

Design and Implementation of a Real-time Spray Cooling Control System for Continuous Casting of Thin Steel Slabs

Kai Zheng
Mittal Steel Company
3001 E Columbus Drive,
East Chicago, IN, 46312
Tel: 219-399-6494
E-mail: kai.zheng@mittalsteel.com

Bryan Petrus, Brian G. Thomas, Joseph Bentsman
University of Illinois
1206 West Green Street
61801 Urbana (IL)
Tel.: 217-333-6919
Fax: 217-244-6534
E-mail: bgthomas@uiuc.edu

Key words: Continuous casting, Secondary spray cooling, Real-time control, Heat Transfer Model, Software sensor, Temperature control, Thin slabs

INTRODUCTION

Robust and accurate control of secondary cooling is vital to the successful operation of any caster and the production of high quality products. This is especially important in thin slab casters, because high casting speed and a tight machine radius often requires carefully-prescribed temperature profiles to avoid various cracking problems. This can be achieved only by successful control of the secondary cooling to maintain these optimal temperature profiles through changes in casting process conditions. Thus, there is great incentive to implement control systems to optimize spray cooling.

Secondary cooling presents several challenges to control. Conventional feedback control systems based on hardware sensors have never been successful because emissivity variations from intermittent surface scale and the harsh environment of the steam-filled spray chamber makes optical pyrometers unreliable. Thin-slab casting is particularly difficult because the mold high casting speed shortens the response time. Thus, recent dynamic control systems have been developed based on real-time computational models. However, thin-slab casters are more difficult due to the thinner shell, the higher casting speed, and the increased relative importance of solidification in the mold, which is not easy to predict accurately. This paper presents a new real-time control system that has been developed to control spray cooling in thin-slab casters, and is now being implemented at the Nucor Steel casters in Decatur, Alabama.

Several previous attempts have been made to implement real-time dynamic control of cooling of continuous casters. It has long been recognized that the spray water flow should be adjusted so that each portion of the strand surface experiences the desired thermal history. This is especially important, and not always intuitive, during and after transients such as casting slowdowns during ladle exchanges. Okuno et al [1] and Spitzer et al [2] each proposed real-time model-based systems to track the temperature in horizontal slices through the slab to maintain surface temperature at 4-5 set points. Computations were performed every 20s and online feedback-control sensors calibrated the system. In practice, these systems have been problematic, owing to the unreliable sensors.

Barozzi et al developed a system to dynamically control both spray cooling and casting speed simultaneously [3]. Feedforward control was used to allow the predicted temperatures to match the setpoints, but their heat flow model was relatively crude, owing to the slow computer speed of that time. Optimizing spray cooling to avoid defects using fundamentally-based computational models was proposed by Lally [4]. At that time, the slow computer speed and inefficient fundamental computational models and control algorithms made online control infeasible.

In recent years, several open-loop model-based control systems have been developed to control spray water cooling under transient conditions for conventional thick-slab casters. These systems employ online computational models to ensure that each portion of the shell experiences the same cooling conditions. Spray water flow rates have been controlled in a thick slab casting plant using a 1-D finite difference model [5] that updates about once every minute. Hardin et al [6] and Leehuenkilpi and coworkers [7, 8] have developed 2-D and 3-D heat flow models to control spray cooling in real time. One Leehuenkilpi model, DYN3D, uses steel properties and solid fraction / temperature relationships based on multicomponent phase diagram computations [9]. Another, DYNCOOL, has been used to control spray cooling at Rautaruukki Oy Raahe Steel Works [10].

Although these model-based control systems are significant achievements, none of the models are robust enough for general use. Each must be tuned extensively on an individual caster, owing to non-general heat transfer coefficients and the use of ad-hoc heuristic methods, rather than rigorous control algorithms. None of the previous models uses sensor data input for the mold water cooling, which is readily available and reliable. Finally, none of these models has been applied to a thin slab caster, which has the control problems associated with higher speed, and where cooling in the mold is more important.

In this paper, a shell temperature/solidification model, yielding control-oriented real-time shell temperature profile estimates and predictions and further referred to as the software sensor, is first developed based on a comprehensive and efficient computational model: CON1D [11]. Then, 10 independently tuned proportional-integral-derivative (PID) controllers together with classical anti-windup [12] are designed to maintain the shell surface temperature profile at the desired setpoint for different casting speeds and different grades such that cracks are minimized. Based on the software sensor and the controller, a real-time spray cooling control system is developed that also includes a monitor interface that allows real-time monitoring of the shell surface temperature predictions, the predicted metallurgical length, control commands, casting conditions, etc., and TCP/IP server and client routines for communication among the software sensor, controller, monitor interface and the caster level 2 automation system. Simulation results demonstrate that significantly tighter shell surface temperature control performance is achieved. This control system is currently being implemented in the thin slab casters of Nucor Steel at Decatur, Alabama.

CONTROL DIAGRAM

The new dynamic control system for thin slab casters is based on the control diagram shown in Fig. 1.

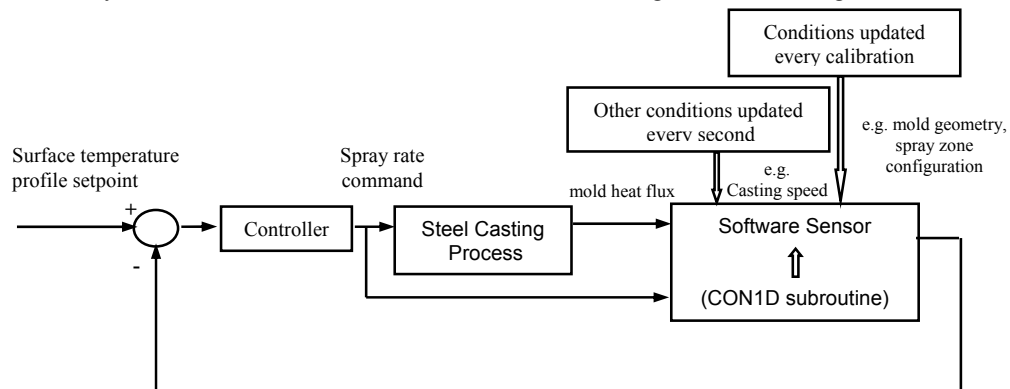


Fig. 1 Software sensor based control diagram

As seen in this diagram, the core of the system is a software sensor based on the CON1D model. The function of the software sensor is to provide a real-time shell temperature profile estimate and prediction based on all the available casting conditions. The latter can be categorized into two groups: 1) conditions updated every second, such as mold heat flux, casting speed, spray cooling rate commands, steel composition, etc; and 2) conditions updated only when the software sensor is calibrated, such as spray nozzle configuration, mold geometry, etc. The estimated/predicted shell temperature profile is then compared against a pre-determined surface temperature profile setpoint, which, as will be described later in this paper, is also a function of a variety of variables such as mold heat flux, etc. The mismatch between the estimate/prediction and the setpoint, i.e. the tracking error, is then sent to a dynamic controller to compute the water flow rate command required to drive the mismatch to zero. Finally, sending the computed command to the operating caster and also to the software sensor for estimation/prediction at the next second completes the loop.

