The annealing furnace is divided in different stages of heating and cooling as shown in Figure 8.1. These stages are necessary to give steel the desired properties by heating to particular temperatures and profiles thereby determining the grain structure within the metal, and to prepare for further processing such as galvanizing. In the heating zone, the cold strip is heated to the highest temperature of the annealing profile. Because of the danger of oxidation and flame length hot spots to the strip, the heat is radiated to the strip using ceramic radiant tubes, themselves fired by gas.

The soaking zone is dedicated to maintain strip temperature at 700 °C to 800 °C. and is electrically heated.

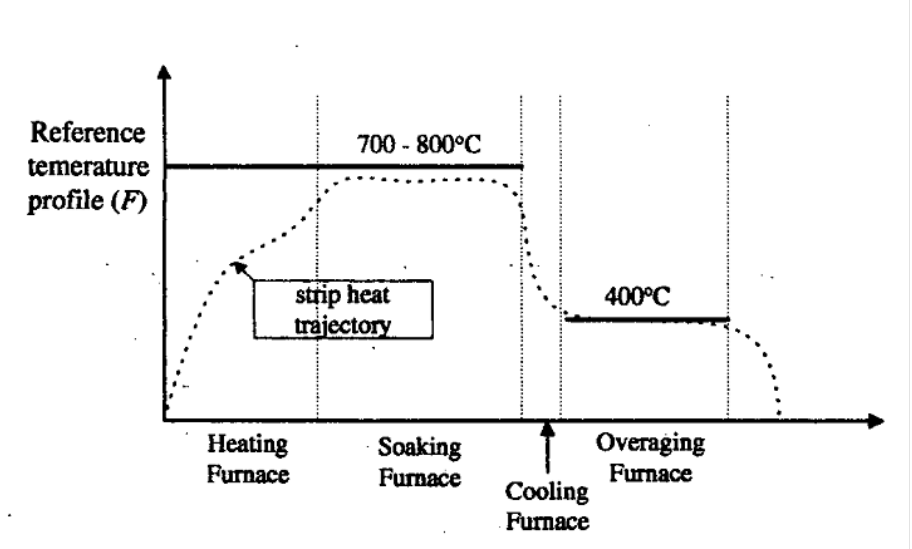

Figure 8.2: Temperature trajectory of the strip through different zones

The first of the cooling section is the slow cooling chamber, where two variable speed blowers circulate the atmosphere gas through two water-cooled heat exchangers. The second of the primary cooling sections is the fast cooling chamber, where the strip is cooled by a variable speed fan and the cooling length is determined by the positions of three dampers along its vertical length. This chamber is also split vertically for temperature profile control across the width of the strip using a scanning pyrometer.

The over-aging chamber provides electrical heating to maintain the strip at an intermediate temperature at around 400 °C. The temperature trajectory of the strip is shown in Figure 8.2 [CKC00].

It is desirable for productivity to maintain strip movement through the furnaces at all times. However, it is inevitable that faults occur and the strip stops. At this stage, there is a risk of the strip being overheated and breaking, causing major disruption. The control system, therefore, looks ahead and takes action at low line speed, line deceleration and line stop. These essential actions are to minimise the heating furnace firing and maintain atmosphere gas pressure due to the cooling strip.

To prevent oxidation of the strip, the atmosphere around the strip inside the furnaces is a controlled mixture of H2 and N2 (cracked ammonia NH3 providing 5% H2 and 95% N2) although hydrogen only is sometimes used. The furnace themselves are sealed gas tight and each of them is maintained under slight positive pressure.

8.1.2 Continuous galvanizing

Galvanizing is the practice of immersing clean, oxide-free iron or steel into molten zinc in order to form a zinc coating that is metallurgically bonded to the iron or steel's surface. The zinc coating protects the surface against corrosion by : 1) shielding the base metal from the atmosphere; 2) providing a cathodic or sacrificial protection since zinc is more electropositive than iron or steel.

In fact, the main difference of galvanizing process with annealing process is that it have a step of galvanizing with zinc pot as shown in Figure 8.3.

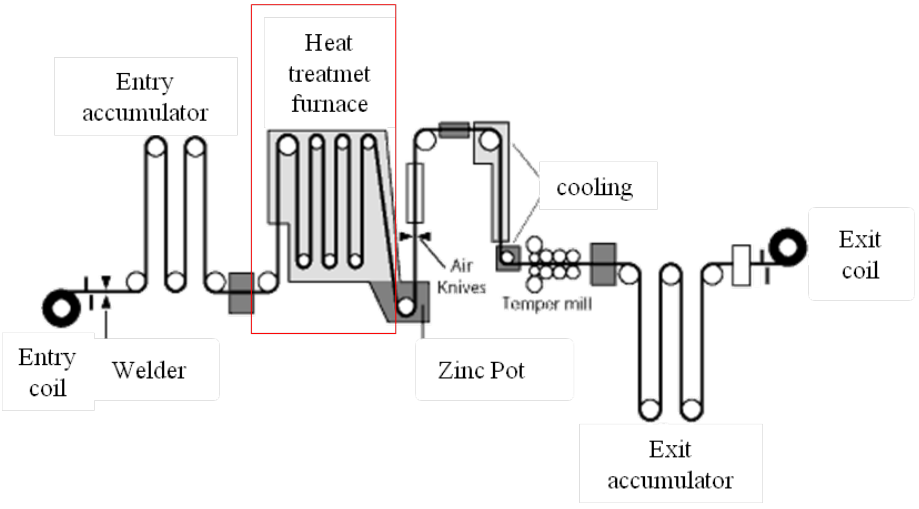

Figure 8.3: Continuous galvanizing process

The contiuous galvanizing processes a different type of product compared to that of continuous annealing process. The final products have more value-added than that of continuous annealing process. The strip is heated in the furnace and then immersed in zinc bath. The final step consists of quenching, removing excess zinc and

inspection. The major control involves Strip temperature, bath immersion time, bath withdrawal rate and steel cooling rate, which is more exigent than the control in furnace of continuous annealing process.

8.1.3 Furnace control

The control system of furnaces in the continuous galvanizing and annealing is quite similar to that of the slab reheating furnace. There are two main levels. The level 1 controls the temperature of the zones by varying the fuel flow rates, and the level 2 calculate the temperature set-points of furnace zones in order to achieve desired temperature of strip. The furnace zone temperatures are usually controlled by PID controller as depicted in Figure 8.4. Temperature of furnace zones are coupled with each other. In furnaces of annealing and galvanizing process,

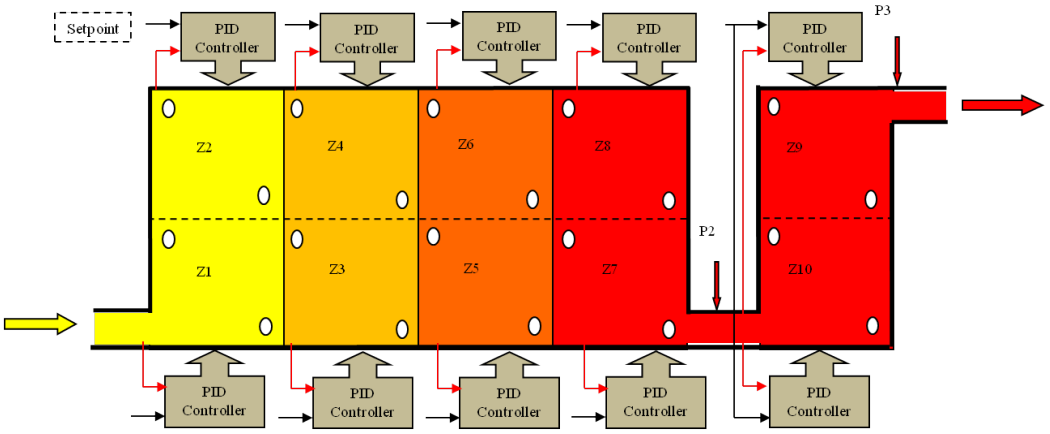

Figure 8.4: Temperature control of furnace with PID controllers

number of controllable zones are greater than that of reheating furnaces. Another important difference is that in continuous galvanizing and annealing radiant tubes are used as indirect heat sources while reheating furnaces use gas burners which is direct heat sources. In addition, the gaseous environment inside furnace of annealing and galvanizing is highly controlled and monitored to avoid surface reaction of steel strip and there is no circulation of gas. Therefore, the coupling effects between zones are not due to gas circulation but mainly to the flow of hot strip.

The control challenge of furnaces annealing and galvanizing is that it requires high control precision of strip temperature since temperature error constraints can be very tight (5°C). On the other hand, the strip is transported through the furnace at high speed (500 m/s). As a result, the dynamic of these furnaces is much faster than that of the reheating furnaces.

In order to successfully apply the proposed hierarchical MPC design in the galvanization furnaces, one must developed highly accurate coupling model of the furnace zones. The sampling period must be smaller than that utilized in reheating furnaces.

