Received April 28, 2019, accepted May 14, 2019, date of publication June 7, 2019, date of current version July 1, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2921582*

# GPU-Based Model Predictive Control of Nonlinear Parabolic Partial Differential Equations System and Its Application in Continuous Casting

**YUAN WANG, XIAOCHUAN LUO, (Member, IEEE), AND HAIXIAO WANG**
College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China

Corresponding author: Xiaochuan Luo (luoxch@mail.neu.edu.cn)

This work was supported by National Key R&D Program of China (2017YFB0304100), National Natural Science Foundation of China (51634002), Fundamental Research Funds for the Central Universities (N170708020) and the Open Research Fund from the State Key Laboratory of Rolling and Automation, Northeastern University (2018RALKFKT008).

**ABSTRACT** Temperature control of steel billets plays an important role in the quality of steel billets. This paper was motivated by the necessity for setting a value of spray cooling water flow with regard to the accuracy, fast applicability to real-time optimization, and capability of non-steady operation scenarios, especially for the change of casting speed. Therefore, this paper is focused on the GPU-based model predictive control (MPC) for temperature control of steel billets. The system dynamics in MPC is a heat transfer model characterized by the nonlinear parabolic partial differential equations (PDEs). This paper presented two algorithms. First, an adaptive trust-region Levenberg–Marquardt method (ATR-LM) based on the measured surface temperature is presented to estimate the unknown parameters in the heat transfer model. The corresponding experimental results indicate that the presented method can reduce the iteration time. Second, for the purpose of satisfying the real-time requirement, the stream parallel sparse Jacobian method (SP-SJ) is presented to solve the dynamic optimization problem associated with the PDEs. The corresponding experimental results show that the presented method exhibits satisfactory computational performance and achieves satisfactory control performance.

**INDEX TERMS** MPC, PDEs, graphic processing unit (GPU), parallel computation, continuous casting.

## I. INTRODUCTION

A typical continuous casting of steel is a process in which molten steel (liquid) is continuously solidified into steel billets (solid). As shown in Fig.1, steel billets undergo three cooling areas: a mold, a secondary cooling zone (SCZ), and an air cooling zone. Mold can remove the heat of molten steel to grow a solid shell of sufficient thickness. The SCZ is beneath the mold where there are various cooling water sprays to continue cooling the steel billets. The SCZ plays a significant role during the cooling process of the steel billets. The primary control target of SCZ is to maintain the stability of the temperature of steel billets in SCZ, and the controlled variable is cooling water flow rate [1].

Thus, the temperature field of steel billets is a key state variable. The cooling process can be described by nonlinear PDEs. However, the operation conditions always change, such as the frequent changes of casting speed and the changes of casting temperature. Frequent changes in casting speed may result in large temperature fluctuations. Subsequently, the large temperature fluctuation may result in the cracking of steel billets. In order to improve the quality of steel billets, the control law should be adjusted with respect to the water flow rate to eliminate the temperature fluctuation.

Attempts to improve the control performance of spray cooling control system have resulted in the application of numerous control schemes in various situations. The simplest parameter control method is the speed-tie approach. The speed-tie approach determines the water flow rate which is proportional to the casting speed. Specifically, the relationship between the water flow rate and casting speed is shown as follows [2]

$$u_i(t) = a_i[V_c(t)]^2 + b_i V_c(t) + c_i, \quad i = 1, 2, \cdots Nz \qquad (1)$$

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Touriño.

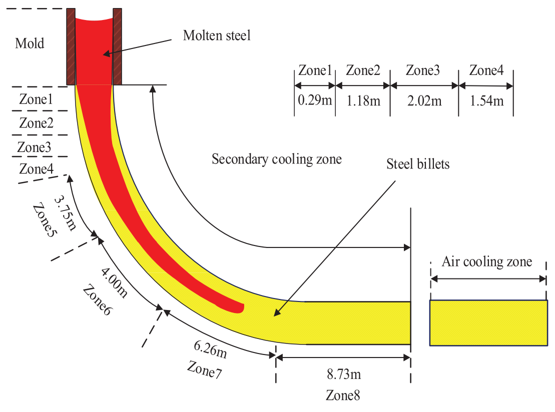

**FIGURE 1.** **Cooling process of steel billets.**

where $V_c$ represents the casting speed $(m/s)$, $a_i$, $b_i$, and $c_i$ are the parameters determined by operator's experience.

The parameter control method does not take into account the process variables in continuous casting such that an unsteady casting speed may cause the steel billets to crack. Actually, it belongs to open-loop control method. An abrupt variation in casting speed would produce a corresponding change in the water flow rate, which would result in considerable surface temperature fluctuations. They do not adjust the water flow dynamically. Thus, the parameter control method only works exceedingly satisfactory for steady-state operations.

In order to overcome the drawbacks of the parameter control method, the water flow rate should be dynamically adjusted owing to the difference between the measured temperature and the target temperature. It is obvious that the simple form of this method is the PI control algorithm [4]–[6]. The credibility and accuracy of the measured temperature are important for this method. However, the measured temperature is unreliable due to poor production environment, such as the oxide skin of steel billets, dust, and water vapor, etc [5]. Moreover, the internal temperature of the steel billets cannot be measured. Additionally, the metallurgical principles including temperature constraints and temperature gradient constraints are crucial for the quality of steel billets. Obviously, this method cannot consider these constraints.

The quality control of steel billets is fundamental for reducing the reheating of steel billets and ensuring the metallurgical principles. This cannot be achieved without a detail 3D heat transfer model. The idea of using heat transfer model to optimize a continuous caster has been described [7]. Petrus *et al.* [8] implemented a computational approach integrated with GA to determine operational parameters. Another approach to dynamic spray cooling modeling and control was developed by Louhenkilpi *et al.* [9]. This model uses both feedback and feedforward techniques to control the temperatures at end-zone locations. Louhenkilpi *et al.* [10] presented an online model for a dynamic spray cooling control system (CASIM). Also, a control model called DYNCOOL has been developed by Louhenkilpi based on the real-time model [11].

## A. MOTIVATION AND INNOVATION

As mentioned above, the traditional methods work exceedingly satisfactorily under steady-state operation condition. However, unsteady-state operation conditions (the variation of processing time, tundish replacement, and device failure) often occur, which has resulted in time conflicts or the interruption of production [12]. Thus, the schedule system would adjust the setting value of the casting speed to maintain the normal production of the casting machine [13]. The temperature of steel billets changes as the casting speed changes. If the water flow rate is set inappropriately, the temperature fluctuation would become large and the metallurgical constraints cannot be satisfied, so the steel billets have a significant possibility for crack.

MPC has been proved to be an efficient tool to improve the quality of steel billets. As mentioned above, the idea of using MPC to search the optimal water flow rate has been reported in [10], [15]. The temperature fluctuation can be formulated as the control target and the metallurgical principles are intuitive to formulate the cost function and the constraints, respectively.

Several MPC systems have been developed to control the cooling water flow rate under transient conditions for continuous caster. The simple MPC has been applied to control the cooling spray water flow rate using 1-D heat transfer model to predict the temperature behavior of steel billets [9]. Based on 2-D heat tranfer model, Hardin *et al.* [3] developed MPC system to control cooling water flow rate in real time. Louhuenklpi *et al.* [10] have developed MPC system, DYN3D, to adjust cooling water flow rate dynamically.

Several previous attempts have made to implement MPC systems of continuous caster. Actually, MPC belongs to a kind of dynamic optimization problem. Thus, MPC has very high real-time calculation requirements. Meanwhile, 3D heat transfer model used in MPC has a large computational cost. In the face of more and more complicated heat transfer model, the numerical solution method of MPC appears powerless. How to build an effective dynamic 3D heat transfer model for real-time calculation with efficiency is an exceedingly practical, important and challenging problem. This is the key difficulty to be solved in this study.

In [14], GPU-based 3D heat transfer model for dynamic simulation of continuous casting was developed. In [15], a simple and trial MPC for continuous casting including GPU-based heat transfer model was presented but the constraints were not considered, and the detail of the numerical solution method of MPC was not described. Inspired by [14], [15], this study is devoted to a new numerical solution method of MPC associated with GPU-based 3D heat transfer model. The difference between the current study and those in literature [14], [15] is that this work considers both the temperature and reheating of steel billets that are subjected to constraints. Moreover, this work not only uses GPU-based heat transfer model but also design a new perturbation vector to recycle the wasteful computation of sparse Jocabian matrix.

The primary contributions of this work: (1) A new iterative algorithm (ATR-LM method) is presented to identify the unknown parameters. Compared with Landweber method [25] and Cao method [24], the presented algorithm has satisfactory convergence property and can reduce the iteration time. (2) SP-SJ method is presented to solve the dynamic optimization problem associated with MPC. Compared with traditional solution scheme of MPC for SCZ, the calculation time of the presented algorithm decreases to 0.49%, which is the fundamental of the real-time optimization.

### B. STRUCTURE

The remainder of this paper is organized as follows. Section 2 describes dynamic control model of continuous casting. Section 3 reviews the strategies for identifying heat transfer coefficients and presents the ATR-LM method. Section 4 introduces MPC of continuous casting based on SP-SJ method and the simulation experiments are shown.

## II. DYNAMIC CONTROL MODEL OF CONTINUOUS CASTING

Production environments are subject to various unexpected disruptions in continuous casting such as casting speed changes, and traditional control schemes could not be adapted to accommodate these uncertain events. As mentioned earlier, casting speed is the most vital factor for a spray cooling system, but it is not a fixed number. Casting speed is a coefficient in the PDEs describing the process of continuous casting. Continuous caster operator needs the surface temperature of steel billets (the solution of PDEs) to be static.

Based on the explanation in Section I, the heat transfer model is an important component of the spray cooling control system. In a continuous caster, molten steel is solidified into steel billets. This process can be considered as the release and transfer of heat energy. Therefore, the heat transfer model can be used to describe the solidification process of steel billets [3].

$$\begin{aligned}\rho(T)\,C_e(T)\left(\frac{\partial T}{\partial t}+V_c\frac{\partial T}{\partial y}\right) &= \frac{\partial}{\partial x}\left(\lambda(T)\frac{\partial T}{\partial x}\right)\\ &\quad+\frac{\partial}{\partial y}\left(\lambda(T)\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(\lambda(T)\frac{\partial T}{\partial z}\right),\\ &\quad (x,y,z)\in(0,l_x)\times\left(0,l_y\right)\times(0,l_z) \end{aligned}\tag{2}$$

where $\rho(T)$ is the density of steel ($kg/m^3$), $C_e(T)$ is the effective heat capacity ($J/(kg\cdot K)$), $k(T)$ is the thermal conductivity ($W/(m\cdot K)$), $T(x,y,z,t)$ is the temperature distribution ($K$), $V_c$ is the casting speed ($m/s$), $l_x$ is the width of steel billets ($m$), $l_y$ is the length of steel billets ($m$), $l_z$ is the thickness of steel billets ($m$), $t_f$ is simulation time ($s$).

The effective heat capacity $C_e(T)$ can be represented as [2], [8]

$$C_e(T)=\begin{cases} c_l & T>T_l,\\ c_l+(c_s-c_l)f_s+\dfrac{L}{T_l-T_s} & T_s\le T\le T_l\\ c_s & T<T_s,\end{cases}\tag{3}$$

where $T_s$ is the solidus temperature ($K$), $T_l$ is the liquids temperature ($K$), $c_s$ is the specific heat of solid phase ($J/(kg\cdot K)$), $c_l$ is the specific heat of liquid phase ($J/(kg\cdot K)$), $L$ is the latent heat ($J/kg$), $k(T)$ can be described by [3]

$$k(T)=\begin{cases} k_s, & T<T_s\\ f_sk_s+m(1-f_s)k_l, & T_s\le T\le T_l\\ k_l, & T_l<T\end{cases}\tag{4}$$

where $k_s$ is the thermal conductivity of steel in the solid zone ($W/m\cdot K$), $k_l$ is the thermal conductivity of steel in the liquid zone ($W/m\cdot K$), and $m$ is thermal conductivity enhancement factor [15]

$$f_s=\begin{cases} 1, & T<T_s\\ \dfrac{T_l-T}{T_l-T_s}, & T_s\le T\le T_l\\ 0, & T_l<T\end{cases}\tag{5}$$

The initial temperature is equal to the casting temperature.

$$T(x,y,z,0)=T_{cast}\tag{6}$$

where $T_{cast}$ is the casting temperature ($K$).

Boundary conditions are shown as follows [2], [18].

$$\begin{aligned} -k\left.\frac{\partial T}{\partial x}\right|_{x=0}&=q, & -k\left.\frac{\partial T}{\partial x}\right|_{x=l_x}&=q\\ T(x,0,z,t)&=T_{cast}, & -k\left.\frac{\partial T}{\partial y}\right|_{y=l_y}&=0\\ -k\left.\frac{\partial T}{\partial z}\right|_{z=0}&=q, & -k\left.\frac{\partial T}{\partial z}\right|_{z=l_z}&=q \end{aligned}\tag{7}$$

In the mold, the boundary conditions are expressed by the following equation [2], [18]

$$q=A-B\sqrt{L/V_c}\tag{8}$$

where $q$ represents the mold heat flux ($W/m^2$), and $L$ is the length of mold ($m$). The values of $A$ and $B$ are shown in Appendix.