SYSTEM ARCHITECTURE

To realize the control diagram shown in Fig. 1, a comprehensive control system has been developed. The main hardware of the system consists of two powerful workstations with dual Intel® Xeon® Processors and 2 gigabytes memory each. One workstation runs LINUX operating system and the other QNX real-time operating system, which are referred to as the LINUX and the QNX workstations, respectively. The communication between them and between the QNX workstation and the process level 2 system is realized by TCP/IP server and client programs, which is another part of the software subsystem of the control system. The software subsystem is composed of several programs listed in Table I, including the software sensor in Fig. 1, which is called CONONLINE.

Table I. Software programs in the control system

Program Name	Function	Language
CONONLINE	estimating/predicting the profile of shell temperature and thickness based on CON1D	Fortran
Controller	computing the required spray water flow rate to maintain temperature setpoint	C
Monitor	displaying in real-time shell temperature, thickness profile estimate/prediction, computed water flow rate and casting conditions	
TCP/IP server	working with TCP/IP client programs to transfer data between workstations	
TCP/IP client	working with TCP/IP server program to transfer data between workstations	

Note that the software programs Controller, Monitor, TCP/IP server, and TCP/IP client are written in C due to the excellent communication and graphics libraries available in this language. CONONLINE, on the other hand, is written in Fortran since it is based on the CON1D model, which was written in Fortran due to its powerful and efficient computation capabilities. Since no Fortran compiler is available for the QNX operating system, CONONLINE runs on the LINUX workstation. The monitor also runs on the LINUX workstation to exploit the graphing power GTK+ and GDK provide for LINUX. The controller runs on the QNX workstation to ensure real-time output of the spray rate command. The architecture of the system is shown in Fig. 2.

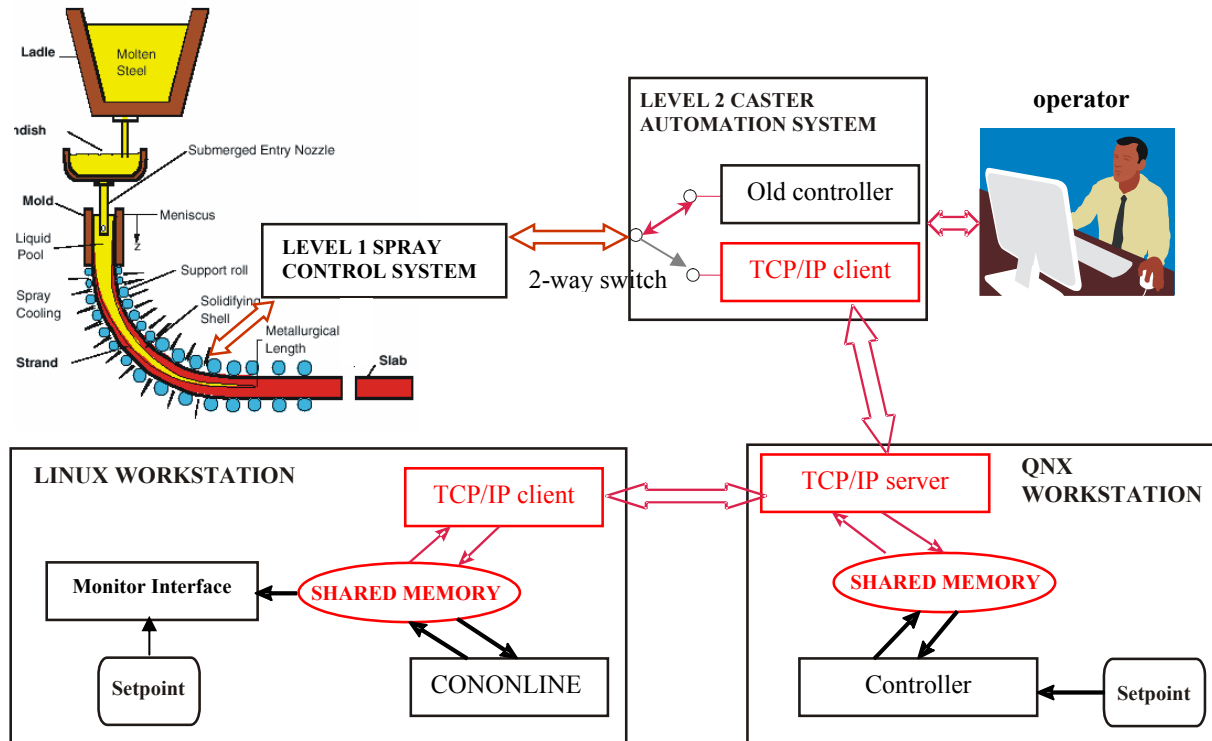


Fig. 2 Software sensor based control system architecture

The control system in Fig. 2 has two main modes of operation: 1) shadow mode and 2) implementation mode, depending on the 2-way switch in the level 2 system. In shadow mode, the switch allows one-way transfer of data from the caster to the TCP/IP client in the level 2 system, but the spray rate commands are only monitored, while the old controller controls the secondary cooling in the actual caster. This operating mode allows the control system to be tested. When the desired control system performance is obtained, the switch can be turned to implementation mode, disconnecting the old controller, and allowing the spray rate command from the new control system to take over.

When the control system runs in implementation mode, at each second, the level 2 system collects casting conditions such as casting speed, mold heat flux, etc. and sends them out via the TCP/IP client to the QNX workstation. The casting conditions are received by the TCP/IP server in the QNX workstation and then the TCP/IP client in the LINUX workstation. These data are then made available immediately to other programs running on both workstations via the shared memory in each workstation. The TCP/IP client and server programs are designed to exchange data in each shared memory approximately 10 times per minute and each transmission takes less than 20 ms. This transmission rate, being an order of magnitude higher than the sampling rate of the system, which is 1 second, provides sufficiently fast data sharing between different control system components.