8.2 Hot blast stoves

8.2.1 Description

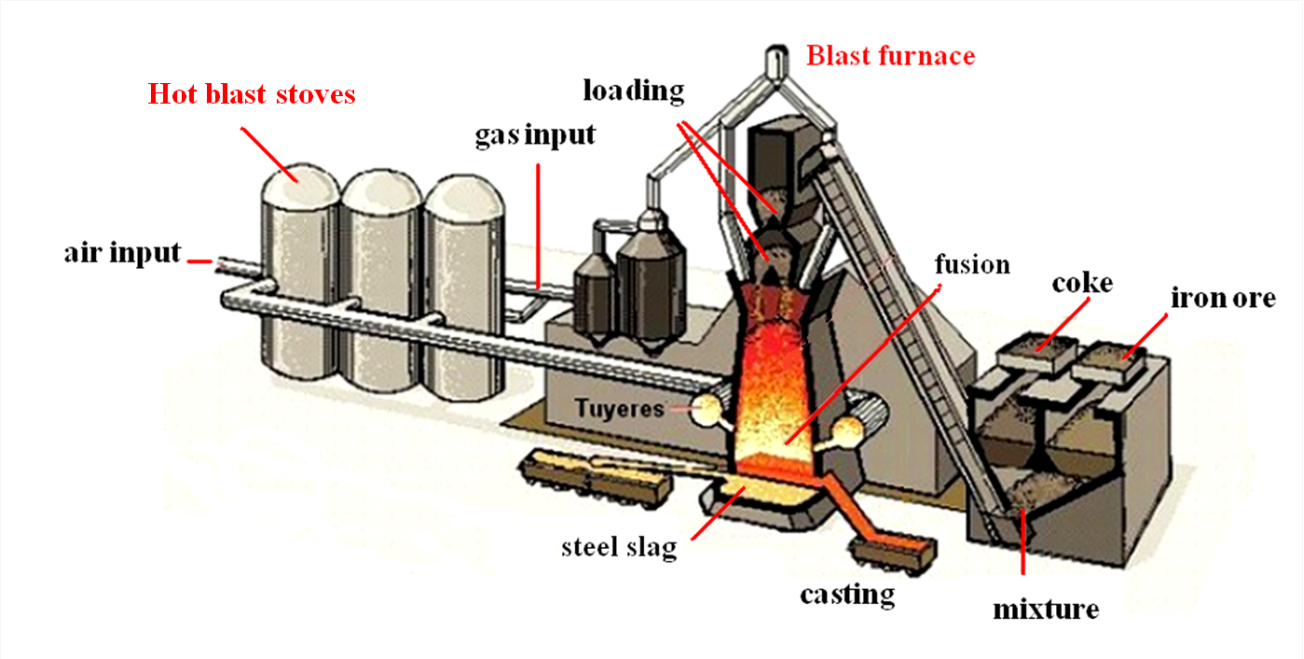

Figure 8.5: Hot blast stoves with blast furnace

Hot blast stoves are utilized for preheating of air blown into a blast furnace as shown in Figure 8.5. In the figure we can see that coke and iron ore are grinded and mixed together. The mixture of coal and iron ore is then transfered and loaded into blast furnace from the upper part. Hot combustion air (enriched with oxygen), also called blast air, is supplied to the blast furnace from the lower part at high pressure and at an averaged temperature of 1200 °C. To do the task the hot blast stoves firstly burn a gas mixture to generate the necessary heat. Then, the heat generated from combustion process is stocked in a solid structure called the checkerwork by heat transfer as shown in Figure 8.6. The latter solid structure is constructed from different layers of refractory brick of different types. One typical setup of different type of refractory bricks from top to bottom is mullite and silica, super-duty and high-allumia. These layers contain several flues so that air or gas can flow through. After the heat is stocked in the solid mass, the cold combustion air needed for blast furnace is now blown into the hot blast stoves and then heated by the hot solid mass. Therefore, a hot blast stove functions through two alternate cycles of heating and cooling, or also called gas phase and blast phase referring to periods when the stove is filled with gas and blast air, respectively. Specifically, these phases can be described as follows:

- Gas phase: the flow of gas in this phase is shown in Figure 8.6. The stove is heated by combustion of fuel gas in the combustion chamber of the stove. The wast gas which is the combustion product passes through the dome (top) of the stove and then goes down through the refractory bricks. At the end of the gas phase, most energy of combustion has been absorbed by the refractory layers. Therefore, temperature of waste gas is kept at low level which is around 300 °C.

- Blast phase: in this phase, cold blast air is blown into the stove, the path of blast air is in the opposite direction of waste gas in gas phase. Cold blast air firstly is heated by the hot refractory layers at the bottom of the stove. Subsequently, it goes up to the dome of the stove then delivered to the blast furnace as hot blast air of 1200 °C.

After gas phase, there is a changeover phase which is about five minutes long. The main task in this phase is to clean the impurities and dust produced from the waste gas. For that, a small flow of cold wind is blown into the combustion chamber. After the blast phase, other changeover phases are used to rebalance the pressure inside the stove, especially inside the refractory brick structure. Temperatures of the refractory layers have to be restricted

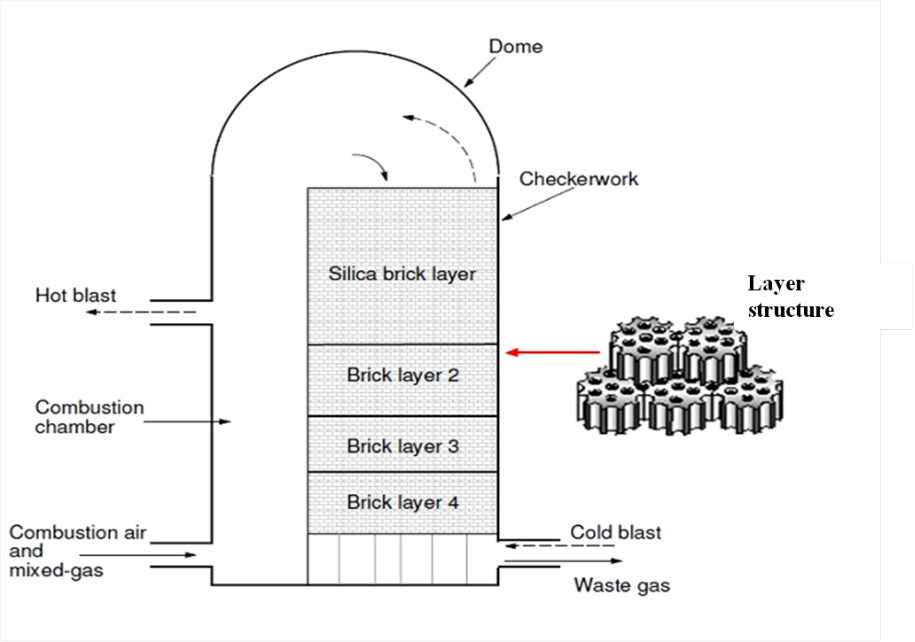

Figure 8.6: Internal structure of hot blast stoves

in different levels due to the specific characteristic of each type of material. From the top of the stove, high temperature material is used and at the bottom only low temperature material is needed. In Figure 8.7, we can see the temperature of the fluids (blast air or waste gas) and the solid (refractory layers) at different levels of a hot blast stove. At bottom of the stove, in the blast phases, we see that fluid temperature is dropped vertically because cold air is suddenly blown into the stove. Then temperature of the solid responds to fluid temperature as a first order transfer function. In the gas phase, gas temperature increases proportionally with the solid temperature. The difference between temperature of fluid and solid is reduced from the bottom to the top of the stove. One hot blast stove can only blow hot blast during its blast phase whereas the blast furnace needs a constant supply of hot blast. So several stoves must run in dephased conditions to guarantee a constant hot blast supply to the blast furnace. The minimum number of stove to ensure it is two but generally the 3 stoves configuration is the most utilized. For this configuration there is always one stove on blast phase while the two others are on gas or on changeover phase as shown in Figure 8.8.