In secondary and air cooling zones, the boundary conditions can be expressed as [2], [18]

$$q=h(T-T_w)\tag{9}$$

where $h$ represents the heat transfer coefficient ($W/m^2K$), $T$ is the surface temperature of steel billets ($K$), $T_w$ is the temperature of cooling water ($K$).

Heat transfer mechanisms in the secondary cooling zones are shown in Fig. 2. In the spraying area, the steel billets

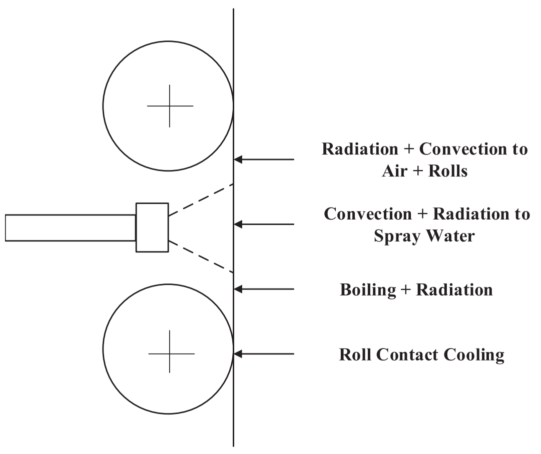
Radiation + Convection to Air + Rolls
Convection + Radiation to Spray Water
Boiling + Radiation
Roll Contact Cooling

**FIGURE 2.** Heat transfer mechanisms in the secondary cooling zones.

are cooled by a spray of cooling water. The heat transfer coefficients $h_{spray}$ are defined as [10]

$$h_{spray} = \frac{1570.0u^{0.55}\left[1.0 - 0.0075T_w\right]}{\beta} \tag{10}$$

where $u$ is the cooling water flow rate $(L/m^2 \cdot s)$, $\beta$ is a machine dependent calibration factor in the $i$ th cooling zone and $T_w$ is the temperature of the cooling water $(K)$. $\beta$ is the unknown parameter to be identified.

Radiation is computed by

$$h_{rad} = \sigma\epsilon(T + T_w)(T^2 + T_w^2) \tag{11}$$

where $\sigma$ is defined as the Stefan-Boltzmann constant $(J/K)$, and $\epsilon$ is the emissivity.

The emissivity as a function of surface temperature is given by the following equation [17]

$$\epsilon = \frac{0.85}{1 + exp(42.68 - 0.02682T)^{0.0115}} \tag{12}$$

In the roll contact area, the heat transfer coefficients can be obtained by the following equation [19]

$$h_{roll} = \frac{f_{roll}}{L_{roll}(1 - f_{roll})}\left[(h_{rad} + h_{spray})L_{spray} + h_{rad}(L_{spraypitch} - L_{spray} - L_{rollcontace})\right] \tag{13}$$

A typical $f_{roll}$ value of 0.05 produces local temperature drops beneath the rolls $100K$.

In this study, the PDEs (2) are discretized by the finite difference explicit scheme. According to Taylor's formula, the following equations can be obtained.

$$\begin{aligned}
&\rho\left(T_{ijk}^t\right)C_p\left(T_{ijk}^t\right)\left(\frac{T_{ijk}^{t+\Delta t} - T_{ijk}^t}{\Delta t} + V_c\frac{T_{ij+1k}^t - T_{ijk}^t}{\Delta y}\right)\\
&= + \frac{k\left(T_{ijk}^t\right)}{(\Delta x)^2}T_{i+1jk}^t - \frac{2k\left(T_{ijk}^t\right)}{(\Delta x)^2}T_{ijk}^t + \frac{k\left(T_{ijk}^t\right)}{(\Delta x)^2}T_{i-1jk}^t\\
&\quad + \frac{k\left(T_{ijk}^t\right)}{(\Delta y)^2}T_{ij+1k}^t - \frac{2k\left(T_{ijk}^t\right)}{(\Delta y)^2}T_{ijk}^t + \frac{k\left(T_{ijk}^t\right)}{(\Delta y)^2}T_{ij-1k}^t\\
&\quad + \frac{k\left(T_{ijk}^t\right)}{(\Delta z)^2}T_{ijk+1}^t - \frac{2k\left(T_{ijk}^t\right)}{(\Delta z)^2}T_{ijk}^t + \frac{k\left(T_{ijk}^t\right)}{(\Delta z)^2}T_{ijk-1}^t\\
&\quad i = 2, \cdots nx - 1, j = 2, \cdots ny - 1, k = 2, \cdots nz - 1
\end{aligned} \tag{14}$$

where $i, j, k$ are the indexes of space nodes in $x, y, z$; $nx$, $ny$, $nz$ are the number of nodes in $x$, $y$, $z$; and $\Delta x$, $\Delta y$, $\Delta z$ are the space steps in $x$, $y$, $z$, respectively; $t$ and $t + \Delta t$ are defined as before and after the incremental time interval, respectively. For simplicity, $T(x_i, y_j, z_k, t)$ is replaced by $T_{ijk}^t$.

Let $a_{ijk}^t = k(T_{ijk}^t)/(\rho(T_{ijk}^t)C_e(T_{ijk}^t))$, the above equation can be rewritten as

$$\begin{aligned}
T_{ijk}^{t+\Delta t} &= \frac{a_{ijk}^t\Delta t}{(\Delta x)^2}T_{i+1jk}^t + \left(\frac{a_{ijk}^t\Delta t}{(\Delta y)^2} - \frac{V_c\Delta t}{\Delta y}\right)T_{ij+1k}^t\\
&\quad + \left(1 - \frac{2a_{ijk}^t\Delta t}{(\Delta x)^2} - \frac{2a_{ijk}^t\Delta t}{(\Delta y)^2} - \frac{2a_{ijk}^t\Delta t}{(\Delta z)^2} + \frac{V_c\Delta t}{\Delta y}\right)T_{ijk}^t\\
&\quad + \frac{a_{ijk}^t\Delta t}{(\Delta z)^2}T_{ijk+1}^t + \frac{a_{ijk}^t\Delta t}{(\Delta x)^2}T_{i-1jk}^t + \frac{a_{ijk}^t\Delta t}{(\Delta y)^2}T_{ij-1k}^t\\
&\quad + \frac{a_{ijk}^t\Delta t}{(\Delta z)^2}T_{ijk-1}^t, \quad i=2\cdots nx-1, j=2, \cdots ny - 1,\\
&\qquad k = 2, \cdots nz - 1
\end{aligned} \tag{15}$$

The stability condition of the finite difference method is shown as follows

$$\Delta t \leq \frac{1}{\frac{2k}{C\rho\Delta x^2} + \frac{2k}{C\rho\Delta y^2} + \frac{2k}{C\rho\Delta z^2} + \frac{C\rho V_c}{\Delta y}} \tag{16}$$

The values of some of the parameters of heat transfer model are given in Appendix A.

## III. IDENTIFICATION OF UNKNOWN PARAMETERS IN CONTINUOUS CASTING BASED ON ATR-LM METHOD

The unknown parameter $\beta$ in heat transfer model is shown in (10). In many research works, $\beta$ is determined by the lab trials. Recently, some research works obtain heat transfer coefficients by using the surface temperature measurements, which is proven to be a more effective method. Thus, this study also uses the surface temperature measurements to identify $\beta$.

For simplicity, the heat transfer coefficients $h$ are identified in our model instead of $\beta$, which are identical. The estimation parameters refer to $\mathbf{h} = [h_1, h_2, \cdots, h_{N_Z}]^T$. To estimate the unknown parameters, the sum of squares of residuals is minimized. Thus, the cost function has the following form

$$f(\mathbf{h}) = \frac{1}{2}\|r(\mathbf{h})\|^2 \tag{17}$$

Obviously, $f(\mathbf{h})$ is the function from $R^N$ to $R$, and $N$ is the number of measured point.

The residual vector $r(\mathbf{h})$ between the predicted temperature $\mathbf{T_p}$ and measured temperature $\mathbf{T_m}$ refers to

$$r(\mathbf{h}) = \mathbf{T_p} - \mathbf{T_m} \tag{18}$$

where

$$\mathbf{T_p} = \left(T_p^1, T_p^2, \cdots, T_p^N\right)^T, \quad \mathbf{T_m} = \left(T_m^1, T_m^2, \cdots, T_m^N\right)^T$$

$r(\mathbf{h}) = (r_1, r_2, \cdots, r_N)^T$ is the $N$ dimensional residual vector.

Several optimization methods have been extensively developed to solve this problem, such as gradient method [20], conjugate gradient method [21], [22], Levenberg-Marquardt method (LM) [18], [23], Cao method [24], and Lanbweber method [25]. The basic concept of these optimization methods is to update $\mathbf{h}$ for each iteration via the gradient of cost function or the value of cost function. The identification problem of heat transfer coefficients belongs to the ill-posed problem [24], meaning that a small error in the measured data can result in substantially large errors in the answers, so an ill-posed problem suffers from numerical non convergence. Levenberg-Marquardt (LM) method has been proved to be successful for identification of heat transfer coefficients [18], [23], and LM method is shown in Algorithm 1. More details of LM method can be found in [23].

**Algorithm 1** LM [23]

```
Input:   $\mathbf{T_m}$
Output:  The estimation of heat tranfer coefficients $\bar{\mathbf{h}}$
1:    Set iteration time $k = 0$, maximum iteration
      number $S = 50$, stop criterion $\eta = 10^{-4}$.
2:    Solve the PDEs with the boundary condition
      $\mathbf{h}_k$, and the predicted surface temperature
      $T_p(\mathbf{h_k})$ can be obtained.
3:    Compute the Jacobian matrix $\mathbf{J_k}$ using (19).
4:    Compute the $d_k$ using (22).
5:    $\mathbf{h_{k+1}} = \mathbf{h_k} + \mathbf{d_k}$
6:    Let $k = k + 1$.
7:    if $|f_{k+1} - f_k| \le \eta$ or $k > S$,
8:        Go to step 2,
9:    else
10:       output $\bar{\mathbf{h}}$.
```

The derivatives of $r(\mathbf{h})$ can be expressed in terms of the Jacobian matrix.

$$\mathbf{J}(\mathbf{h}) = \begin{bmatrix} \frac{\partial r_1}{\partial h_1} & \frac{\partial r_1}{\partial h_2} & \cdots & \frac{\partial r_1}{\partial h_N} \\ \frac{\partial r_2}{\partial h_1} & \frac{\partial r_2}{\partial h_2} & \cdots & \frac{\partial r_2}{\partial h_N} \\ \vdots & & \ddots & \vdots \\ \frac{\partial r_N}{\partial h_1} & \frac{\partial r_N}{\partial h_2} & \cdots & \frac{\partial r_N}{\partial h_N} \end{bmatrix} \tag{19}$$

$\mathbf{h_k}$ is defined as the $\mathbf{h}$ at iteration time $k$. For simplicity, this paper rewrites $\mathbf{J}(\mathbf{h_k})$ as $\mathbf{J_k}$, $\mathbf{r}(\mathbf{h_k})$ as $\mathbf{r_k}$, $\mathbf{f}(\mathbf{h_k})$ as $f_k$. Using the Taylor approximation $\mathbf{r}(\mathbf{h_k}+\mathbf{d_k})$, the cost function approximator $m_k(\mathbf{d_k})$ is defined as follows

$$m_k(\mathbf{d_k}) = \frac{1}{2}\|\mathbf{J_k}\mathbf{d_k} + \mathbf{r_k}\|^2 \tag{20}$$

where $m_k(\mathbf{d_k})$ is the approximator of $f_{k+1}$ and $m_k(\mathbf{0})$ is equal to $f_k$, and $\mathbf{d_k} \in R^N$ is the search direction at iteration time $k$.

Subsequently, the following subproblem is solved at $k$ iteration time

$$\begin{aligned} &\min_{\mathbf{d_k}} \ m_k(\mathbf{d_k}) \\ &s.t. \ \|\mathbf{d_k}\| \le \Delta_k \end{aligned} \tag{21}$$

where $\Delta_k > 0$ is the trust-region radius and $\mathbf{d_k}$ is the search direction.

The cost function is the first-order Taylor expansion for $r(\mathbf{h_k}+\mathbf{d_k})$. The motivation of LM method is based on the gradient of cost function. The gradient provides a local information of cost function but not represents a global information of cost function, so the trust-region is defined. The constraint $\|\mathbf{d_k}\| \le \Delta_k$ means that this approximation is only effective within the trust-region radius.