The software sensor CONONLINE obtains casting conditions once every second from the shared memory and then accurately estimates/predicts shell temperature profile. The latter is then made available to the controller running on the QNX workstation by the shared memory and TCP/IP transmission. The controller simply reads the predicted shell temperature profile from the shared memory on the QNX workstation and computes the required spray water flow rate such that the estimated/predicted shell temperature profile is maintained at the temperature profile setpoint. The computed spray water flow rate control command is first saved to the shared memory on the QNX workstation, then transmitted to the actual caster for real-time cooling control. At the same time, the command is also transmitted to the LINUX workstation for CONONLINE to start estimation/prediction at the next time step, and to the monitor for the real-time display.

Although not an element of the control diagram in Fig. 1, the monitor is an important component in the control system because it provides real-time access to a variety of variables, estimation/prediction results, and commands, permitting operators and plant metallurgists to monitor the control system performance. The variables include the shell temperature profile and thickness prediction, water flow rate control commands, casting conditions, etc. Screen shots of the monitor interface as well as their detailed explanation will be discussed in a later section.

Notice that both the LINUX and the QNX workstations contain a copy of the shell surface temperature profile setpoint. The setpoint in the LINUX workstation is simply displayed in the monitor for comparison with the estimated/predicted shell surface temperature profile. The setpoint in the QNX workstation is used by the controller to compute the desired spray rate command (cf. Fig. 1).

A detailed description of each component of the control system is presented in the next section.

SYSTEM COMPONENTS

CON1D

CON1D is a simple but comprehensive fundamentally-based model of heat transfer and solidification of the continuous casting of steel slabs, including phenomena in both the mold and the spray regions [11]. The accuracy of this model in predicting heat transfer and solidification has been demonstrated through comparison with analytical solutions and plant measurements [11].

The simulation domain of CON1D, shown in Fig. 3, is a transverse slice through the strand thickness that spans from the center of shell surface of the inner radius to that of the outer radius. The CON1D model computes the temperature and solidification history of the slice as it traverses the path from the meniscus down through the spray zones to the end of the caster. An example of the predicted surface temperature history of the slice is given in Fig. 4. Notice that there are a number of temperature peaks and dips. The temperature dips are caused by water spray impingement and roll contact, whereas the temperature peaks occur where convection and radiation are the only mechanisms of heat extraction. The details of CON1D are summarized as follows.

CON1D solves the 1-D transient heat conduction equation within the solidifying steel shell:

$$\rho_{steel} C p_{steel}^* \frac{\partial T}{\partial t} = k_{steel} \frac{\partial^2 T}{\partial x^2} + \frac{\partial k_{steel}}{\partial T} \left(\frac{\partial T}{\partial x} \right)^2$$

This fundamentally-based model predicts shell thickness, temperature distributions in the mold and shell, total heat removal, heat flux profiles down the mold face, mold water temperature rise, ideal taper of the mold walls, and other phenomena. The calculation takes

advantage of the high Peclet number of the process, which renders axial heat conduction negligible. The effect of non-uniform distribution of superheat is incorporated using the results from previous 3-D turbulent fluid flow calculations within the liquid pool. The two heat flux profiles from superheat and surface extraction are shown with the model domain and typical temperature profile along the strand in the mold region in Fig. 5.

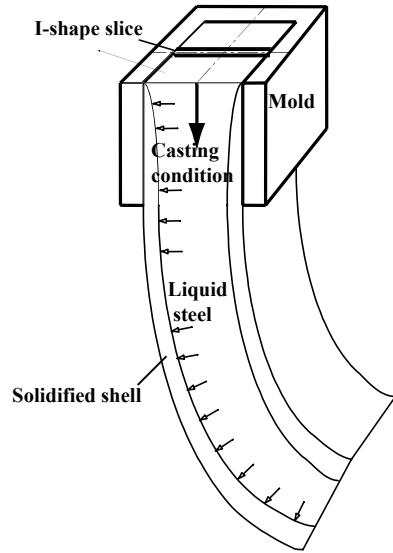


Fig. 3. CON1D simulation domain

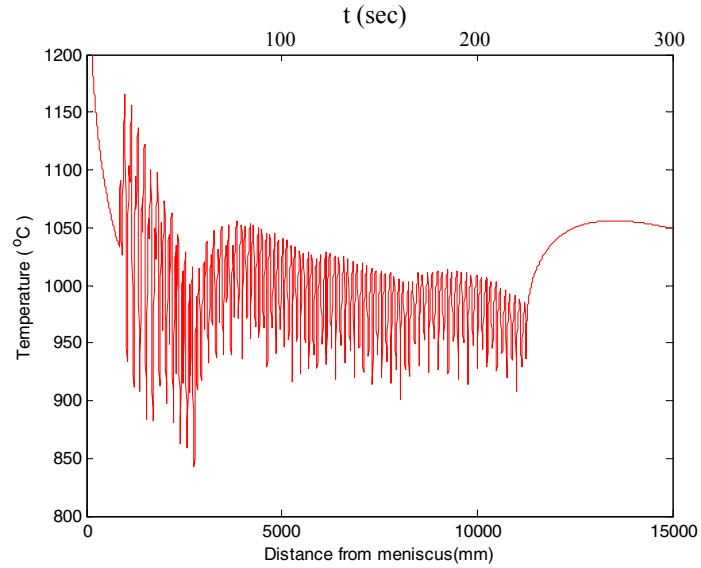


Fig. 4. Example of CON1D output: slice surface temperature history

Below the mold, heat flux from the strand surface varies greatly between each pair of support rolls according to spray nozzle cooling (based on water flux), h_{spray} , radiation, $h_{\text{rad_spray}}$, natural convection, h_{conv} , and heat conduction to the rolls, h_{roll} , as shown in Fig. 6. Incorporating these phenomena enables the model to simulate heat transfer during the entire continuous casting process. The heat extraction due to water sprays is a function of water flow [13] of the following form:

$$h_{\text{spray}} = A \cdot Q_{\text{water}}^c \cdot (1 - b \cdot T_{\text{spray}})$$

where Q_{water} ($\text{L}/\text{m}^2\text{s}$, where L stands for liters) is water flux in spray zones and T_{spray} is the temperature of the spray cooling water. In Nozaki's empirical correlation [14], $A=0.3925$, $c=0.55$, $b=0.0075$, which has been used successfully by other modelers [13-15]. Other relationships describing the variation of heat flux with nozzle type, nozzle-to-nozzle spacing, spray water flow rate, and distance of the spray nozzles from the strand surface are given in the literature [16-18]. Very recent experimental work has generated a larger database of fundamental heat transfer relationships for spray cooling [19, 20]. To avoid cracks, it is often necessary to keep the strand above a certain critical temperature, such as the AR3 temperature, $\sim 700^\circ\text{C}$. Recent methods to avoid overcooling of the strand include air mist cooling [20, 21].