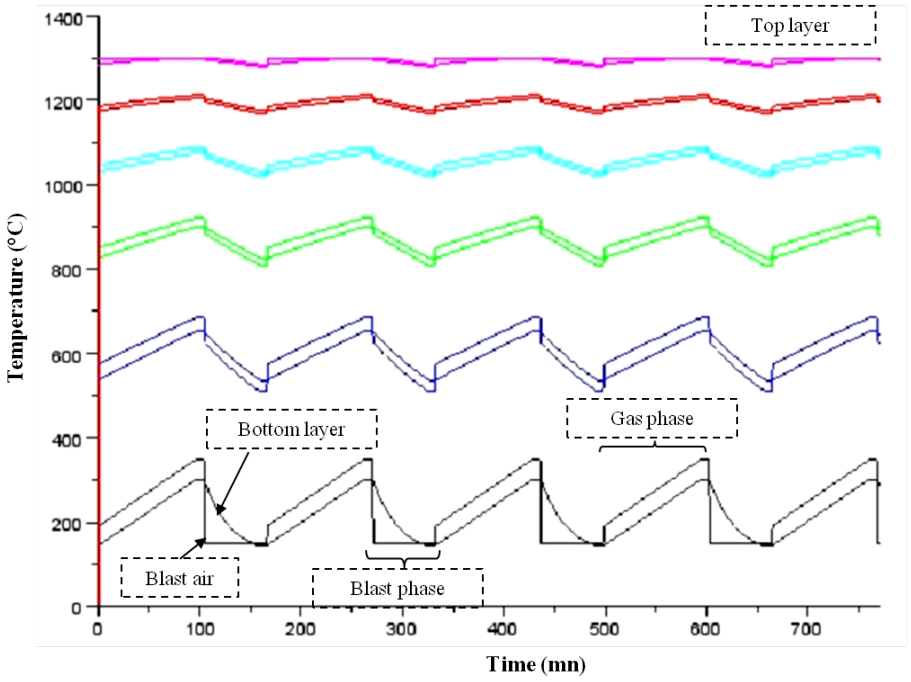

Figure 8.7: Temperature of the fluids (blast air, waste gas) and the solid (refractory layers) at different levels of a hot blast stove

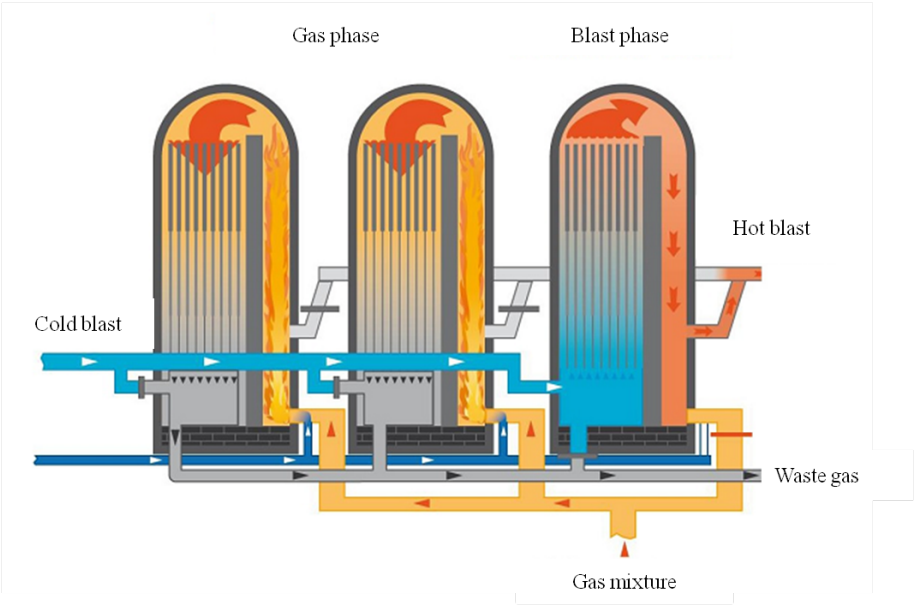

Figure 8.8: Hot blast stoves

8.2.2 Control task

The control task of hot blast stove is to determine the minimum amount of fuel necessary during gas phase to achieve the blast air flow rate and temperature requirements for the next blast phase. It is wanted that at the end of the blast phase, cold blast air flow is zero. However, in practice, the later flow can not precisely reach zero value, so a small amount of additional heat is put into the stove to ensure that the temperature and flow rate requirements of the hot blast are met. The additional heat can be measured by the by-pass valve position at the end of the blast phase.

Control constraints will be firstly the upper and lower bounds of fuel flow rates at the gas phase. Then, upper and lower temperature limits of the dome, the interface and the bottom of the stoves. These latter values are given in Table 8.1. Finally, a maximum allowed value of final by-pass valve position at the end of the blast phase is imposed.

Location	Maximum constraints	Minimum constraints
Dome	1500 °C	900 °C
Interface	1200 °C	800 °C
Bottom	350 °C	200 °C

Table 8.1: Temperature constraints of hot blast stove

As the fuel consumption of the hot stoves represents from 10% to 15% of the total energy consumption of a steel plant, the economical and environmental challenges for control are considerable. The MPC strategy can be considered in order to optimize the fuel flow rate profile of the hot stoves during the gas phase. The control is located at high level of the system and therefore is similar to the level 2 of the slab reheating furnace, because what are needed to be optimized are the fuel flow rate set-points given to subordinate levels. Currently, elaborated numerical models of hot stoves have been developed by ArcelorMittal R&D. These models can be used for the predictions of different process variables. Based on these predictions, an objective function can be formulated and computed taking into account the fuel consumption, constraint of fuel flow rates, the temperature of the delivered hot blast, temperature and constraints of different part of the stoves. Then optimal profiles of fuel flow rates in gas phases of the stoves are calculated via an appropriate optimization algorithm. One particular point is that the stoves function in cycles, the ratio between gas phase and blast phase can also be an important variable affecting the performance of the stoves which can be considered as input to the optimization problem.

Chapter 9

Conclusions and perspectives

9.1 Summary

The objectives of this work were to develop the MPC strategy under a hierarchical distributed architecture of a slab walking-beam reheating furnace. The function of slab reheating furnaces is to bring temperature of steel slabs from an initial temperature to around 1250 °C with a high homogeneity before entering hot rolling mills. The modeling of the furnace is crucial for observation of process variables such as: slab temperature profile, wall temperature, etc., and for the temperature control of the furnace. A non-steady-state model replaces the steady-state model of the furnace. The new model can cope with the increasing diversification of slabs in terms of desired final temperature, dimensions, material properties, etc. The non-steady-state model is validated against industrial data. This model is then utilized in the design of furnace control.

The control task of reheating furnaces concerns mainly two levels. The level 1 controls furnace zone temperatures by varying the fuel flows supplied to the furnace zones. And the level 2 controls the final slab temperature by varying the zone temperature set-points.

The level 1 should accurately track the temperature set-points of furnace zones, and it should take into account other important control objectives as minimum energy consumption, and minimum operating costs. To solve the stated control problem, the distributed MPC strategy is applied based on a model of furnace zones. This later model takes into account the interaction between furnace zones. Two approaches are carried out to capture coupling effect between furnace zones. The first approach considers that temperature of one zone is a function of not only the heat input into that zone but also that of its neighboring zones. This leads to a so-called input coupling model. The second approach is more direct; it considers that the temperature of one zone is a function of the heat input into that zone and the temperature of its neighboring zones. This leads to a so-called output coupling model.

By observations in the real furnace, the temperature of furnace zones behaves as a first order transfer function with respect to input powers and zone temperatures. Based on the structure of first order transfer function and

the furnace data, least square method is utilized to identify the parameters of two models. These models are then validated against measurement of zone temperatures and zone heat powers. The results of the validations showed that output coupling model is better than input coupling model in capturing the dynamic of furnace temperatures. Therefore, the output coupling has been used for the subsequent design of distributed MPC.

A non-cooperative distributed MPC algorithm is constructed for networked system which has sub-systems interacting with each other through outputs. Local MPC controllers communicate with each other about their actions. Each local MPC controller optimizes only the performance of its corresponding sub-system knowing how others sub-systems will behave and their influences. In order to cooperate actions of local MPC controllers, a cooperative algorithm is designed for serially connected networked system. In this later algorithm, each local MPC controller try to optimize not only the performance of its own sub-system but also two other neighbors of it. The communication between controllers is increased compared to non-cooperative strategy. However, cooperative distributed strategy brings the overall system to a better optimum point. Two algorithms are simulated for a illustrative example showing their stability and convergence property.

Using a simulation on Matlab environment, the proposed distributed MPC algorithms have been tested in an example scenario taken from real furnace system. The simulation results obtained with a validated model of furnace zones allow a comparison with the performance of the PI controllers. In general, better tracking performance and lower energy consumption is observed for distributed MPC compared to PI controllers. Moreover, the proposed method is implemented in the level 1 of a real furnace. The results obtained from this application are compared with the results of the level 1 that has been previously used. The comparison showed that distributed MPC can save up to 3 % of energy consumption while assuring good tracking of zone temperature set-points. In addition, the oxygen ratio in the furnace is reduced, which leads to less scale formation and decarburization and therefore less material losses. The global gas demand fluctuation is reduced, which leads to more stable pressure of gas supply station. Especially, automatic mode usage is increased by 10 % in soaking zone, which means that furnace operator needs less manual intervention in furnace operation.