According to Karush–Kuhn–Tucker (KKT) condition, the search direction $\mathbf{d_k}$ can be obtained

$$\mathbf{d_k} = -\left(\mathbf{J_k}^T\mathbf{J_k} + \lambda_k\mathbf{I}\right)^{-1}\mathbf{J_k}^T\mathbf{r_k} \tag{22}$$

One important new problem arises: How to set the parameters $\lambda_k$ which controls the radius of trust-region. In the following part of this section, an ATR-LM method that has the effective strategy in practice is described. The primary ingredient in the LM method is the strategy for selecting the parameter $\lambda_k$ at each iteration. The presented strategy is based on the agreement between the approximator function $m_k(\mathbf{d_k})$ and the cost function $f_{k+1}$ at previous iterations. The $\rho(k)$ is defined as

$$\rho_k = \frac{f(\mathbf{h_k}) - f(\mathbf{h_k} + \mathbf{d_k})}{m_k(0) - m_k(\mathbf{d_k})} \tag{23}$$

Note that $\rho(k)$ is a ratio of the actual reduction to the predicted reduction. If $\rho_k$ is near 1, it means that a satisfactory agreement exists between $m_k(\mathbf{d_k})$ and $f_{k+1}$ over this step, so it is confident to use the Taylor expansion for the next iteration and the trust-region should become large. If $\rho_k$ is positive and exceedingly small, it means that $m_k(\mathbf{d_k})$ is not a satisfactory estimation of $f_{k+1}$, so trust-region should become small. If $\rho_k$ is negative, the next iteration of the cost function is larger than the current iteration, so this search direction should be rejected.

According to the above description, the updated technique of $\lambda$ is shown in Algorithm 2. Furthermore, the convergence analysis of the ATR-LM is also provided in Appendix B.

### *A. NUMERICAL EXPERIMENTS*

To verify the presented ATR-LM method, the nonlinear PDEs problem is investigated. The parameters are shown as follows $V_c = 0.02m/s$, $S = 50$, $\eta = 10^{-4}$ and other parameters are shown in Appendix A. The purpose of this study is to identify the heat transfer coefficients on the boundary conditions (10). To generate the measured temperature, the exact values of the heat transfer coefficients are assigned as $\mathbf{h}^*$. Solve the

**Algorithm 2** ATR-LM

```
Input:   $\mathbf{T_m}$
Output:  The estimation of heat tranfer coefficients $\bar{\mathbf{h}}$
1:   Set iteration time $k = 0$, maximum iteration
     number $S = 50$, stop criterion $\eta = 10^{-4}$.
2:   Solve the PDEs with the boundary condition $\mathbf{h}_k$,
     and the predicted surface temperature
     $T_p(\mathbf{h_k})$ can be obtained.
3:   Compute the Jacobian matrix $\mathbf{J_k}$ using (19).
4:   Compute the $\rho_k$ using (23)
5:   Update $\lambda_{k+1}$.
6:   if $\rho_k > \frac{3}{4}$
7:       $\lambda_{k+1} = \frac{1}{2}\lambda_k$
8:       $\mathbf{d_k} = -\left(\mathbf{J_k}^T\mathbf{J_k} + \lambda_k\mathbf{I}\right)^{-1}\mathbf{J_k}^T\mathbf{r_k}$
9:       $\mathbf{h}_{k+1} = \mathbf{h}_k + \mathbf{d}_k$
10:  else if $\frac{1}{4} \le \rho_k \le \frac{3}{4}$
11:      $\lambda_{k+1} = \lambda_k$
12:      $\mathbf{d_k} = -\left(\mathbf{J_k}^T\mathbf{J_k} + \lambda_k I\right)^{-1}\mathbf{J_k}^T\mathbf{r_k}$
13:      $\mathbf{h_{k+1}} = \mathbf{h_k} + \mathbf{d_k}$
14:  else if $0 \le \rho_k < \frac{1}{4}$
15:      $\lambda_{k+1} = 2\lambda_k$
16:      $\mathbf{d_k} = -\left(\mathbf{J_k}^T\mathbf{J_k} + \lambda_k\mathbf{I}\right)^{-1}\mathbf{J_k}^T\mathbf{r_k}$
17:      $\mathbf{h_{k+1}} = \mathbf{h_k} + \mathbf{d_k}$
18:  Let $k = k + 1$.
19:  if $|f_{k+1} - f_k| \le \eta$ or $k > S$,
20:      Go to step 2,
21:  else
22:      output $\bar{\mathbf{h}}$.
```

**TABLE 1.** Measured temperature.

| Regions | $h^*$ $(W/m^2K)$ | $T_e$(K) | Measured position(m) |
|---|---|---|---|
| Zone 1 | 1200 | 1225.19 | 0.95 |
| Zone 2 | 740 | 1206.93 | 1.68 |
| Zone 3 | 420 | 1236.37 | 3.28 |
| Zone 4 | 390 | 1202.91 | 5.06 |
| Zone 5 | 330 | 1168.75 | 7.72 |
| Zone 6 | 280 | 1115.57 | 11.61 |
| Zone 7 | 250 | 1047.81 | 16.74 |
| Zone 8 | 160 | 985.19 | 24.23 |

direct heat transfer model associated with the boundary conditions $\mathbf{h}^*$. Subsequently, the temperature field of the steel billets $T(x, y, z; \mathbf{h}^*)$ can be obtained. Finally, eight measured points $\mathbf{T_e}$ are selected. $\mathbf{T_e}$ and $\mathbf{h}^*$ are shown in Table 1.

Certain noise is added into the measured points.

$$\mathbf{T_m} = \mathbf{T_e} + \delta \tag{24}$$

where $\delta$ is the normal distribution with zero mean value and $\delta$ variance.

Using the measured temperature, Landweber [25], Cao [24], LM [23] and ATR-LM are used to estimate the heat transfer coefficients $\bar{\mathbf{h}}$. To measure the performance of different methods, this experiment uses the estimation value of the heat transfer coefficients $\bar{\mathbf{h}}$ and $\mathbf{h}^*$. The comparison results between the four methods are shown in Table 2.

**TABLE 2.** Comparison ATR-LM with other three methods.

| $\delta$ | methods | $\lVert\mathbf{h}^* - \bar{\mathbf{h}}\rVert/8$ | running time (s) |
|---|---|---|---|
| 0 | ATR-LM | 0.97 | 110.24 |
| | LM | 0.96 | 140.24 |
| | Cao | 1.01 | 204.89 |
| | Landweber | 0.94 | 207.10 |
| 0.1 | ATR-LM | 1.28 | 114.35 |
| | LM | 1.16 | 141.14 |
| | Cao | 1.67 | 202.89 |
| | Landweber | 1.27 | 209.89 |
| 1.0 | ATR-LM | 3.22 | 129.12 |
| | LM | 5.22 | 157.02 |
| | Cao | 3.68 | 212.67 |
| | Landweber | 3.76 | 212.14 |
| 10.0 | ATR-LM | 18.27 | 197.18 |
| | LM | 37.36 | 203.01 |
| | Cao | 40.8 | 213.31 |
| | Landweber | 29.34 | 211.97 |

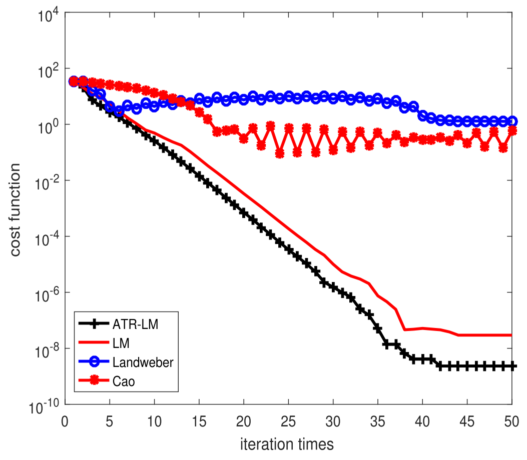

**FIGURE 3.** Comparison with the convergence behavior.

Evidently, when $\delta = 0$, those four methods have almost the same performance. So does $\delta = 0.1$. When $\delta = 1$, the LM method has the worst performance, because the parameter $\lambda_k$ of the LM method is unreasonable. On the contrary, the presented ATR-LM method is better than the Cao and Landweber methods. When $\delta = 10$, the presented ATR-LM method has the best performance. The actual values of the heat transfer coefficients determined by ATR-LM and empirical values of heat tranfer coefficients are shown in Appendix Table 11.

Fig. 3 shows a comparison of the convergence behaviors of Landweber, Cao, LM and ATR-LM. It is concluded that LM and ATR-LM methods have better performance than Cao and Landweber, because the LM and ATR-LM methods belong to second order optimization methods owing to the approximate Hessian matrix. On the contrary, Cao and Landweber do not have this property, so the convergence behaviors are not satisfactory. In addition, the convergence analysis of ATR-LM methods are given, which supports our results in theory.

Thus, this study should provide an insight into the parameter $\lambda_k$, as shown in Fig. 4. From Fig. 4, it is evident that $\lambda_k$ becomes small as the iteration times $k$ increases. The relatively large value of $\lambda_k$ means that the trust-region is

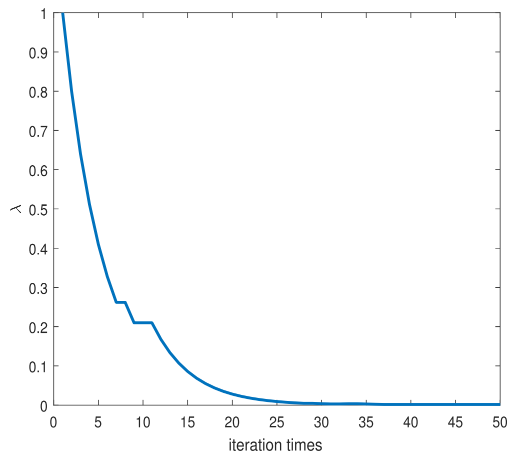

**FIGURE 4.** **The behaviour of λ with iteration times.**

**TABLE 3.** **Measured temperature.**

| Regions | Length(m) | $T_m$(K) | Measured position(m) |
|---|---|---|---|
| Zone 1 | 1.09 | 1232.19 | 0.95 |
| Zone 2 | 2.27 | 1220.93 | 1.68 |
| Zone 3 | 4.29 | 1238.38 | 3.28 |
| Zone 4 | 5.83 | 1192.76 | 5.06 |
| Zone 5 | 9.61 | 1172.36 | 7.72 |
| Zone 6 | 13.61 | 1115.27 | 11.61 |
| Zone 7 | 19.87 | 1041.27 | 16.74 |
| Zone 8 | 28.60 | 994.19 | 24.23 |

small. At the primer iteration phase, the current solution is far from the optimum, and the $m_k(\mathbf{d_k})$ is not a satisfactory estimation of the cost function $f_{k+1}$. Thus, the trust-region should be small to limit the unsatisfactory estimation of the Hessian matrix. On the contrary, the relatively small value of $\lambda_k$ means that the trust-region is large. At the late iteration times, the $m_k(\mathbf{d_k})$ becomes a satisfactory estimation of $f_{k+1}$. Owing to $m_k(\mathbf{d_k})$, the ATR-LM have a better performance.

### *B. INDUSTRIAL APPLICATIONS*

In this subsection, an actual surface temperature measurement is conducted in a steel plant. Based on the measured surface temperature, the heat transfer coefficients can be estimated. Subsequently, the corrected heat transfer model is used to predict the thickness of steel billets, which can confirm the effectiveness of the corrected model. In continuous casting, this process is called pin-shooting experiment. The related parameters are shown in Appendix A.

#### 1) TEMPERATURE MEASUREMENTS AND MODEL IDENTIFICATION

Because of the high temperature of steel billets, the non-contact temperature measurement with a measuring error less than 8K is used to measure the surface temperature of the steel billets. The measured surface temperature is shown in Table 3.

ATR-LM is used to estimate the heat transfer coefficients using the measured surface temperature. Fig. 5 shows a comparison between the predicted surface temperature (ATR-LM) and the predicted surface temperature

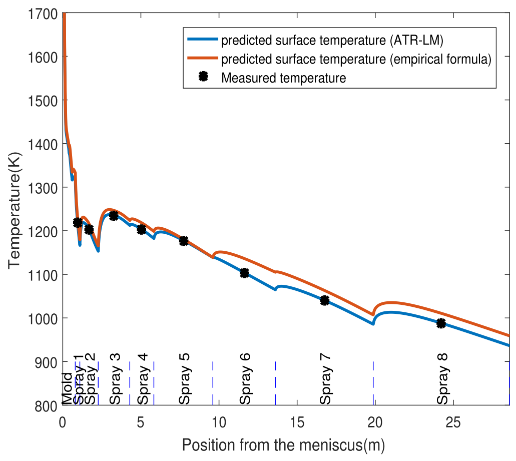

**FIGURE 5.** **Comparison between the measured and predicted surface temperature.**

(empirical formula). The predicted surface temperature obtained by using the corrected heat transfer model can follow the measured surface temperature exceedingly satisfactorily.