Radiation, $h_{\text{rad_spray}}$ ($\text{W}/\text{m}^2\text{K}$) is calculated by:

$$h_{\text{rad_spray}} = \sigma \cdot \epsilon_{\text{steel}} (T_{\text{sK}} + T_{\text{ambK}}) (T_{\text{sK}}^2 + T_{\text{sprayK}}^2)$$

where T_{sK} and T_{sprayK} are T_{s} and T_{spray} expressed in Kelvin. Natural convection is treated as a constant input for every spray zone. For water-cooling only, it is not very important, therefore it is simplified to $8.7\text{W}/\text{m}^2\text{K}$ everywhere. Larger values can be entered for h_{conv} to reflect the stronger convection when there is air mist in the cooling zone. Heat extraction into the rolls is calculated based on the fraction of heat extraction to the rolls, f_{roll} , which is calibrated for each spray zone:

$$h_{\text{roll}} = \frac{(h_{\text{rad_spray}} + h_{\text{conv}} + h_{\text{spray}}) \cdot L_{\text{spray}} + (h_{\text{rad_spray}} + h_{\text{conv}}) \cdot (L_{\text{spray pitch}} - L_{\text{spray}} - L_{\text{roll contact}})}{L_{\text{roll contact}} \cdot (1 - f_{\text{roll}})} \cdot f_{\text{roll}}$$

A typical f_{roll} value of 0.05 produces local temperature drops beneath the rolls of about 100°C . Beyond the spray zones, heat transfer simplifies to radiation and natural convection.

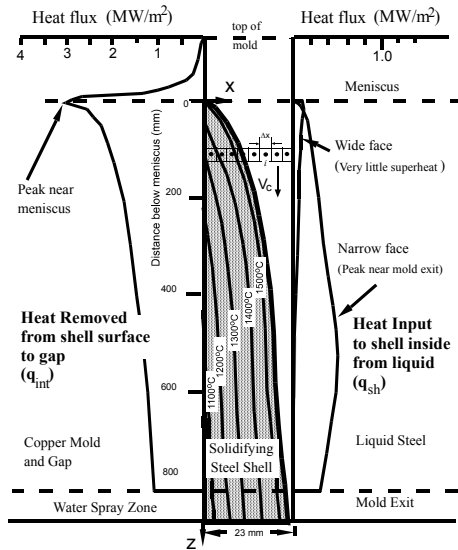


Figure 5. Model of solidifying steel shell domain showing typical isotherms and heat flux conditions

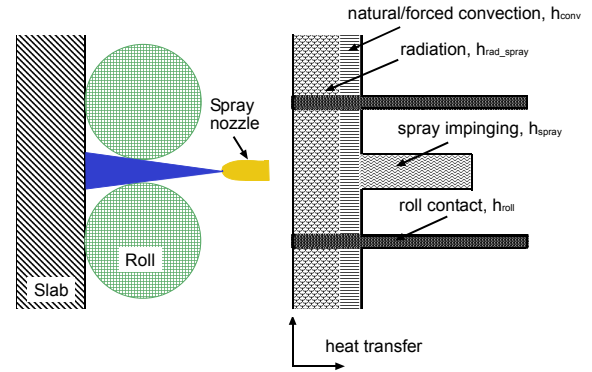


Figure 6. Schematic of spray zone region

CON1D has been calibrated offline to match experimental measurements on several different operating slab casters. This versatile modeling tool has been applied to a wide range of practical problems in continuous casters. With this tool used as the temperature predictor/estimator the closed-loop diagram of Fig. 1 takes the form shown in Fig. 7, where I.C. and B.C. stand, respectively, for initial and boundary conditions.

Theoretical Block Diagram

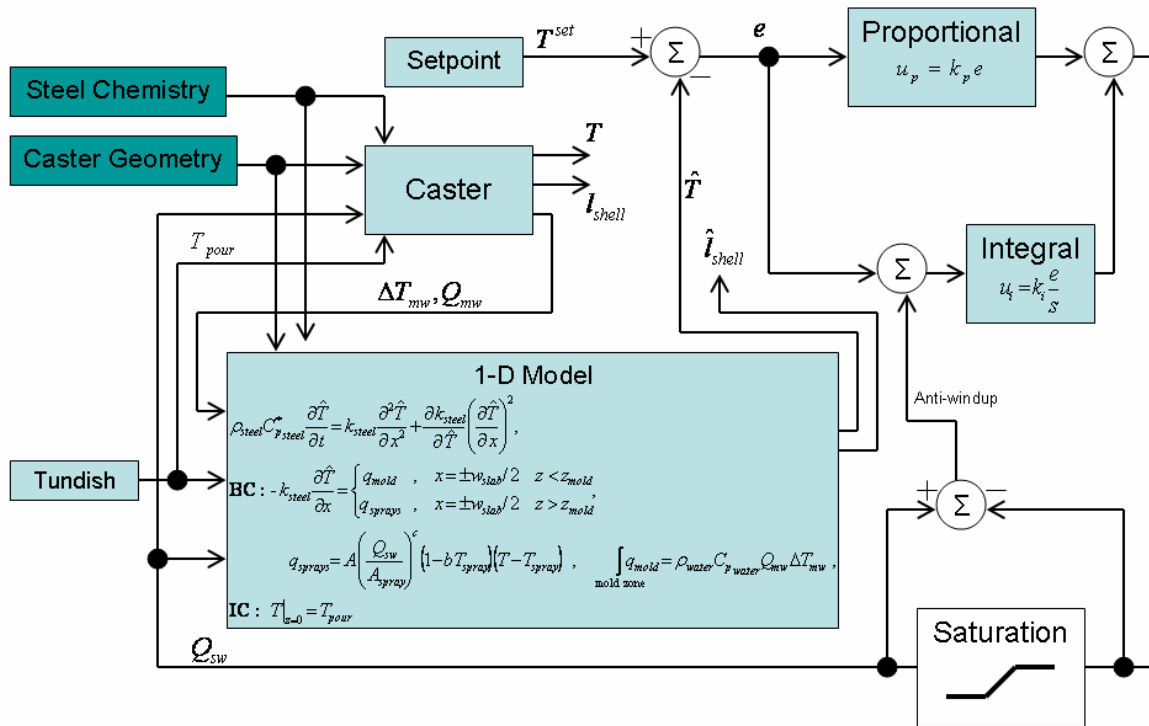


Figure 7. Closed-loop diagram with CON1D estimator/predictor

Software sensor --- CONONLINE

The goal of the control system is to control the shell surface temperature profile. Due to limitations on real-time computing power, only a 1-D temperature profile is considered in this version of the system. Thus, the function of the software sensor is to accurately predict the surface temperature profile along the entire caster at every given time. This temperature profile history is denoted $T(z, t)$, where z is the distance from the meniscus and t is the time.