At level 2 of the furnace, zone temperature set-points are calculated so that slabs can reach their desired final temperatures. Computation of zone temperature set-points is crucial for the success of heating task as well as for improvement of energy efficiency of the furnace. This task is subject to heating requirement of slabs, slab temperature and zone temperature constraints, and fuel supply constraints. Provided with a predefined schedule of slabs, MPC strategy utilizes the presented non-steady-state model to predict slab temperatures and furnace consumption from a given control input sequence of zone temperature set-points. The heating objectives are formulated as minimization criterions, and the limits and bounds on temperatures are accounted as penalty terms. Therefore, the control problem of level 2 is constructed as a unconstrained minimization problem.

Particular attention should be drawn to the formulation of the objective functions. Slabs are taken into account with different weighting coefficients based on their priority. In general, slabs are accounted by their discharging order, or in other word according to their position inside the furnace, the more a slab is close to the furnace exit, the more it is important because there is less time to heat it. In addition, a slab with higher desired final temperature is more prioritized than another neighboring slab because that would avoid under-heating and slab rejection at

hot rolling mill. In the formulation of minimization criterion concerning temperature gradient of slabs, soaking zone is especially dedicated to homogenization of slab temperature, therefore, slabs located in soaking zone are more prioritized than those of other zones.

In term of optimization algorithm, direct search algorithms are studied because the optimization problem is based on simulation of the furnace. Numerical non-steady-state model provides more accurate estimation temperature variables but also demands more computational resource than the steady-state model. Using gradient-based optimization methods needs analytic form of cost function gradient or their estimation by finite differencing. Nelder-Mead algorithm is utilized due to its advantages like: simple implementation, fast decline of function value.

Using a simulation on Intel Fortran environment, the proposed MPC strategy and optimization algorithm for level 2 is tested in a scenario of real furnace system. In this scenario, 900 slabs are processed within 3 days with an average heating time of 180 min. As mentioned before, the sequencing of slabs is predefined with an average throughput rate of 281 ton/h. The simulation obtained with a validated simulator of the furnace allow comparison of MPC strategy and the existing strategy of level 2. The obtained results show that utilization of MPC controller can save 5 % of furnace energy consumption while achieving higher slab final temperature and slab temperature homogeneity.

A study aiming at maximization of furnace productivity is carried out. In this study, the sequencing of slabs and optimization of zone temperature set-points are considered at the same time. In addition to the cost function used in the optimization zone temperature set-points, the average time interval between discharging events is accounted. As a new control input beside from the zone temperature set-points, the average time interval between discharging events is subject to a lower bound due to constraint of downstream process, e.g., hot rolling mill.

Particular point in the design MPC for scheduling task is that the number of slab taken into account in each execution of optimization is constant whereas the prediction horizon is variable from one furnace simulation to another according to the time interval.

The simulation in Intel Fortran environment shows that furnace production rate can be increased with a cost on energy consumption. In addition, accuracy of final slab temperature and its homogeneity can be significantly improved. In practice, these results prove that MPC can be used to inform the furnace operator to adjust furnace throughput rate to instantaneous furnace capacity. This information will help the furnace operator to adjust the scheduling of slabs so that furnace can operate at its maximum rate.

9.2 Conclusions

The presented results for level 1 of the furnace demonstrate the feasibility and the capabilities of the proposed distributed MPC method. The output coupling model is able to capture the coupling effects between furnace zones with a system of first order transfer functions. Although expected to give a limited optimality compared

to cooperative algorithm, the non-cooperative algorithm is able to provide satisfactory results for the considered walking-beam reheating furnace.

The MPC designed for level 2 has been tested successfully with data of real system. The MPC controller shows more accurate and homogeneous final slab temperature profiles compared to the results of the existing controller. As a result, overheating of slab is reduced significantly and consequently the energy efficiency is enhanced. Despite highly expensive non-steady-state furnace model used in the optimization, the Nelder-Mead simplex algorithm is able to deliver the satisfactory temperature set-points in period of a sample time meaning the proposed MPC controller for level 2 can be implemented in the real furnace.

The slab scheduling can be improved with the MPC method. Instead of using a fixed prediction horizon, we use a variable one whereas number of slabs taken into account is constant. The choice of an average time interval of discharging events as the additional control input is proved to be relevant. The obtained results show that the throughput rate can be increased up to 15 ton/h while temperature heating requirement are improved, especially the homogeneity of slabs. The propose MPC controller will not actually serve as furnace controller of level 2 but to inform furnace operator to modify the slab scheduling so that the furnace can operate at its maximum capacity.

9.3 Perspectives

The proposed distributed MPC strategy has been implemented and currently runs in full time in the furnaces of hot rolling mill of Florange plant ArcelorMittal. With the concluding results of MPC level 2 simulated based on the data of the real system, it is planned to install the MPC controller on the considered furnace for experimental validation. On the same hot rolling mill there is another reheating furnace having a similar design but different dimensions of zones. Adaptation of the solution to this new furnace will only involve the parameter modification of the non-steady-state model of the furnace.

In the proposed solution, the non-steady-state model used for furnace simulation and optimization of temperature setpoints is quite time-consuming. This restricts the optimality of the solution. Therefore, we want to use another model which is less time-consuming but is able to capture the transient of the furnace operation. In the presented heat transfer model, around 90% of calculation time is dedicated to radiative heat transfer. Therefore, the simplification of model will involve mainly the calculation of radiative heat transfer in the furnace. Other type of heat transfer like conduction or convection will remains the same as in the developed non-steady state model.

On the other hand, reducing the calculation time of the model will permit the utilization of gradient-based methods which can lead to faster convergence of the optimization problem. For that, the objective function will need to be differentiated by finite-differencing.

In addition, simplified model will give more margin on real-time constraints. The control horizon can be increased, which means that longer period of temperature setpoints can be planned at each execution of optimization.

Robustness of the MPC controller level 2 in response to the uncertainty in the model is also and important direction that should be exploited. In practice, the uncertainty can come from unscheduled stop of the furnace which is due to furnace incidents. The recurrent changing of process data should be also analyzed more in simulation studies. These process data could involve degradation of furnace wall material, change of cooling systems, etc.

In the development of MPC for level 2, we have used constant limits of zone temperatures as constraints in the optimization problems. In real-time implementation, these limits will actually vary over time. At a specific time instant, the limits of zone temperatures depends not only on power available of furnace zones but also on the slab charge located in each zone. To tackle this issue, the furnace model may effectively give estimations of these limits which might then be utilized in the optimization problems.

The large perspective of this work is the implementation of MPC level 2 for all reheating furnaces of ArcelorMittal steel plants, which is expected to generate significant economical and environmental impact.

Other type of reheating furnaces which are dedicated to different type of products such as blooms, billets, tubes, etc., may be studied in order to apply distributed MPC and hierarchical MPC.

Appendix A

BISRA Table

Temperature (K)	Conductivity	Kirchoff coefficient	Enthalpy	index
1400	0.0760	130.75	226.30	1
1350	0.0740	127.00	218.30	2
1300	0.0730	123.32	210.30	3
1250	0.0720	119.70	202.30	4
1200	0.0710	116.13	194.30	4
1150	0.0700	112.60	186.30	6
1100	0.0680	109.15	178.30	7
1050	0.0670	105.78	170.30	8
1000	0.0660	102.45	162.30	9
950	0.0650	99.18	154.30	10
900	0.0650	95.93	146.40	11
850	0.0660	92.65	136.40	12
800	0.0720	89.20	126.80	13
750	0.0760	85.50	116.40	14
700	0.0810	81.58	103.20	15
650	0.0860	77.40	92.90	16
600	0.0900	73.00	83.30	17
550	0.0940	68.40	74.30	18
500	0.0980	63.60	65.90	19
450	0.1040	58.55	58.00	20
400	0.1110	53.18	50.50	21
350	0.1160	47.50	43.40	22
300	0.1220	41.55	36.60	23
250	0.1270	35.33	30.00	24
200	0.1330	28.83	23.60	25
150	0.1380	22.05	17.40	26
100	0.1440	15.00	11.40	27
50	0.1500	7.65	5.60	28
0	0.1560	0.00	0.00	29

Appendix B

Coordinate search algorithm

Let consider the unconstrained minimization problem of a continuously differentiable function

$$\min_{x \in \Re^n} f(x) \tag{B.1}$$

$$\text{where } f(x) : \Re^n \to R. \tag{B.2}$$

Coordinate search is a pattern search method. To define the pattern for coordinate search, the basic matrix B is usually the identity matrix, i.e., $B = I_n$. For example, with $n = 2$, we have

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{B.3}$$

The generating matrix for each iterate k is fixed as $C_k = C_0$ which contains in its columns all possible combination of $-1, 0, 1$. Thus, C_k has $p = 3^n$ columns and contains both I_n and $-I_n$. Matrix M can be defined as $M = I_n$, and L consists of the remaining $3^n - 2n$ columns of C. For example, with $n = 2$,

$$C_k = C_0 = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix} = [M \ -M \ L],$$

with

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and

$$L = \begin{bmatrix} 1 & 1 & -1 & -1 & 0 \\ 1 & -1 & -1 & 1 & 0 \end{bmatrix}.$$

All possible trial points defined by the pattern matrix $P = BC_0$, for a given step length Δ_k are depicted in Figure B.1.