#### 2) MODEL VALIDATION

To test the accuracy of the corrected heat transfer model, the pin-shooting value experiment is conducted to compare the measured shell thickness with the predicted shell thickness. The pin-shooting method can measure the thickness of the steel billets with high accuracy. Then, the corrected heat transfer model is used to predict the shell thickness of the steel billets. However, the pin-shooting experiment has a poor effect on the quality of steel billets, so the result of the pin-shooting method is only used to validate the corrected heat transfer model. The nephogram of steel billets temperature field is shown as follows

The red portion represents the temperature beyond 1700(K) and it is beyond the solid temperature, so the molten steel is still liquid. As shown in Fig. 6, the molten steel inside solid-phase line is in the liquid state and the steel billets outside solid-phase line is in solid state. The pin-shooting experiment shoots the nail into the steel billets, so the thickness of the shell can be measured, as depicted in Fig. 6.

Fig. 7 shows a correlation between the measured shell thickness and the calculated shell thickness. In Fig. 7, the square root law belongs to a type of empirical method utilized to calculate the shell thickness [32]. The predicted values of the corrected heat transfer model are consistent with the measured values. The pin-shooting experiment concludes that the ATR-LM method is effective.

## IV. GPU-BASED MODEL PREDICTIVE CONTROL OF CONTINUOUS CASTING BASED ON SP-SJ METHOD

### *A. DYNAMIC OPTIMIZATION PROBLEM*

The unknown parameters in nonlinear PDEs are identified using the ATR-LM method. The temperature field is expressed by $T(x, y, z, t)$. The setting value for $N_z$, the water

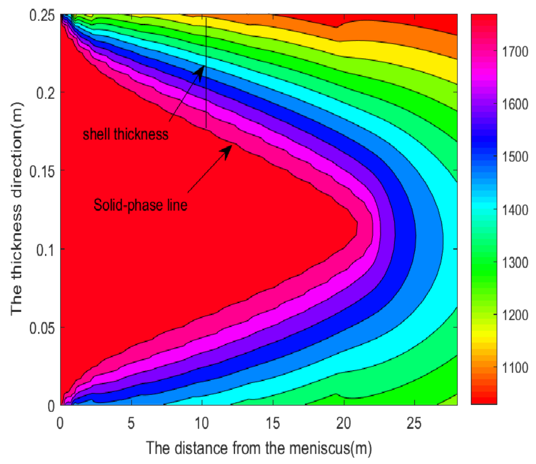

**FIGURE 6.** Measuring the shell thickness of the steel billets by the pin-shooting experiment.

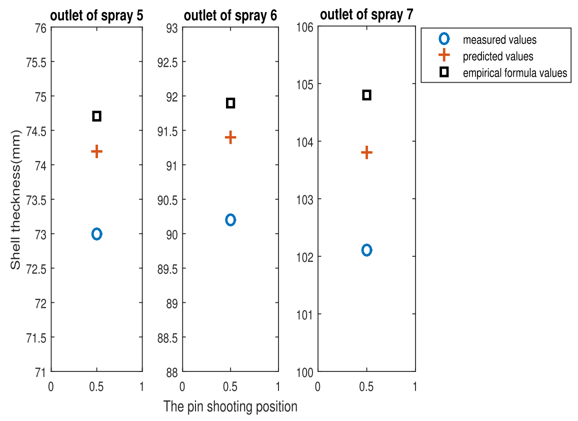

**FIGURE 7.** Comparison between the measured and calculated steel billets shell thickness.

flow rate $u(t) = [u_1(t), u_2(t), \cdots, u_{Nz}(t)]^T$ is changed from $u(t)$ to $u(t+1)$ at time $t+1$. Subsequently, the PDEs (1) with its boundary conditions can be solved, i.e., the temperature field at the next time step $T(t+1)$ can be obtained. Thus, the dynamics of temperature can be regarded as a discrete-time state-space dynamic model, which can be represented as

$$T(t+1) = f(T(t), Q(t+1)), \quad t > 0 \tag{25}$$

where $f$ represents the dynamics of nonlinear PDEs. It should be mentioned that the nonlinear coefficients $\rho(T)$, $c_e(T)$, $k(T)$ result in $f$ is implicit.

The surface average temperature of the steel billets for each cooling section can be obtained

$$T^n_{ave}(t) = \frac{1}{m_i}\sum_{j=1}^{m_i} T^t_{ijk}, \quad \forall j \in y_n \le j \times ny < y_{n+1},$$
$$i = \frac{nx-1}{2}, \; k = 0 \tag{26}$$

Based on (25) and (26), the surface average temperature of steel billets $\mathbf{T_{ave}}(t) = \left[T^1_{ave}(t), T^2_{ave}(t), \cdots, T^{Nz}_{ave}(t)\right]^T$ can be got. $T^i_{ave}(t+k)$ is the surface average temperature of the steel billets at cooling section $i$, and $T^i_{aim}(t+k)$ is the goal temperature of the steel billets at cooling section $i$.

The models used in MPC are generally intended to represent the behavior of complex dynamical systems. Based on the prediction of this model, an optimal control problem is intuitive and naturally derived via minimizing the sum of squares of future control errors over a finite time horizon while satisfying constraints [28]. The cost function is shown as follows

$$\min \sum_{k=1}^{N_p} \left\| \mathbf{T_{ave}}(t+k) - \mathbf{T_{aim}}(t+k) \right\|^2 \tag{27}$$

where $N_p$ is the prediction horizon.

1. Surface temperature constraints: The surface temperature of steel billets should be outside the low ductility temperature range (less than $\underline{T}$(978.15(K)) or over $\overline{T}$(1233.15(K))). According to the approximate mechanism analysis and engineering experience, the large temperature gradient may cause cracks. Taking these factors into account, it is difficult to define the lower and upper bounds for all grades of steel. Nevertheless, Petrus *et al.* [8] recommend that the lower temperature bound is between 973.15(K) and 1123.15(K) and the upper temperature bound is between 1173.15(K) and 1273.15(K). Considering the experience of continuous caster operator, this study has the following constraints

$$\underline{T} \le T^i_{ave} \le \overline{T}, \quad \forall i = 1, 2, \cdots, N_z \tag{28}$$

2. Reheating constraints between adjacent sections: When the steel billets process a high cooling efficient spray cooling zone to a lower one, reheating may occur. The reheating of the steel billets has resulted in the development of the cracking, so the following constraints can be obtained.

$$-100K/m \le \frac{\partial T^i_{ave}}{\partial z} \le 100K/m, \quad \forall i = 1, 2, \cdots, N_z \tag{29}$$

where $\frac{\partial T^i_{ave}}{\partial z}$ represents the reheating of the steel billets in the cooling section $i$.

The reheating is defined by the temperature gradients along with the casting direction. Thus, the temperature gradients are replaced by the difference between the average temperature in the cooling section $i$ and the average temperature in the cooling section $i+1$ gradient.

$$-100K/m \le \frac{T^{i+1}_{ave} - T^i_{ave}}{L_{i+1} - L_i} \le 100K/m$$
$$\forall i = 1, 2, \cdots, N_z - 1 \tag{30}$$

3. Shell thickness at the mold exit: The shell thickness at the mold exit must be greater than 10% of the thickness of steel billets. This constraint avoids breakout occurrences caused by extraction stresses and liquid ferrostatic pressure, and can be defined as

$$X_s\left(T^t_{ijk}\right) \ge 0.1 l_x, \tag{31}$$

where $X_s$ are the axes of coordinate ($m$) and the temperature of steel billets is equal to solidus temperature.

4. Spray cooling water flow constraints:

$$\underline{u}_i \le u_i(t+k) \le \bar{u}_i, \quad \forall k=1,2,\cdots,N_p,\ \forall i=1,2,\cdots,N_z \tag{32}$$

The values of $\underline{u}_i$ and $\bar{u}_i$ are shown in Appendix.

The inequality constraints can be transformed into the cost function. Subsequently, we have

$$\min_u L(u(t+k)) = \sum_{k=1}^{K} H(t+k) \tag{33}$$

with the definition

$$\mathbf{H}(t+k) = \|\mathbf{T_{aim}}(t+k) - \mathbf{T_{ave}}(t+k)\|^2 + \mu^T C(u(t+k)) \tag{34}$$

where

$$\begin{aligned} C(u) = \Big(&\underline{T}^i - T_{ave}^i, \cdots, \underline{T}^i - T_{ave}^{Nz}, T_{ave}^1 - \overline{T}, \cdots, \\ &T_{ave}^{Nz} - \overline{T}, L - \frac{\partial T_{ave}^1}{\partial z}, \cdots, L - \frac{\partial T_{ave}^{Nz-1}}{\partial z}, \frac{\partial T_{ave}^1}{\partial z} - D, \\ &\cdots, \frac{\partial T_{ave}^{Nz-1}}{\partial z} - D, \underline{u} - u_1, \cdots, \underline{u} - u_{Nz}, u_1 - \bar{u} \cdots, \\ &u_{Nz} - \bar{u}\Big)^T, 0.1 l_x - X_s \in R^{6Nz-1} \end{aligned}$$

There are several numerical optimization algorithms that can be used to solve this problem such as quasi-Newton methods and conjugate gradient methods. Most of those methods are based on the gradient of the cost function. The gradient $\nabla L(u) = g_{t+k}$ can be obtained

$$\begin{aligned} g_{t+k} &= \left(\frac{\partial H(t+k)}{\partial u(t+k)}\right)^T + \left(\frac{\partial H(t+k+1)}{\partial u(t+k)}\right)^T, \\ k &= 1, 2\cdots, K-1 \end{aligned} \tag{35}$$

$$g_K = \left(\frac{\partial H(K)}{\partial u(K)}\right)^T \tag{36}$$

## B. SPARSE JACOBIAN COMPUTATION

According to chain rule, the following equation can be obtained.

$$\frac{\partial H(t+k+1)}{\partial \mathbf{u}(t+k)} = \frac{\partial H(t+k+1)}{\partial \mathbf{T_{ave}}(t+k+1)} \frac{\partial \mathbf{T_{ave}}(t+k+1)}{\partial \mathbf{u}(t+k)} \tag{37}$$

Because the first term $\frac{\partial H(t+k+1)}{\partial \mathbf{T_{ave}}(t+k+1)}$ is easy to obtain, the key to calculate the gradient is to compute the second term $\frac{\partial \mathbf{T_{ave}}(t+k+1)}{\partial \mathbf{u}(t+k)}$. Actually, $\frac{\partial \mathbf{T_{ave}}(t+k+1)}{\partial \mathbf{u}(t+k)}$ is a Jacobian matrix, which is calculated by (38), as shown at the bottom of this page.

Because the relationship between $\mathbf{T_{ave}}$ and $\mathbf{u}$ is governed by nonlinear PDEs, the analytic form of $\frac{\partial \mathbf{T_{ave}}}{\partial \mathbf{u(t+k)}}$ does not exist. Thus, the partial derivation at a given point can be approximated

$$\frac{\partial \mathbf{T_{ave}}}{\partial u_j} \approx \frac{\mathbf{T_{ave}}(\mathbf{u}+\varepsilon \mathbf{e_j}) - \mathbf{T_{ave}}(\mathbf{u})}{\varepsilon}, \quad j = 1, 2, \cdots Nz \tag{39}$$

where $\varepsilon$ is a small positive scalar and $\mathbf{e_j}$ is the $i$ th unit vector, i.e., the vector whose elements are all 0 expect for a 1 in the $i$ th position.

The Jacobian matrix can be built by merely applying this formula. In general, this process requires evaluation of $\mathbf{T_{ave}}$ at $\mathbf{u}$ point and the $N_z$ perturbed points $\mathbf{u} + \varepsilon \mathbf{e_i}$: a total of $N_z + 1$ points. Therefore, $N_z + 1$ times solution of nonlinear PDEs are required to calculate the Jacobian matrix. This is an exceedingly large computation task.

For continuous casting, the Jacobian matrix has a sparse structure. Furthermore, the Jacobian matrix is a tridiagonal matrix. Because the water flow rate of 1st section $u_1(t+k\Delta t)$ only affects the average surface temperature of 1st section $T_{ave}^1(t+k\Delta t+\Delta t)$ and its neighboring region $T_{ave}^2(t+k\Delta t+\Delta t)$. $u_2(t+k\Delta t)$ only affects $T_{ave}^2(t+k\Delta t+\Delta t)$ and its neighboring regions $T_{ave}^1(t+k\Delta t+\Delta t)$ and $T_{ave}^3(t+k\Delta t+\Delta t)$. Furthermore, a perturbation $\varepsilon e_1$ to the first component of $\mathbf{u}$ will affect only the first and second components of $\mathbf{T_{ave}}(t+k\Delta t+\Delta t)$. The remaining components will remain unchanged, so that the Jacobian matrix has numerous zeros elements. Evidently, the evaluations of the components $T_{ave}^3(t+k\Delta t+\Delta t), T_{ave}^4(t+k\Delta t+\Delta t), \ldots, T_{ave}^{N_z}(t+k\Delta t+\Delta t)$ are wasteful.