When plotted on a two-dimensional t - z grid, the desired output domain of the software sensor is a horizontal line, shown in Fig. 8. For instance, at time t_0 the sensor must predict $T(z, t_0)$ for $0 \leq z \leq z_c$, where z_c is the caster length. However, if the casting speed is a constant, the surface temperature history included in the temperature history of the slice output from CON1D is a straight line in Fig. 8 having an angle θ with the t axis. It also follows that $\tan(\theta)$ equals the casting speed. Thus θ is always between 0 and 90 degrees. If the casting speed varies with time, then the CON1D output domain is not a straight line, but a curve in general.

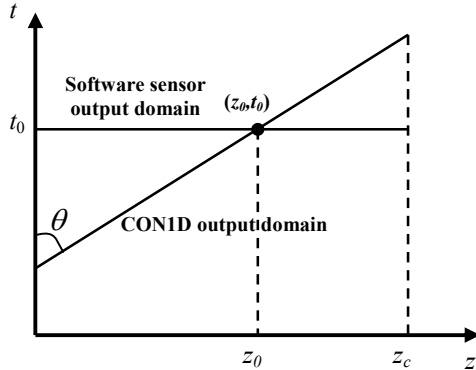


Fig. 8. CON1D and software sensor output domain comparison

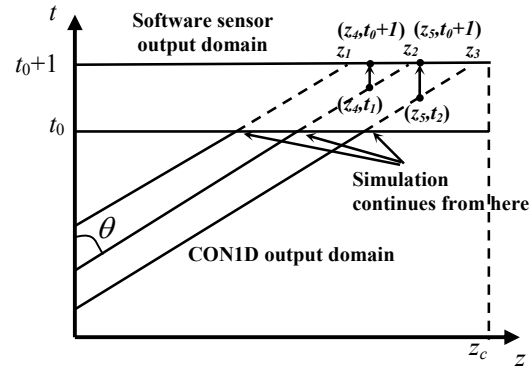


Fig. 9. Illustration of the incremental run of CON1D and shell surface temperature profile approximation

It is clear from Fig. 8 that the CON1D output cannot be directly used as the software sensor output. In fact, Fig. 8 shows that each complete run of CON1D contributes only one data point to the desired software sensor output, namely, $T(z_0, t_0)$. Thus, the results from many CON1D runs each starting at a different time must be interpolated, in order to achieve a single software sensor output. Data points in the temperature profile estimation/prediction which come directly from CON1D output without any approximation, such as $T(z_0, t_0)$, will be referred to as the *exact* estimation/prediction.

Since a typical run of CON1D takes about 0.6 seconds on the LINUX workstation, obtaining N data points in the shell temperature profile by having N complete runs of CON1D would require $0.6N$ seconds. This problem is addressed by running CON1D only for an incremental time period for each slice and approximating the shell surface temperature profile by interpolating between the latest temperature histories available from each CON1D slice, described as follows.

The software-sensor program CONONLINE was developed to manage the temperature profile histories of N different slices, each starting at the meniscus at a different time to achieve a fixed z -distance spacing between the slices along the caster. The CON1D program is converted to a subroutine of CONONLINE so that the simulation can restart from any location in the caster, given the previously-calculated and stored temperature distribution of that slice across the slab thickness at the given time. Exploiting this feature, exact temperature estimation/prediction can be obtained from N incremental CON1D simulations while requiring about the same computational time as one complete CON1D simulation. This is illustrated in Fig. 9 using $N=3$ slices for simplicity, although the real system typically uses $N=200$.

At time t_0 a temperature profile prediction is obtained from the simulation results of the N slices being tracked as follows. Exact shell surface temperature predictions are known at N locations from previous calculations, including N complete restarting temperature distributions. As shown in Fig. 9 for $N=3$, the simulation is restarted for each of the 3 slices and computed for 1 second of casting time. This produces exact shell surface temperature predictions at the 3 locations at the next second $t_0 + 1$,

giving $T(z_1, t_0 + 1)$, $T(z_2, t_0 + 1)$, and $T(z_3, t_0 + 1)$. If, for example, the current casting speed is 3 m/min, i.e. 50mm/s, then the simulation produces temperature histories for an additional 50mm. This efficient method of running N incremental slices allows the software sensor to achieve N exact shell surface temperature predictions. For a caster of 15m length, a maximum of $15000/50 = 300$ slices can be tracked with a total computation time of 0.6 second per second of casting time. In consideration of sparing more time for other programs and securely finishing the prediction within one second, the number of slices is chosen to be 200, which yields an approximate computation time of 0.4 seconds for the software sensor.

The management of the 200 slices proceeds as follows. First the inter-slice distance is determined, for the 15m caster as $15000/200 = 75$ mm. Slices to be simulated form a queue, shown in Fig. 11, along the casting direction where the first slice is the one entering the queue first, i.e. the one closest to the caster bottom, and the last slice is the one entering the queue last, i.e. the one closest to the meniscus. At caster startup, one slice is added to the queue and simulated, or “cast”, by CON1D. This slice becomes the first slice. At the time when the first slice is 75mm from the meniscus, a second slice is then added to the queue. Likewise, a third slice is added to the queue when the second slice is 75mm from the meniscus. This procedure is repeated until the startup is complete, i.e. the first slice is cast out of the caster, i.e. reaches the end of the caster. At this time the first slice is removed from the slice queue, and a new slice is added to the queue such that the total number of slices is always 200 after startup is complete. Note that all the slices are “cast” using CON1D with the same casting speed --- current casting speed and 200 slices are evenly distributed along the total caster length of 15 m.

With 200 slices, exact shell surface temperature estimates/predictions are obtained each second for 200 locations. The remaining problem is how to obtain temperature estimates/predictions for the locations in between. Linear interpolation would be very inaccurate because it neglects most of the temperature peaks and dips caused by spray impingement, roll contact, and convection, etc, causing errors as large as 100 degrees at some locations, as is shown in Fig. 10.