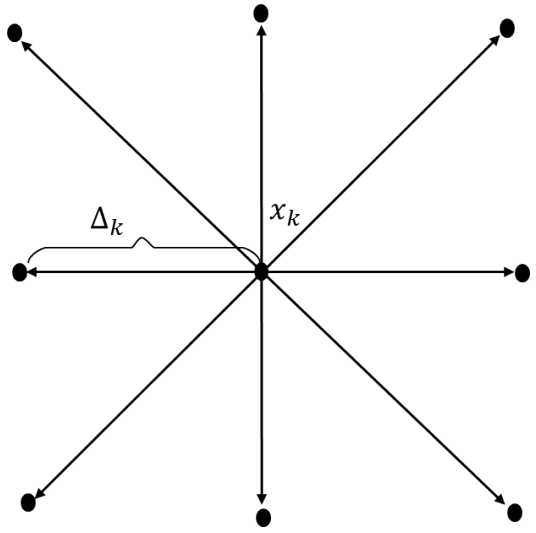

Figure B.1: Trial points of coordinate search in $\Re^2$ for a given step length Δ_k.

To guide the search for descent point, the algorithm of explanatory moves is given as following algorithm where e_i, $i = 1, ..., n$, denote the unit coordinate vectors:

Explanatory moves algorithm for coordinate search.

The current iterate is given with x_k, Δ_k, $f(x_k)$, and B, set $s_k = 0$, $\epsilon_k = 0$, and $min = f(x_k)$.

For $i = 1, ..., n$ do

- (a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
- (b) If $f(x_k^i) < min$ then $\epsilon_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$ and $s_k = s_k^i$.
  Otherwise
  - (i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
  - (ii) If $f(x_k^i) < min$ then $\epsilon_k = f(x_k) - f(x_k^i)$, $min = f(x_k^i)$ and $s_k = s_k^i$.

Returns

In Figure B.2, all possible subsets of trial points that the coordinate search may conduct is shown. Solid circles indicate successful intermediate trial points, i.e., $f(x_k^i) < f(x_k)$. While open circles indicates unsuccessful intermediate trial points, i.e., $f(x_k^i) \geq f(x_k)$. For iteration k, explanatory moves may compute no less than n trial point as the first case (on the top left), and no more than $2n$ points as the last case (on the bottom right). The updating of step length Δ_k is determined by

- If $\epsilon_k \leq 0$ then $\Delta_{k+1} = 0.5\Delta_k$.
- If $\epsilon_k > 0$ then $\Delta_{k+1} = \Delta_k$.

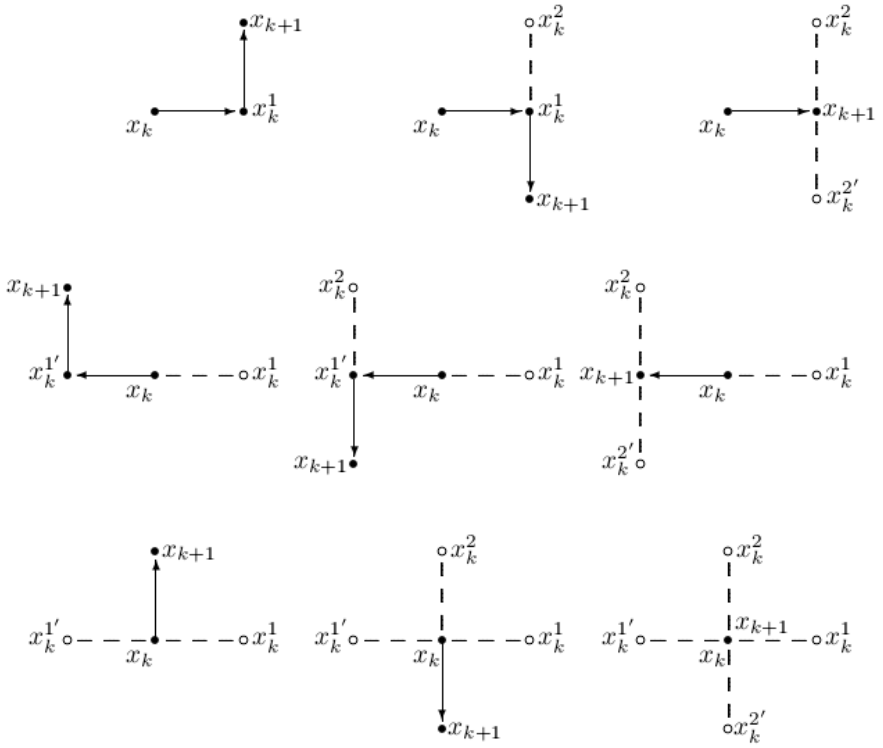

Figure B.2: All possible trial steps for coordinate search in $\Re^2$;$\circ : f(x_k^i) \geq f(x_k)$;$\bullet : f(x_k^i) < f(x_k)$

Appendix C

Nelder-Mead simplex search algorithm

The simplex algorithm follows the following steps:

- Step 1-Initialization
  The initial simplex in $n - dimensional$ space is defined by $(n+1)$ points $\{P_0, P_1, ..., P_n\}$. Evaluation of function value at each point of the simplex $y_i = J_o(P_i)$ for $i = \overline{0,n}$. $y_h = J_o(P_h)$ and $y_l = J_o(P_l)$ represent respectedly highest and lowest value of cost function. $\overline{P}$ is centroid point of the simplex, $[P_i P_j]$ stand for distance from P_i to P_j

- Step 2- Reflection operation
  The centroid point $\overline{P}$ is reflected to a P^* with coordinates determined as following.

$$P^* = (1+\alpha)\overline{P} - \alpha P_h \tag{C.1}$$

  where α is a positive constant, so called reflection coefficient. If $P_l < y^* = J_o(P^*) < P_h$ then P_h is replaced by P^* and the algorithm restarts with new simplex. If $y^* \leq y_l$, go to step 3, if $y^* \geq y_h$, go to step 4.

- Step 3- Expansion operation
  P^* is expanded to P^{**} by a relation

$$P^{**} = \gamma P^* + (1-\gamma)\overline{P}. \tag{C.2}$$

  If $y^{**} < y_l$, P_h is replaced by P^{**} and the algorithm restarts. If $y^{**} > y_l$, P_h is replaced by P^* before restarting.

- Step 4- Contraction operation
  P_h is replaced by P^*. Contracted point P^{**} is determined by

$$P^{**} = \beta P_h + (1-\beta)\overline{P}. \tag{C.3}$$

Where β is contraction coefficient. If $y^{**} \leq y_h$ then P_h is replaced by P^{**} and the algorithm restarts. In the contrary, new simplex is replaced by

$$\begin{aligned} P_i &= \tfrac{P_i + P_l}{2} \\ i &= 1, n \end{aligned} \tag{C.4}$$

If the convergence happens then stop, otherwise go to step 1 and the algorithm restarts

The convergence criterion used to halt the procedure is calculated as $\sqrt{\{\sum (y_i - \overline{y}\}^2 / n}$. As in [LLR04], the simplex of size a can be initialized at P_0 based on the following rule

$$\begin{aligned} P_i &= P_0 + \rho e_i + \textstyle\sum_{k=1, k \neq i}^{n} (q e_k) \\ i &= 1, n \end{aligned} \tag{C.5}$$

where e_i is a unit base vector and

$$\begin{aligned} \rho &= \tfrac{a}{n\sqrt{2}} \left(\sqrt{n+1} + n - 1 \right) \\ q &= \tfrac{a}{n\sqrt{2}} \left(\sqrt{n+1} - 1 \right). \end{aligned} \tag{C.6}$$

And a is taken between 2% and 10% of the smallest domain dimension.

Bibliography

[ACK95] I. Assaf, M. Chen, and J. Katzberg. Steel scheduling optimization for ipsco's rolling mill and reheat furnace. In *WESCANEX 95. Communications, Power, and Computing. Conference Proceedings., IEEE*, volume 2, pages 294--299. IEEE, 1995.

[AGBE11] W. Al-Gherwi, H. Budman, and A. Elkamel. A robust distributed model predictive control algorithm. *Journal of Process Control*, 21(8):1127--1137, 2011.

[ALRS05] D. Auzinger, M. Lichtenwagner, M. Rosenthaler, and A. H. Schutti. Process for the optimized operation of a reheating furnace, March 23 2005. EP Patent App. EP20,040,020,371.

[Bar] P. V. Barr. Examining reheating furnace thermal response to mill delays. In *Proceedings of Materials Science and Technology 2003, Chicago, Illinois, USA*.