*Remark 1:* Assume that $u_1(t+k\Delta t)$ only affects the average surface temperature of section one $T_{ave}^1(t+k\Delta t+\Delta t)$ and its neighboring region $T_{ave}^2(t+k\Delta t+\Delta t)$. If and only if $t_z$ is larger than the ratio of the length of cooling section 2 to casting speed, $u_1(t+k\Delta t)$ can effect $T_{ave}^3(t+k\Delta t+t_z)$. In our case, $\Delta t$ is less than the ratio of the length of cooling section 2 to casting speed, so $u_1(t+k\Delta t)$ cannot effect

$$\begin{aligned} \frac{\partial \mathbf{T_{ave}}(t+k+1)}{\partial \mathbf{u}(t+k)} &= \begin{bmatrix} \frac{\partial \mathbf{T_{ave}}(t+k+1)}{\partial u_1(t+k)} & \cdots & \frac{\partial \mathbf{T_{ave}}(t+k+1)}{\partial u_{N_z}(t+k)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial T_{ave}^1(t+k+1)}{\partial u_1(t+k)} & \frac{\partial T_{ave}^1(t+k+1)}{\partial u_2(t+k)} & \cdots & \frac{\partial T_{ave}^1(t+k+1)}{\partial u_{N_z}(t+k)} \\ \frac{\partial T_{ave}^2(t+k+1)}{\partial u_1(t+k)} & \ddots & & \vdots \\ \vdots & & & \\ \frac{\partial T_{ave}^{N_z}(t+k+1)}{\partial u_1(t+k)} & \cdots & & \frac{\partial T_{ave}^{N_z}(t+k+1)}{\partial u_{N_z}(t+k)} \end{bmatrix} \end{aligned} \tag{38}$$

$T^3_{ave}(t + k\Delta t + \Delta t)$. Therefore, the assumption was established.

To recycle the wasteful evaluations, the perturbation vector should be modified so that it does not have any further effect on $T^1_{ave}(t + k + 1)$ and $T^2_{ave}(t + k + 1)$, but does produce a change in certain components $T^3_{ave}(t + k + 1)$, $T^4_{ave}(t + k + 1), \ldots, T^{N_z}_{ave}(t + k + 1)$, which can be used to compute other columns of the Jacobian matrix. It is easy to understand that the additional perturbation $\varepsilon e_4$ possesses the desired property. Thus, the new perturbation vector is shown as follows

$$\mathbf{p_1} = \varepsilon\left(\mathbf{e_1} + \mathbf{e_4}\right) \tag{40}$$

Subsequently, the following equation can be obtained

$$\left[\frac{\partial T^1_{ave}}{\partial \mathbf{u}}\right]_{1,2} \approx \left[\frac{\mathbf{T_{ave}}\left(\mathbf{u}+\mathbf{p_1}\right) - \mathbf{T_{ave}}\left(\mathbf{u}\right)}{\varepsilon_1}\right]_{1,2} \tag{41}$$

$$\left[\frac{\partial T^4_{ave}}{\partial \mathbf{u}}\right]_{3,4,5} \approx \left[\frac{\mathbf{T_{ave}}\left(\mathbf{u}+\mathbf{p_1}\right) - \mathbf{T_{ave}}\left(\mathbf{u}\right)}{\varepsilon_4}\right]_{3,4,5} \tag{42}$$

where the notation $[\cdot]_{1,2}$ represents the subvector consisting of the 1st and 2nd elements.

Overall, this study can meanwhile compute the two columns of the Jacobian by evaluating the function $\mathbf{T_{ave}}$ at one time. The remainder columns of the Jacobian matrix can be approximated in this economical manner as well, such as $\mathbf{p_2} = \varepsilon\left(\mathbf{e_2} + \mathbf{e_5}\right)$ and $\mathbf{p_3} = \varepsilon\left(\mathbf{e_3} + \mathbf{e_6}\right)$.

Actually, three extra evaluation of $\mathbf{T_{ave}}$ are sufficient to calculate the entire Jacobian matrix. The selections of perturbation $\mathbf{p}$ are defined as

$$\begin{aligned}\mathbf{p_1} &= \varepsilon\left(\mathbf{e_1} + \mathbf{e_4} + \mathbf{e_7} + \cdots\right)\\ \mathbf{p_2} &= \varepsilon\left(\mathbf{e_2} + \mathbf{e_5} + \mathbf{e_8} + \cdots\right)\\ \mathbf{p_3} &= \varepsilon\left(\mathbf{e_3} + \mathbf{e_6} + \mathbf{e_9} + \cdots\right)\end{aligned} \tag{43}$$

*Remark 2:* In fact, for any selection of $N_z$ (irrespective of how large the Jacobian matrix is), three extra evaluations of PDEs are sufficient to approximate the entire Jacobian. In contrast, if the sparse structure is not used in this study to compute the Jacobian, $N_z$ evaluation of PDEs are necessary to approximate the entire Jacobian matrix.

## C. STREAM PARALLEL IMPLEMENTATION OF SPARSE JACOBIAN COMPUTATION

As mentioned above, obtaining the solution of the nonlinear PDEs is the most significant computation bottleneck in the dynamic optimization problem. GPUs with thousands of threads have power ability to handle large parallel computational tasks, which have been applied to problems with high computing costs such as image processing, training neural networks, and operations research.

Recently, some research works have used GPU to solve nonlinear PDEs [14], [15], which are pioneer works for GPU-based 3D heat transfer model. GPU-based 3D heat transfer model has high computational performance owing to massively parallel computations, which are appropriate for real-time applications in casting control and optimization [14]. From the above subsection, it is obvious that every iteration time requires one time gradient calculation, and the one time gradient calculation contains one time Jacobian matrix calculation. Each Jacobian matrix calculation requires four times PDEs solution with different boundary conditions, which are $\mathbf{T(u)}$, $\mathbf{T_{ave}(u + p_1)}$, $\mathbf{T_{ave}(u + p_2)}$, and $\mathbf{T_{ave}(u + p_3)}$. The calculation of PDEs is running on GPUs. Prior to the calculation of PDEs on GPUs, the CPU should transfer the data such as the boundary conditions $\mathbf{u} + \mathbf{p_1}$ from the CPU memory to the GPUs memory. After the GPU finishes the calculation of PDEs, the data $(\mathbf{T_{ave}(u + p)})$ will be copied back to the CPU memory. This process called one-around calculation will be repeated four times with different boundary conditions.

The copy memory operations occupy significant running time, which results in the GPU having considerable free time. Several researchers reported that the copy memory operations are a bottleneck [30] because they are slower than the computing operations. According to our practical experience, the ratio of the running time of the copy memory operation to the entire running time is 56%.

In this study, stream parallelism is applied to the Jacobian matrix calculation. The stream parallel implementation of this application relies on the overlapping of memory copies with the solution of the PDEs. Fig. 8 shows the sequences of the calculation operations of the presented SP-SJ method. This study endeavors to obtain $\mathbf{T_{ave}(u + p_2)}$ to its input data $\mathbf{u} + \mathbf{p_2}$ to the GPU in advance while the solution $\mathbf{T_{ave}(u + p_1)}$ is being executed on the GPU. The execution time with the SP-SJ method can be even more favorable than that without stream parallelism. The process of SP-SJ method is shown in Algorithm 3.

In this study, the implementation of CUDA/C++ kernel is the same as that of [12]. The details of CUDA kernel implementation can be found in [14].

CUDA kernels have been implemented in the calculation of Jacobian matrix (the step 4 of A lgorithm 3). The CUDA kernel is designed for the computation of one time iteration, which provides the temperature field at time step $t + k + 1$. The pseudocode of the main CUDA/C++ kernel reads.

```
   Calculate Temperature $T_{ave}(t + k + 1)$
1: ID is assigned to thread in the CUDA grid to a particular mesh node
2: Update the thermal conductivity, the density, and thermal conductivity
   coefficient in ID
3: Calculate the temperature at time step t+k+1 for ID
```

*Remark 3:* One round of calculation intends to use the GPU to solve the PDEs with given boundary conditions. Thus, two separate rounds of calculations allocate two identical data so that each round can independently work. Owing to the naturally parallel structure of the Jacobian matrix calculation, the stream parallels can work exceedingly satisfactory to accelerate the solution strategy of the dynamic optimization problem (conjugate gradient method).

*Remark 4:* The stability of the MPC of the nonlinear PDEs has been developed, but several assumptions with respect to these results are not satisfied in our dynamic PDEs system or

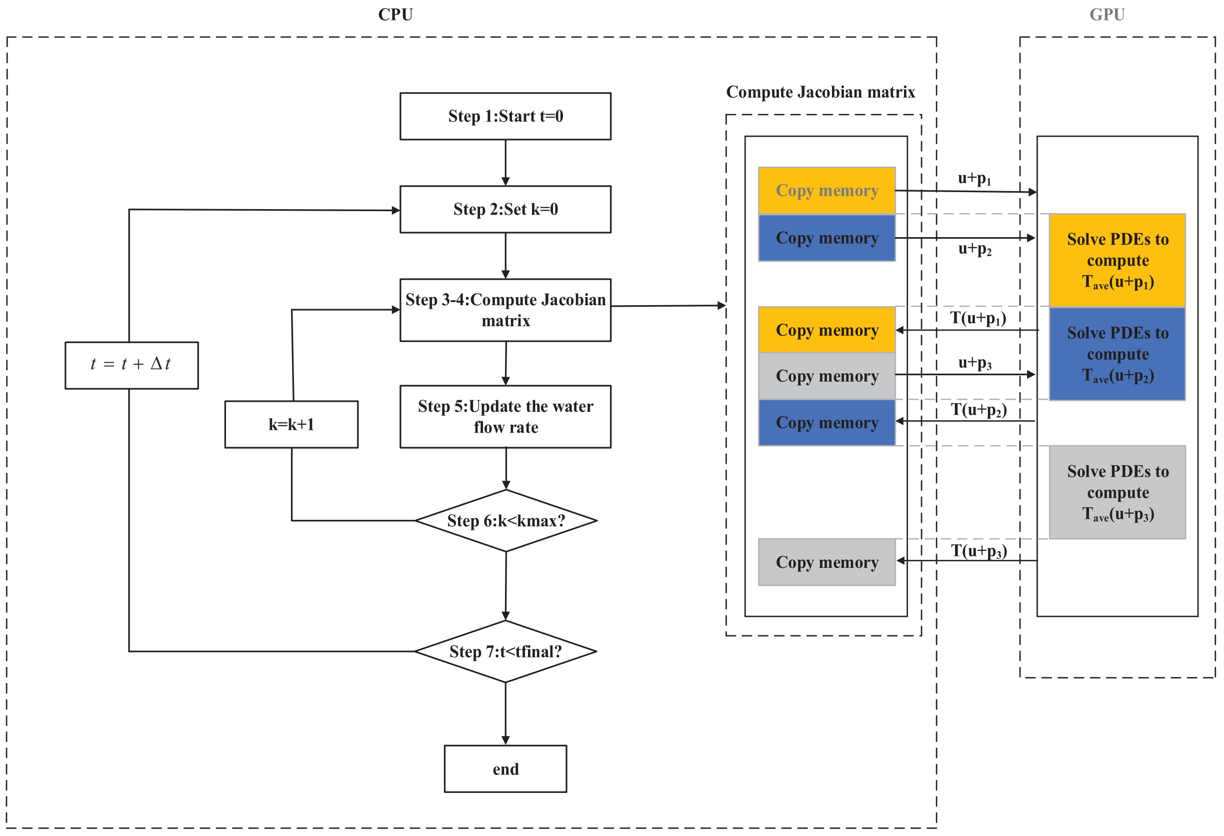
CPU
GPU
Step 1:Start t=0
Step 2:Set k=0
Step 3-4:Compute Jacobian matrix
Step 5:Update the water flow rate
Step 6:k<kmax?
Step 7:t<tfinal?
end
k=k+1
t = t + Δt
Compute Jacobian matrix
Copy memory
u+p1
u+p2
u+p3
T(u+p1)
T(u+p2)
T(u+p3)
Solve PDEs to compute Tave(u+p1)
Solve PDEs to compute Tave(u+p2)
Solve PDEs to compute Tave(u+p3)

**FIGURE 8. A flowchart of GPU-based model for SP-SJ method.**

it is difficult to verify whether these assumptions are satisfied. The stability is quite significant for MPC. Meanwhile, Sanders and Kandrot [28] reported that providing close-loop stability to the MPC of nonlinear PDEs system is challenging in theory. Thus, future work should be focused on the stability of MPC of this class of nonlinear PDEs. In terms of practical applications, the description from the input to the stable state conforms to the second law of thermodynamics. The operation of a close-loop controlled spray cooling system is safe only if the inputs (water flow rate) are bounded, which is satisfied in this study.

### *D. SIMULATION EXPERIMENT AND INDUSTRIAL APPLICATIONS*

Experiments 1-3 are simulation experiment and experiment 4 is an actual experiment. This section made this simulation experiment test the MPC scheme. Simulation experiments are necessary. Possible reason may be (1) Certain important features and capabilities of the control algorithm cannot be varied via an actual steel plant. (2) The MPC strategy should be tested based on a simulation model before it is conducted on an actual system.