A better way is to approximate the temperature prediction at those locations at the current time by the latest temperature estimates/predictions at those locations, which are available from the temperature history of CON1D slices. This is illustrated in Fig. 9, where it can be seen that the temperatures $T(z_4, t_1)$ and $T(z_5, t_2)$ are used to approximate $T(z_4, t_0 + 1)$ and $T(z_5, t_0 + 1)$, respectively. The same approximation method applies to any other locations where exact predictions are not available. Based on this method, a control-oriented shell surface temperature profile $T(z, t)$ is obtained at any time t , given by

$$T(z, t) = T_{1D}^i(z), \text{ if } z_{i+1}(t) < z \leq z_i(t),$$

where $z_i(t), i = 1, 2, \dots, 200$ denotes the location of the i^{th} slice at time t , and $T_{1D}^i(z)$ is its temperature history from CON1D.

It can be seen that the approximation error introduced at location z_4 is the temperature change at this location from time t_1 to $t_0 + 1$, which is a function of slice spacing. It follows that slices should be evenly distributed to minimize the approximation error. The average error can be estimated by the temperature change at a given location in half of the time it takes for a slice to travel the distance of slice spacing. When 200 slices are evenly distributed for a 15m caster casting at 3m/min, it takes a slice 1.5 seconds to travel the slice spacing of 75mm. Thus, the average approximation error is the temperature change in $1.5/2 = 0.75$ seconds, which is usually less than ~30 degrees in transient and decays to zero as the temperature approaches steady state.

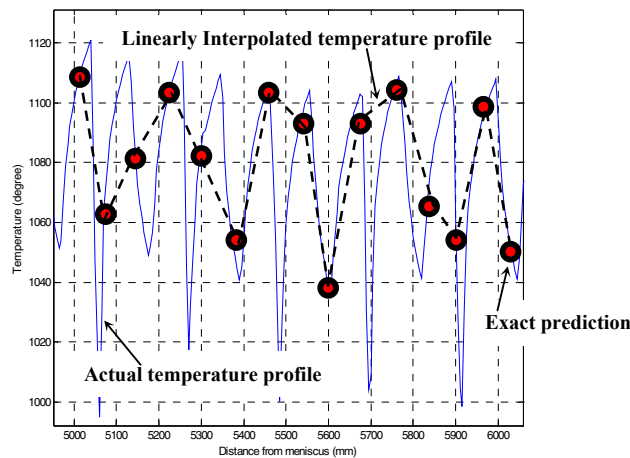


Fig. 10. Example of the actual temperature profile, the exact estimation/prediction and linearly interpolated temperature profile

Control algorithm

Because inter-slice heat transfer is negligible, decentralized single-input-single-output (SISO) controllers, which have no inter-controller interaction, can be used to control the spray-water flow rates to minimize the error between the CONONLINE prediction and the setpoint temperature profile. Single multi-input-multi-output (MIMO) controllers are another option, but are more complicated to design and implement and do not offer much better performance enhancement.

The temperature control problem in a given spray zone can be regarded as a disturbance rejection problem, in which the heat flux from the liquid core at the liquid/solid interface inside the strand can be treated approximately as a constant disturbance and the control goal is to maintain shell surface temperature under this disturbance. In light of this observation, the control law is simply chosen as the standard Proportional-Integral (PI) control. Here, the integral part is necessary for maintaining the surface temperature with no steady-state error under a constant setpoint and rejecting the constant disturbance. Derivative control, which is normally introduced to increase damping and stability margin, is not used since the system itself is well-damped.

Another important feature of the spray area configuration is that the individual spray nozzle headers are grouped into seven inner and seven outer spray zones according to nozzle location and control authority. This configuration is shown in Fig. 11, where nO and nI denote the n th outer and inner zones, respectively. Furthermore, zones 1O and 1I together can only be given only one spray rate command, so all the spray nozzles in zones 1O and 1I share the same spray rate. This is also the case for the subsequent inner and outer zone pairs up to zones 4O and 4I. The remaining 6 zones each have one command, a total of $4+6 = 10$ independent PI controllers are needed. The controller assignment for each zone is listed in Table II. The parameters of each controller are tuned separately to meet the control performance in each zone/zone pair. Further subdivision of the spray flow rates across the strand is prescribed from these 10 control signals according to slab width, and is assumed to vary with casting speed changes and other disturbances.

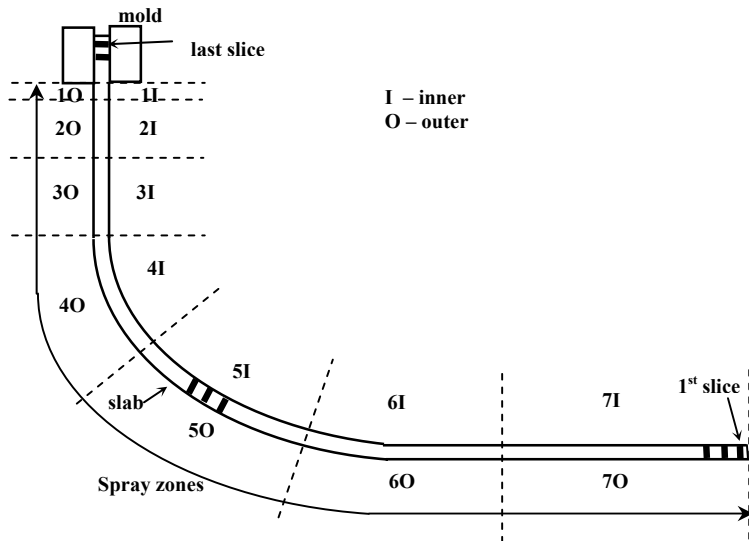


Fig. 11. Center spray area configuration

Table II. controller assignment

Zones	Controller assigned
1O+1I	Controller 1
2O+2I	Controller 2
3O+3I	Controller 3
4O+4I	Controller 4
5O	Controller 5
5I	Controller 6
6O	Controller 7
6I	Controller 8
7O	Controller 9
7I	Controller 10

In accordance with the aforementioned spray area configuration, the control algorithm is proposed as follows.