[BBG08] L. Balbis, J. Balderud, and M. J. Grimble. Nonlinear predictive control of steel slab reheating furnace. In *American Control Conference, 2008*, pages 1679--1684. IEEE, 2008.

[BDM99] K. J. Burnham, A. Dunoyer, and S. Marcroft. Bilinear controller with pid structure. *Computing & Control Engineering Journal*, 10(2):63--69, 1999.

[BML07] J. S. Broughton, M. Mahfouf, and D. A. Linkens. A paradigm for the scheduling of a continuous walking beam reheat furnace using a modified genetic algorithm. *Materials and manufacturing processes*, 22(5):607--614, 2007.

[BRW00] G. G. Botte, J. A. Ritter, and R. E. White. Comparison of finite difference and control volume methods for solving differential equations. *Computers & Chemical Engineering*, 24(12):2633--2654, 2000.

[BSSP04] S. Banerjee, D. Sanyal, S. Sen, and I. K. Puri. A methodology to control direct-fired furnaces. *International journal of heat and mass transfer*, 47(24):5247--5256, 2004.

[CA08] S. Chen and S. Abraham. Modification of reheat furnace practices through comprehensive process modeling. *Iron & steel technology*, 5(8):66--79, 2008.

[CCL05] W. H. Chen, Y. C. Chung, and J. L. Liu. Analysis on energy consumption and performance of reheating furnaces in a hot strip mill. *International Communications in Heat and Mass Transfer*, 32(5):695--706, 2005.

[CJKT02] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *Control Systems, IEEE*, 22(1):44--52, 2002.

[CKC00] Young C Cho, Wook Hyun Kwon, and Christos G Cassandras. Optimal control for steel annealing processes as hybrid systems. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, pages 540--545. IEEE, 2000.

[CMT87] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control—part i. the basic algorithm. *Automatica*, 23(2):137 -- 148, 1987.

[CXZ+03] Z. Chen, C. Xu, B. Zhang, H. Shao, and J. Zhang. Advanced control of walking-beam reheating furnace. *Journal of University of Science and Technology Beijing*, 2003.

[DBS88] D. Dewar, K. Brown, and D. Smith. Computerized reheating furnace control at bsc, ravenscraig works. In *SCANHEATING D-Proceedings of the 2nd Iote˜ational Conference on Heatin˜ Inclyding Detection and Conditioning of Surface Defects on Hot Material. Hot Charging and Hot Direct Rolling, LuleA, Sweden*, 1988.

[DCM+92] B. Doss, E. Chu, H. Mason, R. Ruiz, I. Chan, J. Kleppe, and J. Jensen. Steel process furnace burner control using acoustic pyrometry. In *Proceedings of the International Gas Research Conference, Orlando, USA*, pages 2231--2240, 1992.

[DF01] G. Dutta and R. Fourer. A survey of mathematical programming applications in integrated steel plants. *Manufacturing & Service Operations Management*, 3(4):387--400, 2001.

[DP] Y. Djoko and R. Piernot. Control of slab reheating furnace.

[ESYS97] H. Ezure, Y. Seki, N. Yamaguchi, and H. Shinonaga. Development of a simulator to calculate an optimal slab heating pattern for reheat furnaces. *Electrical engineering in Japan*, 120(3):42--53, 1997.

[FBF+83] P. Fontana, A. Boggiano, A. Furinghetti, G. Cabras, and C. Simoncini. An advanced computer control system for reheat furnaces. *Iron Steel Eng.*, 60(8):55--62, 1983.

[FPSF90] G. Facco, M. E. Petersen, R. J. Schurko, and N. T. Ferguson. State-of-the-art slab reheating furnaces at dofasco. *Iron Steel Eng.*, 67(1):27--36, 1990.

[FPTV92] Brian P Flannery, Wiliam H Press, Saul A Teukolsky, and William Vetterling. Numerical recipes in c. *Press Syndicate of the University of Cambridge, New York*, 24, 1992.

[FRLD06a] L. Ferrand, P. Reynes, and F. Le Duigou. Simulation tools make new furnace technology. *Revue de Métallurgie*, 103:67--75, 2 2006.

[FRLD06b] L. Ferrand, P. Reynes, and F. Le Duigou. Simulation tools make new furnace technology. *Revue de Métallurgie*, 103(02):67--75, 2006.

[FUK+10] S. Fujii, K. Urayama, K. Kashima, J. I. Imura, T. Kurokawa, and S. Adachi. Machine-based modeling of conveyor-type flowshops with application to scheduling and temperature control in slab reheating furnace. In *Control Applications (CCA), 2010 IEEE International Conference on*, pages 2059--2064. IEEE, 2010.

[GGS01] G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control system design*, volume 240. Prentice Hall New Jersey, 2001.

[GSR+98a] D. F. Garcia, M. Sierra, R. Rodriguez, A. Campes, R. Diaz, F. Obese, and J. A. Gonzalez. Development of scalable real-time observers for continuous reheating furnaces based on mathematical modeling techniques. In *Industry Applications Conference, 1998. Thirty-Third IAS Annual Meeting. The 1998 IEEE*, volume 3, pages 2207--2216, Oct 1998.

[GSR+98b] D. F. Garcia, M. Sierra, R. Rodriguez, A. Campes, R. Diaz, F. Obese, and J. A. Gonzalez. Development of scalable real-time observers for continuous reheating furnaces based on mathematical modeling techniques. In *Industry Applications Conference, 1998. Thirty-Third IAS Annual Meeting. The 1998 IEEE*, volume 3, pages 2207--2216. IEEE, 1998.

[HBK09] S. H. Han, S. W. Baek, and M. Y. Kim. Transient radiative heating characteristics of slabs in a walking beam type reheating furnace. *International Journal of Heat and Mass Transfer*, 52(3):1005--1011, 2009.

[HCK10a] S. H. Han, D. Chang, and C. Y. Kim. A numerical analysis of slab heating characteristics in a walking beam type reheating furnace. *International Journal of Heat and Mass Transfer*, 53(19–20):3855 -- 3861, 2010.

[HCK10b] S. H. Han, D. Chang, and C. Y. Kim. A numerical analysis of slab heating characteristics in a walking beam type reheating furnace. *International Journal of Heat and Mass Transfer*, 53(19):3855--3861, 2010.

[HHLW08] C. T. Hsieh, M. J. Huang, S. T. Lee, and C. H. Wang. Numerical modeling of a walking-beam-type slab reheating furnace. *Numerical Heat Transfer, Part A: Applications*, 53(9):966--981, 2008.

[HJ61] R. Hooke and T. A. Jeeves. ``direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212--229, 1961.

[HVS04] M. Honner, Z. Vesely, and M. Svantner. Exodus stochastic method application in the continuous reheating furnace control system. *Scandinavian journal of metallurgy*, 33(6):328--337, 2004.

[HZ82] F. Hollander and S. P. A. Zuurbier. Design, development and performance of online computer control in 3-zone reheating furnace. *Iron and Steel Engineer*, 1982.

[Inc11] Frank P. Incropera. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[IZS04] Z. A. Icev, J. Zhao, and M. J. Stankovski. Supervisory-plus-regulatory control design for efficient operation of industrial furnaces. 2004.

[Jij09] L. M. Jiji. *Heat Conduction*. Springer, 2009.

[JK07] Y. J. Jang and S. W. Kim. An estimation of a billet temperature during reheating furnace operation. *International journal of control automation and systems*, 5(1):43, 2007.

[JKZ05] A. Jaklič, T. Kolenko, and B. Zupančič. The influence of the space between the billets on the productivity of a continuous walking-beam furnace. *Applied thermal engineering*, 25(5):783--795, 2005.

[JLKK10] J. H. Jang, D. E. Lee, M. Y. Kim, and H. G. Kim. Investigation of the slab heating characteristics in a reheating furnace with the formation and growth of scale on the slab surface. *International Journal of Heat and Mass Transfer*, 53(19):4326--4332, 2010.

[JR] S. Joly and A. Raudin. Control of slab reheating furnace.

[KDW97] S. Keplinger, L. Doppelbauer, and K. Wallner. Automation system for reheating furnaces at hutta sendzimira, polland. *Methallurgical Plant and Technology International*, 1997.

[KHK00] J. G. Kim, K. Y. Huh, and I. T. Kim. Three-dimensional analysis of the walking-beam-type slab reheating furnace in hot strip mills. *Numerical Heat Transfer: Part A: Applications*, 38(6):589--609, 2000.

[Kim07] M. Y. Kim. A heat transfer model for the analysis of transient heating of the slab in a direct-fired walking beam type reheating furnace. *International Journal of Heat and Mass Transfer*, 50(19):3740--3748, 2007.