All the experiments in Experiment 1-4 were conducted using Visual Studio 2015 and CUDA 8.0. The parallel-based heat transfer model was launched on NIVIDA Tesla P100. The traditional heat transfer model was launched on i7-3770. The heat transfer model is concerned with the casting conditions, including continuous caster parameters, and the physical parameters of steel billets shown in Table 4, 7-10. Uniform mesh is used. The time step is 0.02s. The number of mesh nodes are 25, 3000, 25 in $x$, $y$, $z$, respectively. The number of blocks is 3000 and the number of threads is $25 \times 25$.

*Experiment 1:* The experiment was conducted to compare the empirical methods and the MPC strategy. In the current subsection, the prediction horizon $N_p$ is 10. MPC is based on an iterative, finite-horizon optimization of a plant model. At time t, the current plant state is sampled and a cost minimizing control strategy is computed (via a numerical minimization algorithm) for a relatively short time horizon in the future: $[t, t + N_p\Delta t]$. Specifically, an online or on-the-fly calculation is used to explore state trajectories that emanate from the current state and find a cost-minimizing control strategy until time $t + N_p\Delta t$. The number of cooling section $N_z$ is 8. The target temperature is shown in Table 4. In this study, the target temperatures are obtained by the steel thermomechanical performance test are specified for the billet center and corner in every zone. The continuous caster operator can give the reference temperature profile. Based on the reference temperature profile, the candidate target temperature can be obtained. Then, the candidate target temperature are input into our algorithm. If the metallurgical principles can be satisfied, this candidate target temperature is set as the target temperature. If the metallurgical principles

**Algorithm 3** SP-SJ method

```
Input:   $\mathbf{T_{aim}}$
Output:  The optimal of water flow rate $\mathbf{u}^*$
1:   Let $t = 0$
2:   Let $k = 0$, set the initial water flow rate $\mathbf{u^k}$
     and the maximum iteration number $kmax$.
3:   Evaluation the steel billets temperature
     $\mathbf{T}(\mathbf{u}^k + \mathbf{p_1}, t)$,
     $\mathbf{T}(\mathbf{u}^k + \mathbf{p_2}, t)$, $\mathbf{T}(\mathbf{u}^k + \mathbf{p_3}, t)$, $\mathbf{T}(\mathbf{u^k}, t)$.
4:   Compute the Jacobian matrix based on (39).
5:   Update the water flow rate $\mathbf{u}^{k+1}$.
       if $k = 1$
         $\mathbf{u^{k+1}} = \mathbf{u^k} + \alpha_k \mathbf{p_k}$
         $\mathbf{p_k} = -\mathbf{g_k}$
       else
         $\mathbf{u^{k+1}} = \mathbf{u^k} + \alpha_k \mathbf{p_k}$
         $\beta_k = \frac{\|\mathbf{g_k}\|^2}{\|\mathbf{g_{k-1}}\|^2}$
         $\mathbf{p_k} = -\mathbf{g_k} + \beta_k \mathbf{p_{k-1}}$
6:   Let $k = k + 1$ and if $k < kmax$ go back to
     Step 3
7:   Let $t = t + \Delta t$ and if $t < tfinal$ go back to
     Step 2, else stop
```

TABLE 4. Target temperature.

| cooling section | target temperature(K) |
|---|---|
| 1 | 1243.5 |
| 2 | 1233.3 |
| 3 | 1224.9 |
| 4 | 1195.6 |
| 5 | 1166.5 |
| 6 | 1117.4 |
| 7 | 1066.8 |
| 8 | 1055.8 |

cannot be satisfied, the condidate target temperature should be adjusted, until the metallurgical principles can be satisfied. Before $t = 400s$, the operation condition is static, i.e., the temperature field of steel billets and the water flow rate of every cooling section are fixed. The casting speed is changed from $0.0167m/s$ to $0.02m/s$ at $t = 400s$. Both of the presented MPC and PI control [6] were applied to set the water flow rate. The experimental results are shown in Fig. 9-12.

Fig. 10 and Fig. 12 show the results of the PI control method. When the casting speed changes from $0.0167m/s$ to $0.02m/s$, the surface temperature of the steel billets cannot stay static. PI control only uses the measured surface temperature to adjust the water flow rate. However, the internal temperature cannot be measured, so the surface temperature will also be affected. Another drawback of PI control is that the PI control method only determines the water flow rate of $i$ th cooling section using the average temperature of $i$ th cooling section and does not consider the adjacent cooling sections. PI control requires a long time to adjust the water flow rate to reduce the temperature variations. On the contrary, from Fig. 9 and Fig. 11, the presented MPC approach considers the entire temperature field of steel billets to determine the water flow rate, so the surface temperature is more stable and the surface temperature fluctuations are relatively small when using the MPC approach.

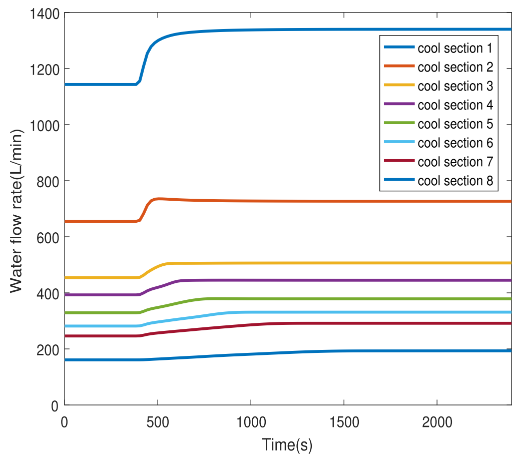

FIGURE 9. Water flow rate by using MPC.

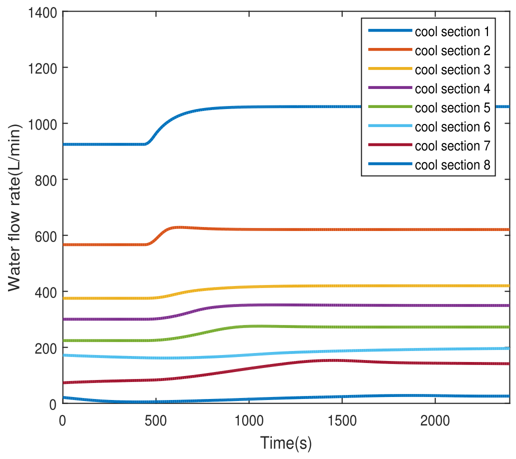

FIGURE 10. Water flow rate by using PI control.

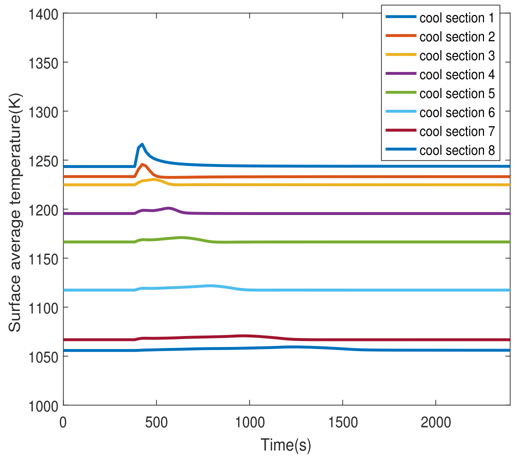

FIGURE 11. Surface average temperature by using MPC.

*Experiment 2:* The four different changes of casting speed status (case A: $0.02m/s$ to $0.03m/s$, case B: $0.025m/s$ to $0.03m/s$, case C: $0.022m/s$ to $0.026m/s$, case D: $0.025m/s$

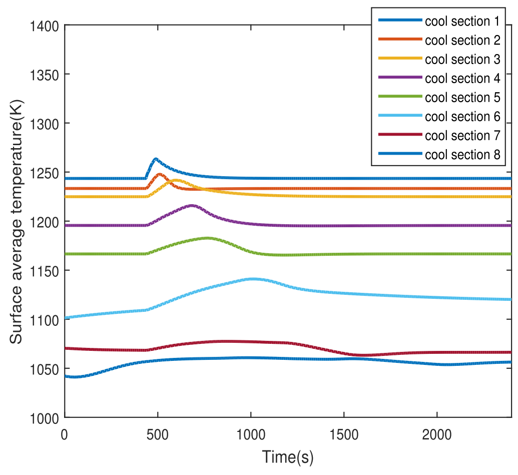

**FIGURE 12.** Surface average temperature by using PI control.

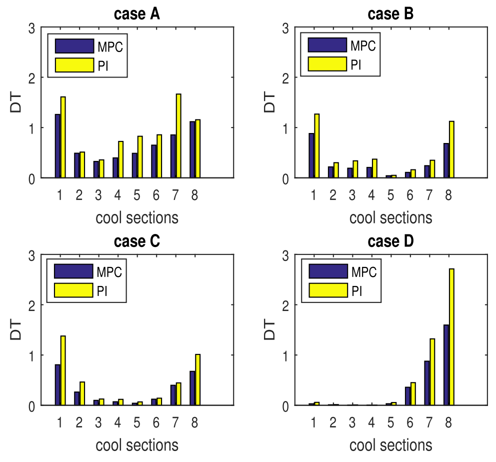

**FIGURE 13.** The DT of the average temperature with the different casting speed changes.

to $0.015m/s$) are conducted to verify the MPC approach. The **DT** is used to evaluate the control performance, which is defined as

$$DT_i = \frac{1}{t_f}\int_0^{t_f} \left|T_{ave,i}(t) - T_{aim,i}(t)\right| dt \tag{44}$$

The **DT** of those four cases are shown in Fig. 13. From Fig. 13, it is obvious that the temperature fluctuations are small when the casting speed changes abruptly.

*Experiment 3:* Experiment 1 and experiment 2 show the control performance of the presented MPC approach.

In this subsection, this work developed this experiment to demonstrate the running time between CPU sequential implementation, CPU parallel implementation, GPU parallel Jacobian computation (PJ) and SP-SJ method. This study implemented CPU parallel codes using OpenMP 2.0. OpenMP is an application programming interface for writing multi-threaded programs in C++, C and Fortran. In this experiment, eight threads are created in OpenMP. The results are listed in Table 5.

In Table 5, the running time represents the time during which a program is executing, and the relative run time is defined as the ratio of the run time for the wall-clock of the actual continuous casting process being simulated. The relative run time is a crucial performance indicator for real-time control and optimization. It should be mentioned that the solution approach for the MPC and its relative run time should be less than 1. From Table 5, the relative run time of SP-SJ method is less than 1. This provides the possibility of the application of MPC. MPC is the dynamic optimization. Thus, MPC starts to run the code to obtain the optimal solution at time $t_s$. $t_c$ is defined as the computational time of MPC. It takes $t_c$ to get the optimal solution, which implies that the optimal solution of $t_s$ can be obtained at time $t_s + t_c$; hence, there exists a delay. The shorter delay $t_c$, the better. Especially for continuous casting, if the water spray cannot be adjusted in time, the steel billets will produce a large temperature fluctuation. It is concluded that the presented SP-SJ method has the best performance and the complicated MPC approach

**TABLE 5.** A comparison of running time.

| Elements | run time(s) | relative run time |
|---|---|---|
| CPU sequential method | 16909.10 | 7.0455 |
| CPU parallel method | 3209.29 | 1.3372 |
| PJ method | 326.10 | 0.1359 |
| SP-SJ method | 83.78 | 0.0349 |

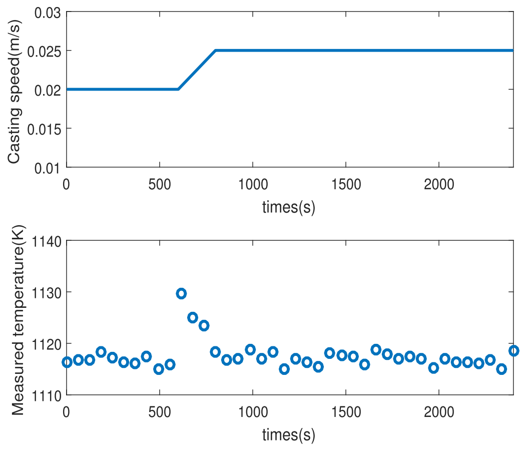

**FIGURE 14.** Measured surface temperature in cooling section 6.

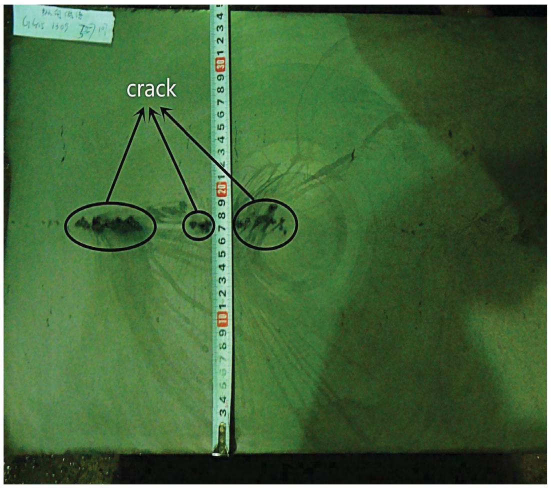

**FIGURE 15.** Micrographs before applying the new system.

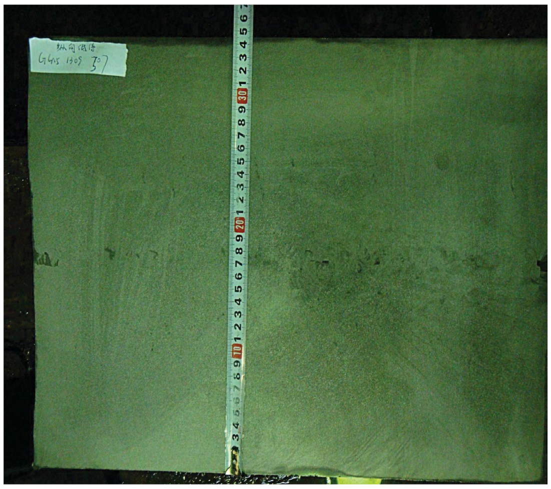
FIGURE 16. Micrographs after applying the new system.