At each time second, t , the inner and outer radii shell surface temperature profile estimates/predictions, denoted as $T_{ip}(z, t)$ and $T_{op}(z, t)$, respectively, are obtained by the software sensor. Representing the shell surface-temperature profile setpoints on the inner and outer radii by $T_{is}(z, t)$ and $T_{os}(z, t)$, respectively, then

1. Form the tracking error for inner and outer radii respectively: $\Delta T_i(z, t) = T_{is}(z, t) - T_{ip}(z, t)$ and $\Delta T_o(z, t) = T_{os}(z, t) - T_{op}(z, t)$

2. Calculate the average tracking error for each zone: $\Delta T_i^j(t) = \frac{\int_{\text{zone } ji} (T_{is}(z,t) - T_{ip}(z,t)) dz}{\int_{\text{zone } ji} dz}$, $j = 1, 2, \dots, 7$, and

$$\Delta T_o^j(t) = \frac{\int_{\text{zone } jo} (T_{os}(z,t) - T_{op}(z,t)) dz}{\int_{\text{zone } jo} dz}, \quad j = 1, 2, \dots, 7$$

3. Calculate the spray rate command, $r^j(t)$, $j = 1, 2, \dots, 10$, for each controller using the PI control law as follows:

$$r^j(t) = k_p^j [\Delta T_i^j(t) + \Delta T_o^j(t)] + k_i^j \int_0^t [\Delta T_i^j(\tau) + \Delta T_o^j(\tau)] d\tau, \quad j = 1, 2, 3, 4. \text{ And}$$

$$r^5(t) = k_p^5 \Delta T_o^5(t) + k_i^5 \int_0^t \Delta T_o^5(\tau) d\tau,$$

$$r^6(t) = k_p^6 \Delta T_i^5(t) + k_i^6 \int_0^t \Delta T_i^5(\tau) d\tau,$$

$$r^7(t) = k_p^7 \Delta T_o^6(t) + k_i^7 \int_0^t \Delta T_o^6(\tau) d\tau,$$

$$r^8(t) = k_p^8 \Delta T_i^6(t) + k_i^8 \int_0^t \Delta T_i^6(\tau) d\tau,$$

$$r^9(t) = k_p^9 \Delta T_o^7(t) + k_i^9 \int_0^t \Delta T_o^7(\tau) d\tau, \text{ and}$$

$$r^{10}(t) = k_p^{10} \Delta T_i^7(t) + k_i^{10} \int_0^t \Delta T_i^7(\tau) d\tau, \text{ where } k_p^j \text{ and } k_i^j, \quad j = 1, 2, \dots, 10, \text{ are the proportional and the integral gains, respectively, for each controller.}$$

Furthermore, classical anti-windup [12] is adopted to avoid integrator windup should the transient control commands become negative or exceed the maximum spray rates. Specifically, if $r^j(t) < 0$, then a modified spray rate control command $r_m^j(t)$ is used in place of

$r^j(t)$ and given in the Laplace domain by $r_m^j(s) = \frac{s}{s + \alpha_j} r^j(s)$, where $r_m^j(s)$ and $r^j(s)$ are the Laplace transforms of $r_m^j(t)$ and $r^j(t)$, respectively, and α_j is the anti-windup tuning parameter for controller j. On the other hand, if $r^j(t) > r_{u\lim}^j$, where

$r_{u\lim}^j$ is the upper limit for controller j, then $r_m^j(s) = \frac{s}{s + \alpha_j} r^j(s) + \frac{\alpha_j}{s + \alpha_j} r_{u\lim}^j$ replaces $r^j(s)$. With this computational setting,

the closed-loop diagram of Fig. 1 takes the form shown in Fig. 12.

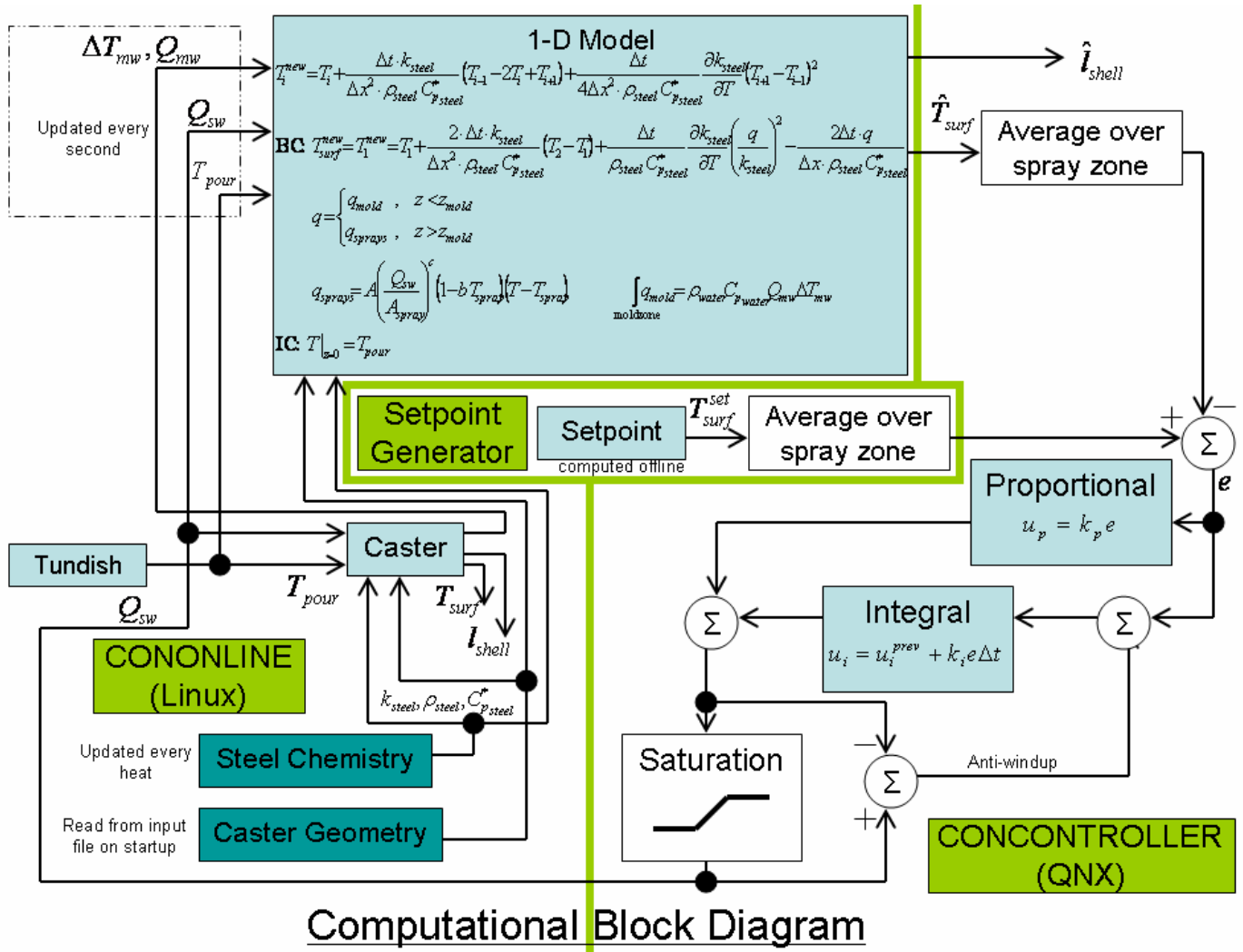


Figure 12. Closed-loop diagram with CONONLINE estimator/predictor

Monitor

The monitor is a very important tool, designed to display a variety of information, including spray rate commands, control performance, casting conditions, etc. Fig. 13 shows a typical screen shot of the monitor interface.