[KKY+00] H. S. Ko, J. S. Kim, T. W. Yoon, M. Lim, D. R. Yang, and I. S. Jun. Modeling and predictive control of a reheating furnace. In *American Control Conference, 2000. Proceedings of the 2000*, volume 4, pages 2725--2729. IEEE, 2000.

[KLT03] T. G Kolda, R. M Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385--482, 2003.

[KMP94] M. K. F. Knoop and J. A. Moreno Pérez. Nonlinear pi-controller design for a continuous-flow furnace via continuous gain scheduling. *Journal of Process Control*, 4(3):143--147, 1994.

[KSB10] D. H. Kang, Lorente S., and A. Bejan. Constructal architecture for heating a stream by convection. *International Journal of heat and Mass Transfer*, 2010.

[KvD94] A. Kusters and G. A. J. M. van Ditzhuijzen. Mimo system identification of a slab reheating furnace. In *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pages 1557--1563. IEEE, 1994.

[LCG98] L. Lopez, M. W. Carter, and M. Gendreau. The hot strip mill production scheduling problem: A tabu search approach. *European Journal of Operational Research*, 106(2):317--335, 1998.

[LCoIE85] W. T. Lankford, United States Steel Corporation, Association of Iron, and Steel Engineers. *The Making, shaping, and treating of steel*. Association of Iron and Steel Engineers, 1985.

[Led86a] B. Leden. A control system for fuel optimization of reheating furnaces. *Scand. J. Metall.*, 15(1):16--24, 1986.

[Led86b] B. Leden. A program for temperature analysis in steel plants. *Scandinavian journal of metallurgy*, 15:215--223, 1986.

[Lev96] W. S. Levine. *The control handbook*. CRC press, 1996.

[LGC] U. Leifgen, S. Ganesaratnam, and L. Croce. A new concept for the control of reheating furnace for slabs. *Proceedings of the* 1^{st} *International Conference on Energy Efficiency and* CO_2 *Reduction in the Steel Industry, EECR STEEL, Düsseldorf, Germany*.

[Li86] Z. Li. *Computer simulation of the push-type slab reheating furnace*. PhD thesis, 1986.

[LLR04] M. A. Luersen and R. Le Riche. Globalized nelder-mead method for engineering optimization. *Computers and structures 82*, 2004.

[LR05] P. Laurinen and J. Röning. An adaptive neural network model for predicting the post roughing mill temperature of steel slabs in the reheating furnace. *Journal of materials processing technology*, 168(3):423--430, 2005.

[LRWW98] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder--mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112--147, 1998.

[LTT00a] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124(1):191--207, 2000.

[LTT00b] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics 124*, 2000.

[Mår92] A. Mårtensson. *Energy efficiency improvement by measurement and control: A case study of reheating furnaces in the steel industry*. Environmental and Energy Systems Studies, Lund University, Lund Institute of Technology, 1992.

[Mck98] K. I. M. Mckinnon. Convergence of the nelder-mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

[MN14] J. M. Maestre and R. R. Negenborn. *Distributed model predictive control made easy*. Springer, 2014.

[MPS04] P. Marino, A. Pignotti, and D. Solis. Control of pusher furnaces for steel slab reheating using a numerical model. *Latin American applied research*, 34(4):249--255, 2004.

[Nas51] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286--295, 1951.

[NM65] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308--313, 1965.

[Nob75] J. J. Noble. The zone method: explicit matrix relations for total exchange areas. *International Journal of Heat and Mass Transfer*, 18(2):261--269, 1975.

[NRAD+12a] X. M. Nguyen, P. Rodriguez-Ayerbe, D. Dumur, A. Mouchette, and F. Lawayeb. Distributed model predictive control of steel slab reheating furnace. In *Proceedings of the IFAC Workshop on Automation in the Mining, Mineral and Metal Industries*, Nagaragawa, Gifu, Japon, 2012.

[NRAD+12b] X. M. Nguyen, P. Rodriguez-Ayerbe, D. Dumur, A. Mouchette, and F. Lawayeb. Distributed Model Predictive Control of Steel Slab Reheating Furnace. In *Proceedings of the IFAC Workshop on Automation in the Mining, Mineral and Metal Industries*, pages CD--Rom, Nagaragawa, Gifu, Japon, 2012.

[NT02] L. Nazareth and P. Tseng. Gilding the lily: A variant of the nelder-mead algorithm based on golden-section search. *Computational Optimization and Applications*, 22(1):133--144, 2002.

[NvWS] E. Nederkoorn, P. van Wilgen, and J. Schuurmans. Nonlinear model predictive control of walking beam furnaces. *Proceedings of the* 1^{st} *International Conference on Energy Efficiency and* CO_2 *Reduction in the Steel Industry, EECR STEEL, Düsseldorf, Germany*.

[Pat80] S. Patankar. *Numerical heat transfer and fluid flow*. CRC Press, 1980.

[PCB02] C. J. Price, I. D. Coope, and D. Byatt. A convergent variant of the nelder--mead algorithm. *Journal of Optimization Theory and Applications*, 113(1):5--19, 2002.

[PD10] G. Poquet and A. Dujin. Pour les ménages, la recherche du confort prime encore, sur les économies dénergie. *Consomation et modes de vie*, 2010.

[PJC70] H. E. Pike Jr and S. J. Citron. Optimization studies of a slab reheating furnace. *Automatica*, 6(1):41--50, 1970.

[Pow64] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155--162, 1964.

[Pri80] J. C. Price. Temperature control for slab reheat furnaces. *Iron Steel Eng.*, 57(9):59--62, 1980.

[PSV92] C. M. Petersen, K. L. Sørensen, and R. V. V. Vidal. Inter-process synchronization in steel production. *The international journal of production research*, 30(6):1415--1425, 1992.

[PW98] L. M. Pedersen and B. Wittenmark. On the reheat furnace control problem. In *American Control Conference, 1998. Proceedings of the 1998*, volume 6, pages 3811--3815. IEEE, 1998.

[RB92] L. Rixin and N. Baolin. Mathematical model for dynamic operation and optimum control of pusher type slab reheating furnace. *Industrial Heating*, 59(3):60--62, 1992.

[Ros60] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175--184, 1960.

[RS08] J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of process control*, 18(9):839--845, 2008.

[RSCG86] J. L. Roth, H. Sierpinski, J. Chabanier, and J. M. Germe. Computer control of slab furnaces based on physical models. *Iron Steel Eng.*, 63(8):41--47, 1986.

[Sch96] K. E. Scholey. *A transient, three-dimensional, thermal model of a billet reheating furnace*. PhD thesis, University of British Columbia, 1996.

[SGW+11] A. Steinboeck, K. Graichen, D. Wild, T. Kiefer, and A. Kugi. Model-based trajectory planning, optimization, and open-loop control of a continuous slab reheating furnace. *Journal of Process Control*, 21(2):279--292, 2011.

[She94] F. Shenvar. Walking beam furnace supervisory control at inland's 80-in. hot strip mill. *Iron and Steel Engineer*, 1994.

[SHH62] W. G. R. F. R. Spendley, G. R. Hext, and F. R. Himsworth. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4(4):441--461, 1962.

[SII+99] S. I. Sugiyama, T. Ishii, M. Ishioka, Y. Hino, and T. Okawa. Schedule-free heating in preheating furnace. *NKK technical review*, (80):35--45, 1999.

[SK13] A. Steinboeck and A. Kugi. Optimized pacing of continuous reheating furnaces. In *Control Conference (ECC), 2013 European*, pages 4454--4459. IEEE, 2013.

[Soc] ASM Heat Trating Society. To achieve vision 2020: An r&d plan for the heat treating community.

[SS04] H. Sibarani and Y. Samyudia. Robust nonlinear slab temperature control design for an industrial reheating furnace. *Computer Aided Chemical Engineering*, 18:811--816, 2004.

[Sta04] D. Staalman. The funnel model for accurate slab temperature in reheating furnaces. *La revue de Métallurgie*, 2004.

[Ste11] A. Steinböck. *Model-based control and optimization of a continuous slab reheating furnace*, volume 12. steinboeck, 2011.

[STY99] Y. Sakamoto, Y. Tonooka, and Y. Yanagisawa. Estimation of energy consumption for each process in the japanese steel industry: a process analysis. *Energy conversion and management*, 40(11):1129--1140, 1999.

[SWHP87] R. J. Schurko, C. Weinstein, M. K. Hanne, and D. J. Pellecchia. Computer control of reheat furnace: A comparison of strategies and application. *Iron and Steel Engineer*, 1987.

[SWK11] A. Steinboeck, D. Wild, and A. Kugi. Feedback tracking control of continuous reheating furnaces. In *Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC), Milan, Italy*, pages 11744--11749, 2011.

[SWK13] A. Steinboeck, D. Wild, and A. Kugi. Nonlinear model predictive control of a continuous slab reheating furnace. *Control Engineering Practice*, 21(4):495--508, 2013.