TABLE 6. Major components of steel grade.

| Element | C | Si | Mn | P | S | Al |
|---|---|---|---|---|---|---|
| Wt | 0.08 | 0.03 | 0.2 | 0.025 | 0.025 | 0.05 |

TABLE 7. Process parameters and some physical properties of steel billets.

| Parameters | Value |
|---|---|
| Steel billets size x/z/y (m) | 0.25 × 0.25 × 28.6 |
| Tundish temperature$(K)$ | 1831.16 |
| Casting speed $(m \cdot s^{-1})$ | 0.02 |
| Density $(kg \cdot m^{-3})$ | $\rho = 7250$ |
| Specific heat $(J \cdot kg^{-1} \cdot K^{-1})$ | $c_s = 540$, $c_l = 500$ |
| A, B | A = 2258000, B = 342000 |
| m | m = 2.8 |
| $T_w$ $(K)$ | 303.15 |

can work exceedingly satisfactorily especially with respect to the large computation problem.

*Experiment 4:* This experiment is an actual experiment and all the experimental data are collected in Benxi Iron and Steel Corporation. The presented MPC approach has been implemented in an actual continuous casting process. As the casting speed changes from $0.02m/s$ to $0.025m/s$, the real-time measured surface temperature of the steel billets at the exit of the cooling section 6 are shown in Fig. 14.

From Fig. 14, the maximum fluctuation is $12.3K$. It is lower than the simulation case value of PI control $38.7K$ from Fig. 12, so the presented MPC approach can make the surface temperature stable. Moreover, the presented MPC approach also can improve the quality of the steel billets.

Fig. 15 and Fig. 16 show the micrographs before and after applying the new control system, respectively. The micrographs of the steel billets can be obtained via observation under a low-magnifying glass. The steel billets were severely cracked when an improper cooling water control scheme was used before applying the new control system. The proposed scheme can reduce the crack index (ranging from 0 to 1), which is used to evaluate the quality of the steel billets within the metallurgy industry. The reduction of data from 0.28 to 0.03 fully proves the improvement of billet quality.

## V. CONCLUSIONS

This paper is devoted to the design of a MPC approach for the continuous casting process. A new ATR-LM was developed to identify the unknown parameters in PDEs using the measured surface temperature and a new SP-SJ method was proposed to solve the dynamic optimization problem for MPC. The contributions of this study are shown as follows

(1)The presented ATR-LM method has better performance than the Cao and Landweber methods. Moreover, the unknown parameters are identified using this method and the prediction shell thickness of the corrected heat transfer model can follow the measured shell thickness obtained via the pin-shooting experiment. Furthermore, the convergence analysis of the ATR-LM is also provided.

(2)Utilizing the sparse structure of the Jacobian matrix, the SP-SJ method was presented to accelerate the computations based on the GPU, which provides the possibility to apply advanced real-time and optimization strategies to continuous casting.

(3)This study discusses the practical application problems and the case of real industrial study is also described. Moreover, the new construction of the MPC based on GPU can be extended to other complex industrial processes that include heat transfer model.

## APPENDIX

### A. PROCESS PARAMETERS AND SOME PHYSICAL PROPERTIES OF STEEL BILLETS.

Thermal conductivity is obtained by the following equations [31]

$$k_l = 35.0 - 0.3574w_{Cr} - 0.5116w_{Mo} - 0.0014w_{Ni} \tag{45}$$

$$\begin{aligned} k_s &= 20.76 + 0.009T - 3.2627w_C - 0.7598w_{Si} \\ &\quad - 0.1432w_{Mn} - 0.2222w_{Mo} + (0.0124 + 2.204 \times 10^{-4}T \\ &\quad + 1.078 \times 10^{-7}T^2 + 7.822 \times 10^{-4}w_{Cr} - 1.741 \times 10^{-7}T \\ &\quad \times w_{Cr})w_{Cr} + (-0.5860 + 8.354 \times 10^{-4}T - 1.368 \\ &\quad \times 10^{-7}T^2 + 1.067 \times 10^{-2}w_{Ni} \\ &\quad - 1.504 \times 10^{-7}Tw_{Ni})w_{Ni} \end{aligned} \tag{46}$$

Liquidus temperature and solidus temperature are obtained by the following equation [31]

$$\begin{aligned} T_l &= 1536.6 - (88w_C + 8w_{Si} + 5w_{Mn} + 25w_{Cu} + 1.5w_{Cr} \\ &\quad + 4w_{Ni} + 2w_{Mo} + 18w_{Ti}) \end{aligned} \tag{47}$$

$$\begin{aligned} T_s &= 1527 - (187.5w_C + 700w_S + 500w_P + 20.5w_{Si} \\ &\quad + 11.5w_{Ni} + 6.5w_{Mn} + 5.5w_{Al} + 2.0w_{Cr}) \end{aligned} \tag{48}$$

The results of ATR-LM and empirical formula are shown in Table 11. The empirical formula can be found in [2].

**TABLE 8.** Continuous caster parameters.

| Sections | Roll in zone | Roll radius(mm) | Length(m) |
|---|---|---|---|
| Mold | - | - | 0.8 |
| Spray 1 | 2 | 75 | 0.292 |
| Spray 2 | 2 | 75 | 1.177 |
| Spray 3 | 8 | 115 | 2.020 |
| Spray 4 | 4 | 135 | 1.545 |
| Spray 5 | 12 | 165 | 3.755 |
| Spray 6 | 12 | 180 | 4.002 |
| Spray 7 | 20 | 200 | 6.261 |
| Spray 8 | 32 | 225 | 8.729 |

**TABLE 9.** y.

| Sections | $\mathbf{y}(m)$ |
|---|---|
| $y_1$ | 0.8 |
| $y_2$ | 1.0 |
| $y_3$ | 2.27 |
| $y_4$ | 4.29 |
| $y_5$ | 5.93 |
| $y_6$ | 9.69 |
| $y_7$ | 13.71 |
| $y_8$ | 19.97 |
| $y_9$ | 28.90 |

**TABLE 10.** Water flow rate constraints.

| Zone | Water flow rate $(\underline{u}, \overline{u})(L/m^2 \cdot s)$ |
|---|---|
| 1 | (260.5, 520.7) |
| 2 | (130.1, 270.8) |
| 3-6 | (62.1, 180.1) |
| 7-8 | (30.5, 80.8) |

**TABLE 11.** The results of ATR-LM and empirical formula.

| Regions | $h$ (ATR-LM) | $h$ (Empirical formula) |
|---|---|---|
| Zone 1 | 1204.7 | 1213.2 |
| Zone 2 | 744.2 | 738.2 |
| Zone 3 | 422.5 | 409.9 |
| Zone 4 | 392.8 | 389.7 |
| Zone 5 | 331.2 | 321.5 |
| Zone 6 | 280.2 | 278.2 |
| Zone 7 | 249.8 | 249.7 |
| Zone 8 | 162.2 | 163.7 |

## B. CONVERGENCE ANALYSIS

*Lemma 1:* The search direction $\mathbf{d_k}$ satisfies with $c_1$, that is

$$m_k(\mathbf{0}) - m_k(\mathbf{d_k}) \geq \frac{1}{2}\left\|\mathbf{J_k}^T\mathbf{r_k}\right\| \min\left(\Delta_k, \frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|}{\left\|\mathbf{J_k}^T\mathbf{J_k}\right\|}\right) \tag{49}$$

*Proof 5.1:* Obviously, $\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k} \geq 0$. There are two cases. Consider the cases of $\|\mathbf{d_k}\| < \Delta_k$ and $\|\mathbf{d_k}\| = \Delta_k$ separately. For the former case, the following equations can be obtained

$$\frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^3}{\Delta_k\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}} \leq 1 \tag{50}$$

From the KKT conditions, the following equation can be obtained

$$\mathbf{d_k} = -\frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^2}{\Delta_k\left(\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}\right)}\mathbf{J_k}^T\mathbf{r_k} \tag{51}$$

Substitute $\mathbf{d_k}$ into $m_k$, the following equation can be obtained

$$\begin{aligned}
&m(\mathbf{0}) - m(\mathbf{d_k}) \\
&= -\frac{1}{2}\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}\frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^4}{\left(\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}\right)^2} + \frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^4}{\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}} \\
&= \frac{1}{2}\frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^4}{\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}} \geq \frac{1}{2}\frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^4}{\left\|\mathbf{J_k}\mathbf{J_k}^T\right\|\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^2} \\
&= \frac{1}{2}\frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^2}{\left\|\mathbf{J_k}\mathbf{J_k}^T\right\|} \geq \frac{1}{2}\left\|\mathbf{J_k}^T\mathbf{r_k}\right\| \min\left(\Delta_k, \frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|}{\left\|\mathbf{J_k}\mathbf{J_k}^T\right\|}\right)
\end{aligned} \tag{52}$$

For the next case, the following equation can be obtained.

$$\mathbf{d_k} = -\frac{\Delta_k}{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|}\mathbf{J_k}^T\mathbf{r_k} \tag{53}$$

Therefore, the following equation can be obtained

$$\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k} \leq \frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|}{\Delta_k} \tag{54}$$

Using (54), the following equation can be obtained

$$\begin{aligned}
&m(\mathbf{0}) - m(\mathbf{d_k}) \\
&= -\frac{\Delta_k}{\left(\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}\right)^2}\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k} + \frac{\Delta_k}{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|}\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^2 \\
&\geq -\frac{\Delta_k}{\mathbf{r_k}^T\mathbf{J_k}\mathbf{J_k}^T\mathbf{r_k}} + \frac{1}{2}\frac{\Delta_k^2}{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^2}\frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|^3}{\Delta_k} \\
&= \frac{1}{2}\Delta_k\left\|\mathbf{J_k}^T\mathbf{r_k}\right\| \geq \frac{1}{2}\left\|\mathbf{J_k}^T\mathbf{r_k}\right\| \min\left(\Delta_k, \frac{\left\|\mathbf{J_k}^T\mathbf{r_k}\right\|}{\left\|\mathbf{J_k}^T\mathbf{J_k}\right\|}\right)
\end{aligned} \tag{55}$$

The desired results are proved.