[SWKK10] A. Steinboeck, D. Wild, T. Kiefer, and A. Kugi. A mathematical model of a slab reheating furnace with radiative heat transfer and non-participating gaseous media. *International Journal of Heat and Mass Transfer*, 53(25):5933--5946, 2010.

[SWKK11] A. Steinboeck, D. Wild, T. Kiefer, and A. Kugi. A fast simulation method for 1d heat conduction. *Mathematics and Computers in Simulation*, 82(3):392--403, 2011.

[TLRY01] L. Tang, J. Liu, A. Rong, and Z. Yang. A review of planning and scheduling systems and methods for integrated steel production. *European Journal of Operational Research*, 133(1):1--20, 2001.

[Tor89] V. J. Torczon. *Multi-directional search: a direct search algorithm for parallel machines*. PhD thesis, Citeseer, 1989.

[Tor97] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1):1--25, 1997.

[Tot04] G. E. Totten. Heat treating in 2020: what are the most critical issues and what will the future look like? *Heat Treatment of Metals*, 31(1):1--3, 2004.

[Tri03] W. Trinks. *Industrial furnaces*, volume 1. John Wiley & Sons, 2003.

[Tse99] P. Tseng. Fortified-descent simplicial search method: A general approach. *SIAM Journal on Optimization*, 10(1):269--288, 1999.

[VB] F. Vanden Berghen. Rosenbrock method.

[vDSK02] G. A. J. M. van Ditzhuijzen, D. Staalman, and A. Koorn. Identification and model predictive control of a slab reheating furnace. In *Control Applications, 2002. Proceedings of the 2002 International Conference on*, volume 1, pages 361--366. IEEE, 2002.

[Ven06] A. N. Venkat. *Distributed Model Predictive Control: Theory and Applications*. PhD thesis, University of Wisconsin--Madison, 2006.

[Ves86] T. Veslocki. Automatic slab heating control at inland's 80-in. hot strip mill. *Iron Steel Eng.*, 63(12):47--54, 1986.

[VGB05] B. Vinsonneau, D. P. Goodair, and K. J. Burnham. Hierarchical supervisory control and plant monitoring applied to industrial furnaces incorporating a priori knowledge. In *Systems Engineering, 2005. ICSEng 2005. 18th International Conference on*, pages 118--123. IEEE, 2005.

[VJKM08] F. Vode, A. Jaklic, T. Kokalj, and D. Matko. A furnace control system for tracing reference reheating curves. *Steel Research International*, 79(5):364, 2008.

[Wö] A. H. Wörk. Optimal control of slab reheating furnaces. In *Proceedings of the IFAC Symposion on the control of distributed parameter systems, Banff, Alberta, Canada*.

[WCGS99] Z. Wang, T. Chai, S. Guan, and C. Shao. Hybrid optimization setpoint strategy for slab reheating furnace temperature. In *American Control Conference, 1999. Proceedings of the 1999*, volume 6, pages 4082--4086. IEEE, 1999.

[Wes88] R. Westdorp. Development and simulation of a control strategy for a reheating furnace. *Journal A*, 29(2):11--16, 1988.

[WK99] H. J. Wick and F. Koster. Estimation of temperature profiles of slabs in a reheat furnace by using the kalman filter. In *Computer Aided Control System Design, 1999. Proceedings of the 1999 IEEE International Symposium on*, pages 613--617. IEEE, 1999.

[Wri10] M. H. Wright. Nelder, mead, and the other simplex method. *Documenta Mathematica*, (7), 2010.

[WWC04] Z. Wang, Q. Wu, and T. Chai. Optimal-setting control for complicated industrial processes and its application study. *Control Engineering Practice*, 12(1):65 -- 74, 2004.

[WYY09] A. A. Waelen, B. Young, and W. Yu. Adaptive supervisory control of an industrial steel slab reheating furnace. *Chemical Product and Process Modeling*, 4(3), 2009.

[YL86] Y. Y. Yang and Y. Z. Lu. Development of a computer control model for slab reheating furnaces. *Computers in Industry*, 7(2):145--154, 1986.

[YL88] Y. Y. Yang and Y. Z. Lu. Dynamic model based optimization control for reheating furnaces. *Computers in industry*, 10(1):11--20, 1988.

[YNY91] N. Yoshitani, Y. Naganuma, and T. Yanai. Optimal slab heating control for reheating furnaces. In *American Control Conference, 1991*, pages 3030--3035. IEEE, 1991.

[YUU94] N. Yoshitani, T. Ueyama, and M. Usui. Optimal slab heating control with temperature trajectory optimization. In *Industrial Electronics, Control and Instrumentation, 1994. IECON'94., 20th International Conference on*, volume 3, pages 1567--1572. IEEE, 1994.

[ZCX+02] B. Zhang, Z. Chen, L. Xu, J. Wang, J. Zhang, and H. Shao. The modeling and control of a reheating furnace. In *American Control Conference, 2002. Proceedings of the 2002*, volume 5, pages 3823--3828. IEEE, 2002.

[ZIS97] C. Zhang, T. Ishii, and S. Sugiyama. Numerical modeling of the thermal performance of regenerative slab reheat furnaces. *Numerical Heat Transfer, Part A Applications*, 32(6):613--631, 1997.

[ZL07] Y. Zhang and S. Li. Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes. *Journal of process control*, 17(1):37--50, 2007.

[ZXWS01] B. Zhang, L. Xu, J. Wang, and H. Shao. Optimization of combustion control based on fuzzy logic. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, volume 3, pages 1080--1083. IEEE, 2001.

Résumé

Dans l'industrie sidérurgique, les fours de réchauffage sont les plus grands consommateurs d'énergie après les hauts fourneaux. En conséquence, la réduction de leur consommation énergétique s'avère être la préoccupation majeure dans le contexte de la commande des fours. Dans un four de réchauffage de brames, des brames d'acier sont chauffées en traversant successivement plusieurs zones de la température ambiante à un profil de température homogène de 1250 °C en sortie du four, avant d'être laminées par la suite dans les laminoirs à chaud. La température des brames est contrôlée principalement par une structure de commande hiérarchisée à deux niveaux (niveau 1 et 2).

L'objectif de ces travaux est d'améliorer la performance du chauffage et en conséquence de réduire la consommation énergétique du four à l'aide d'une stratégie de commande prédictive distribuée et hiérarchisée implantées sur les deux niveaux de commande. Dans un premier temps, une approche de commande prédictive distribuée est développée pour le niveau 1 afin de suivre les consignes de température de chaque zone. L'aspect distribué de la commande permet de prendre en compte les couplages entre les zones tout en induisant une moindre complexité d'implantation par rapport à une approche totalement centralisée. L'implantation industrielle de cette stratégie a permis une amélioration significative de la précision du suivi de température et une réduction de la consommation d'énergie de 3%. Une deuxième étape propose l'élaboration de la commande prédictive hiérarchisée du niveau 2 afin, à partir de la consigne de température de la brame, de déterminer les consignes de température optimales des zones en se fondant sur un modèle de transfert thermique du four. Les résultats de la simulation obtenus avec cette stratégie comparés aux données industrielles montrent une réduction de la consommation énergétique de 5% et une meilleure qualité de chauffage des brames. L'approche précédente est enfin étendue pour prendre en compte et optimiser le cadencement des brames au sein de la commande prédictive du niveau 2, afin d'augmenter la productivité du four. La simulation montre alors une augmentation potentielle de productivité du four de 15 tonnes par heure tout en améliorant la qualité de chauffage des brames.

Abstract

In steel industry, reheating furnaces are the biggest energy consumers after blast furnaces. As a result, reduction of energy consumption is the major concern of furnace control. In a walking-beam slab reheating furnace, steel slabs are heated by moving through successive zones from ambient temperature to a homogenous temperature profile of 1250°C at the furnace exit, to be rolled subsequently in the hot rolling mills. Temperature of slabs is controlled mainly by a two-level hierarchical structure, so called level 1 and level 2.

The aim of this thesis is to improve the heating performance and consequently to reduce the energy consumption of the furnace by using hierarchical distributed model predictive control (MPC) strategy for both levels. In a first step, distributed model predictive controllers are developed for the level 1 in order to track zone temperature set-points. The distributed feature of the control law enables to consider coupling effects between zones while reducing the computation complexity compared to a complete centralized approach. The industrial results showed significant improvement on temperature tracking accuracy and an energy consumption reduction of 3%. In a second step, the hierarchical MPC is constructed for the level 2 in order to determine the optimal zones temperature setpoint from the slab temperature setpoint, based on a numerical heat transfer model of the furnace. The simulation results obtained with this strategy compared against industrial data show an energy consumption reduction of 5% and a better heating quality. The previous structure is finally extended to take into account and optimize the scheduling of the slabs within the MPC level 2 in order to increase productivity of the considered furnace. The simulation shows a potential increase of productivity of the furnace of 15 tons per hour while improving the slab heating quality.