*Theorem 1:* Suppose that $\left\|\mathbf{J_k}\mathbf{J_k^T}\right\| \leq \beta$ for some constant $\beta$, that $f$ is bounded below on the level set S defined by

$$S = \{\mathbf{x} | f(\mathbf{x}) \leq f(\mathbf{x_0})\} \tag{56}$$

Lipschitz continuously differentiable in the neighborhood $S(R_0)$ for some $R_0 > 0$ is referred as

$$S(R_0) = \{\mathbf{x} | \|\mathbf{x} - \mathbf{y}\| < R_0\} \tag{57}$$

The following equation can be obtained

$$\liminf_{k\to\infty}\left\|\mathbf{J_k}^T\mathbf{r_k}\right\| = 0 \tag{58}$$

*Proof 5.2:* By performing some technical manipulation with the ratio $\rho_k$, we have

$$\begin{aligned}
|\rho_k - 1| &= \left|\frac{f(\mathbf{x_k}) - f(\mathbf{x_k} + \mathbf{d_k}) - m_k(\mathbf{0}) + m_k(\mathbf{d_k})}{m_k(\mathbf{0}) - m_k(\mathbf{d_k})}\right| \\
&= \left|\frac{m_k(\mathbf{d_k}) - f(\mathbf{x_k} + \mathbf{d_k})}{m_k(\mathbf{0}) - m_k(\mathbf{d_k})}\right|
\end{aligned} \tag{59}$$

Based on the Taylor's expression, the following equation can be obtained

$$\begin{aligned}
f(\mathbf{x_k} + \mathbf{d_k}) &= f(\mathbf{x_k}) + \left(\mathbf{J_k}^T\mathbf{r_k}\right)^T\mathbf{d_k} \\
&\quad + \int_0^1\left[J^T(\mathbf{x_k} + t\mathbf{d_k})\mathbf{r_k} - \mathbf{J_k}^T\mathbf{r_k}\right]^T\mathbf{d_k}dt
\end{aligned} \tag{60}$$

It follows from the definition of $m_k$ that

$$\begin{aligned}&|m(\mathbf{d_k}) - f(\mathbf{x_k}+\mathbf{d_k})|\\ &= \left|\frac{1}{2}\mathbf{d_k}^T\mathbf{J_k}^T\mathbf{J_k}\mathbf{d_k} - \int_0^1 \left[\mathbf{J^T}(\mathbf{x_k}+t\mathbf{d_k})\mathbf{r_k} - \mathbf{J_k}^T\mathbf{r_k}\right]^T \mathbf{d_k}dt\right|\\ &\le \frac{\beta_1}{2}\|\mathbf{d_k}\|^2 + \beta_2\|\mathbf{d_k}\|^2 \end{aligned} \tag{61}$$

Suppose for contradiction that there exists $\epsilon > 0$ and a positive index $K$ such that

$$\left\|\mathbf{J_k}^T\mathbf{r_k}\right\| \ge \varepsilon \tag{62}$$

for all $k > K$

From Lemma 1, for $k > K$, the following equation holds

$$\begin{aligned} m_k(\mathbf{0}) - m_k(\mathbf{d_k}) &\ge \frac{1}{2}\left\|\mathbf{J_k}^T\mathbf{r_k}\right\| \min\left(\Delta_k, \frac{\|\mathbf{J_k}^T\mathbf{r_k}\|}{\|\mathbf{J_k}^T\mathbf{J_k}\|}\right)\\ &\ge \frac{1}{2}\varepsilon \min\left(\Delta_k, \frac{\|\mathbf{J_k}^T\mathbf{r_k}\|}{\|\mathbf{J_k}^T\mathbf{J_k}\|}\right) \end{aligned} \tag{63}$$

Considering (61), (63) and $\|d_k\| \le \Delta_k$, we have

$$|\rho_k - 1| \le \frac{2\Delta_k^2(\beta/2+\beta_1)}{\varepsilon \min(\Delta_k, \varepsilon/\beta)} \tag{64}$$

$\bar{\Delta}$ is defined as follows for all $\Delta_k \le \bar{\Delta}$

$$\overline{\Delta} = \min\left(\frac{1}{2}\frac{\varepsilon}{(\beta_1/2+\beta_2)}, R_0\right) \tag{65}$$

Note that $\bar{\Delta} \le \frac{\varepsilon}{\beta_1}$, which implies that $\Delta_k \in [0, \bar{\Delta}]$. The following equation can be obtained

$$\min(\Delta_k, \varepsilon/\beta) = \Delta_k \tag{66}$$

Since,

$$\begin{aligned} |\rho_k - 1| &\le \frac{2\Delta_k^2(\beta_1/2+\beta_2)}{\varepsilon\Delta_k} = \frac{2\Delta_k(\beta_1/2+\beta_2)}{\varepsilon}\\ &\le \frac{2\bar{\Delta}(\beta_1/2+\beta_2)}{\varepsilon} \le \frac{1}{2} \end{aligned} \tag{67}$$

Based on (67), $\rho_k > \frac{1}{4}$. Because of $\rho_k > \frac{1}{4}$, $\Delta_{k+1} \ge \Delta_k$ whenever $\Delta_k$ falls below the threshold $\bar{\Delta}_k$. The reduction of $\Delta_k$ can happen in the presented algorithm only if

$$\Delta_k \ge \bar{\Delta} \tag{68}$$

Then, the following equation can be obtained

$$\Delta_k \ge \min\left(\Delta_K, \bar{\Delta}/4\right) \tag{69}$$

Suppose now that there is an infinite subsequence $K1$ such that $\rho_k \ge 1/4$ for $k \in K1$. For $k \in K1$ and $k \ge K$, from equation (63), we have

$$\begin{aligned} f(\mathbf{x_k}) - f(\mathbf{x_{k+1}}) &= f(\mathbf{x_k}) - f(\mathbf{x_k}+\mathbf{d_k})\\ &\ge \frac{1}{4}[m_k(\mathbf{0}) - m_k(\mathbf{d_k})]\\ &\ge \frac{1}{2}\varepsilon \min(\Delta_k, \varepsilon/\beta) \end{aligned} \tag{70}$$

Thus, $f$ is monotonically decreasing and bounded below. For $k \to \infty$, $f(\mathbf{x_k})$ must possess a limit. Based on the Cauchy's convergence principle,

$$\lim_{k\to\infty} f(\mathbf{x_k}) - f(\mathbf{x_{k+1}}) = 0 \ge \frac{1}{4}c_1\varepsilon \lim_{k\to\infty}(\min(\Delta_k, \varepsilon/\beta)) \ge 0 \tag{71}$$

Since

$$\lim_{k\in K1, k\to\infty} \Delta_k = 0 \tag{72}$$

This conclusion contradicts with (69). Hence, our original assertion (62) must be false, giving (58).

### C. ABBREVIATION

MPC: model predictive control
PDE: partial differential equation
GPU: graphic process unit
SCZ: secondary cooling zone
SP-SJ: stream parallel sparse Jacobian
PJ: parallel Jacobian
GA: genetic algorithm
LM: Levenberg-Marquardt
ATR-LM: adaptive trust-region Levenberg-Marquardt

## ACKNOWLEDGMENT

The authors would like to thank the research support of Benxi Iron and Steel Corporation.

## REFERENCES

[1] T. Mauder, S. Cenek, and S. Josef, "Optimal control algorithm for continuous casting process by using fuzzy logic," *Steel Res. Int.*, vol. 86, no. 7, pp. 785–798, 2015.
[2] J. Zhang, D. Chen, S. Wang, and M. Long, "Compensation control model of superheat and cooling water temperature for secondary cooling of continuous casting," *Steel Res. Int.*, vol. 82, no. 3, pp. 213–221, 2011.
[3] R. A. Hardin, K. Liu, C. Beckermann, and A. Kapoor, "A transient simulation and dynamic spray cooling control model for continuous steel casting," *Metall. Mater. Trans. B*, vol. 34, no. 3, pp. 297–306, Jun. 2003.
[4] J. Zhang, D.-F. Chen, C.-Q. Zhang, S.-G. Wang, and W.-S. Hwang, "Dynamic spray cooling control model based on the tracking of velocity and superheat for the continuous casting steel," *J. Mater. Process. Technol.*, vol. 229, no. 11, pp. 651–658, Mar. 2016.
[5] S. Chaudhuri, R. K. Singh, K. Patwari, S. Majumdar, A. K. Ray, A. K. P. Singh, and N. Neogi, "Design and implementation of an automated secondary cooling system for the continuous casting of billets," *ISA Trans.*, vol. 49, no. 1, pp. 121–129, Jan. 2010.
[6] X.-Y. Wang, Q. Liu, B. Wang, X. Wang, J.-S. Qing, Z.-H. Hu, and Y. H. Sun, "Optimal control of secondary cooling for medium thickness slab continuous casting," *Ironmaking Steelmaking*, vol. 38, no. 7, pp. 552–560, 2011.
[7] C. A. Santos, J. A. Spim, and A. Garcia, "Mathematical modeling and optimization strategies (genetic algorithm and knowledge base) applied to the continuous casting of steel," *Eng. Appl. Artif. Intell.*, vol. 16, nos. 5–6, pp. 511–527, 2003.
[8] B. Petrus, K. Zheng, X. Zhou, B. G. Thomas, and J. Bentsman, "Real-time, model-based spray-cooling control system for steel continuous casting," *Metall. Mater. Trans. B*, vol. 42, no. 1, pp. 87–103, Feb. 2011.
[9] S. Louhenkilpi, M. Mäkinen, S. Vapalahti, T. Räisänen, and J. Laine, "3D steady state and transient simulation tools for heat transfer and solidification in continuous casting," *Mater. Sci. Eng., A*, vols. 413–414, no. 3, pp. 135–138, Dec. 2005.
[10] S. Louhenkilpi, E. Laitinen, and R. Nieminen, "Real-time simulation of heat transfer in continuous casting," *Metall. Mater. Trans. B*, vol. 24, no. 4, pp. 685–693, Aug. 1993.

[11] L. Tang and X. Wang, "An improved particle swarm optimization algorithm for the hybrid flowshop scheduling to minimize total weighted completion time in process industry," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 6, pp. 1303–1314, Nov. 2010.

[12] S. Jiang, M. Liu, and J. Hao, "A two-phase soft optimization method for the uncertain scheduling problem in the steelmaking industry," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 416–431, Mar. 2017.

[13] L. Klimeš and J. Štětina, "A rapid GPU-based heat transfer and solidification model for dynamic computer simulations of continuous steel casting," *J. Mater. Process. Technol.*, vol. 226, pp. 1–14, Dec. 2015.

[14] L. Klimes and J. Stetina, "Unsteady model-based predictive control of continuous steel casting by means of a very fast dynamic solidification model on a GPU," *Mater. Technol.*, vol. 48, no. 4, pp. 525–530, Aug. 2014.

[15] D. Słota, "Identification of the cooling condition in 2-D and 3-D continuous casting processes," *Numer. Heat Transf., B, Fundam.*, vol. 55, no. 2, pp. 155–176, Sep. 2009.

[16] Y. S. Touloulian, R. W. Powell, C. Y. Ho, and P. B. Klemens, "Thermophysical properties of matter," in *Proc 4th Int. Conf. Continuous Casting*, 1972, vol. 645, no. 8, pp. 416–431.

[17] Y. Wang, X. Luo, Y. Yu, and Q. Yin, "Evaluation of heat transfer coefficients in continuous casting under large disturbance by weighted least squares Levenberg-Marquardt method," *Appl. Therm. Eng.*, vol. 111, no. 8, pp. 989–996, Jan. 2017.

[18] Y. A. Meng and B. G. Thomas, "Heat-transfer and solidification model of continuous slab casting: CON1D," *Metall. Mater. Trans. B*, vol. 34, no. 5, pp. 685–705, Oct. 2003.

[19] C. A. Santos, A. Garcia, C. R. Frick, and J. A. Spim, "Evaluation of heat transfer coefficients along the secondary cooling zones in the continuous casting of steel billets," *Inverse Problems Sci. Eng.*, vol. 14, no. 6, pp. 687–700, 2006.

[20] Y. Wang, X. Luo, Y. Yu, and H. Cui, "Optimal control of two-dimensional parabolic partial differential equations with application to steel billets cooling in continuous casting secondary cooling zone," *Optim. Control Appl. Methods*, vol. 37, no. 8, pp. 1314–1328, Nov./Dec. 2016.

[21] M. Baghban, S. H. Mansouri, and Z. Shams, "Inverse radiation-conduction estimation of temperature-dependent emissivity using a combined method of genetic algorithmand conjugate gradient," *J. Mech. Sci. Technol.*, vol. 28, no. 2, pp. 739–745, Feb. 2014.

[22] Y. Yu and X. Luo, "Estimation of heat transfer coefficients and heat flux on the billet surface by an integrated approach," *Int. J. Heat Mass Transf.*, vol. 90, no. 67, pp. 645–653, Nov. 2015.

[23] L. Cao and H. Han, "Convergence analysis of the homotopy perturbation method for solving nonlinear ill-posed operator equations," *Comput. Math. Appl.*, vol. 61, no. 8, pp. 2058–2061, Apr. 2011.

[24] D. Garmatter, B. Haasdonk, and B. Harrach, "A reduced basis Landweber method for nonlinear inverse problems," *Inverse Problems*, vol. 32, no. 3, Feb. 2016, Art. no. 035001.

[25] J. Yang, Z. Xie, J. Ning, W. Liu, and Z. Ji, "A framework for soft sensing of liquid pool length of continuous casting round blooms," *Metall. Mater. Trans. B*, vol. 45, no. 4, pp. 1545–1556, Aug. 2014.

[26] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.

[27] A. Steinboeck, D. Wild, and A. Kugi, "Nonlinear model predictive control of a continuous slab reheating furnace," *Control Eng. Pract.*, vol. 21, no. 4, pp. 495–508, Apr. 2013.

[28] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Reading, MA, USA: Addison-Wesley, 2010.

[29] A. R. Brodtkorb, T. R. Hagen, and M. L. Sætra, "Graphics processing unit (GPU) programming strategies and trends in GPU computing," *J. Parallel Distrib. Comput.*, vol. 73, no. 1, pp. 4–13, Jan. 2013.

[30] J. Miettinen, "Calculation of solidification-related thermophysical properties for steels," *Metall. Mater. Trans. B*, vol. 28, no. 2, pp. 281–297, Apr. 1997.

[31] M. Javurek, "CFD-Modeling of inclusion behavior during ladle exchange in a continuous casting system," in *Proc. 5th WCCM V*, Jul. 2002, pp. 1–13.

[32] B. Wang, Z.-P. Ji, W.-H. Liu, J.-C. Ma, and Z. Xie, "Application of hot strength and ductility test to optimization of secondary cooling system in billet continuous casting process," *J. Iron Steel Res. Int.*, vol. 15, no. 4, pp. 16–20, Jul. 2008.