From Fig. 13, it can be seen that the monitor provides clear and immediate feedback to operators and engineers by displaying the following information graphically:

- Shell surface temperature profile estimation/prediction
- Shell surface temperature profile setpoint
- Shell thickness profile estimation/prediction
- Old controller command
- New controller command

This allows for:

- Monitoring of the control system performance
- Monitoring of the temperature profile, to ensure that quality limits are maintained
- Monitoring of the shell evolution and the metallurgical length, to ensure that the machine support length is never exceeded.
- Comparing the old and the new controller commands for better understanding of spray cooling

Casting conditions received can also be shown in the monitor, as can be seen in Fig. 14. This allows an easy check of the casting conditions and the TCP/IP server and client operation.

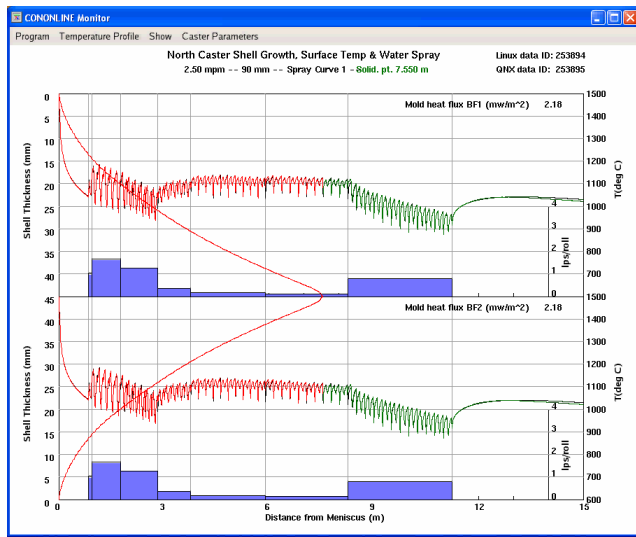


Fig. 13 Monitor interface displaying on both radii: the shell surface temperature profile estimation/prediction, its corresponding setpoints, the spray-water flow rates commanded and their setpoints, the shell thickness profile prediction, and other information.

Table of Nomenclature for Fig 12

$k_{steel}, \rho_{steel}, C_{p steel}^*$	Thermodynamic variables for steel
Q_{sw}	Spray water volumetric flow rate
Q_{water}	Water flux on spray area
A, b, c	Empirical coefficients from [14]
A_{spray}	Area spray impinges on slab
Q_{mw}	Mold water volumetric flow rate
ΔT_{mw}	Mold water temperature change
T_{pour}	Pour temperature
T_{surf}	Surface temperature of slab
T_{surf}^{set}	Surface temperature setpoint
$\hat{T}_{surf}$	Surface temperature estimate
e	Temperature error
l_{shell}	Metallurgical length
k_p, k_i	Controller gains

Setpoint Generation

The shell surface temperature profile setpoint is the desired shell surface temperature profile, which often changes with the steel grade. Previous theoretical knowledge about optimizing spray cooling has been defined in terms of desired temperature profiles at steady state casting conditions. Plant experience is defined by a set of spray water set points that is known by empirical means to avoid problems. To combine these two types of knowledge, surface temperature profile setpoints are generated in this work by running CON1D at a given casting speed with a good set of spray rates, and applying the same resulting surface temperature profile as the setpoint for all casting speeds. Taking 3.56 m/min as a typical successful casting speed, temperature setpoints were obtained for 8 different spray patterns, each defined by a set of spray rates at 3.56 m/min and corresponding to a set of grades for that pattern.

It is important to maintain the desired temperature profile setpoint for a given steel grade in order to avoid the thermal stresses caused by changing temperature gradients and to avoid the steel ductility troughs associated with particular temperature ranges. Both of these factors help to avoid cracks, even through casting condition changes, startup, and tailout. This approach is successful in part because steel thermal properties are relatively independent of steel grade and casting speed. A problem, however, arises due to mold heat flux variations. These variations cause the slab temperature at mold exit to change significantly with varying steel grade and casting speed. Fixing the temperature setpoint below the mold would cause the spray rates, and corresponding shell surface temperature, to change sharply, especially in the first 2 spray zones below the mold. The associated sharp changes in shell surface temperature are undesirable.

To avoid this problem, the temperature profile setpoints are chosen to vary with mold heat flux, and consequently also on mold exit temperature. Each setpoint is generated as follows. First, the expected mean mold heat flux Q_{mold0} is estimated as a function of mold powder and casting speed, as

$$Q_{mold0} = 4.63 \cdot 10^6 \mu^{-0.09} T_{flow}^{-1.19} V_c^{0.47} \left\{ 1 - 0.152 \exp \left[- \left(\frac{0.107 - \%C}{0.027} \right)^2 \right] \right\}$$

where:

Q_{mold0} is the mean mold heat flux (MW/m²),
 μ is the powder viscosity at 1300° C, (Pa-s),
 T_{flow} is the melting temperature of the mold flux (°C),
 V_c is the casting speed (m/min),
 $\%C$ is the carbon content.

Then, for spray zones 1-4, five different temperature profile setpoint curves are generated using CON1D with 70% Q_{mold0} , 85% Q_{mold0} , 115% Q_{mold0} , and 130% Q_{mold0} . The effect of mold heat flux variations diminishes with distance down the strand, so the temperature setpoint for the remaining zones 5 through 7 uses the original fixed setpoint corresponding with Q_{mold0} . The 5 temperature setpoint curves for spray pattern 1 are shown in Fig. 14.

It can be seen that these setpoints produce mold exit temperature ranging from 850 ~1250 °C. For a particular mold heat flux, or mold exit temperature, interpolation is used to generate a temperature setpoint profile from the setpoints in Fig. 14 such that the predicted mold exit temperature is equal to the mold exit temperature setpoint. The impact of mold heat flux variations is thus evenly distributed over the first 4 spray zones and thereby avoids causing any sharp spray rate changes or surface temperature changes.

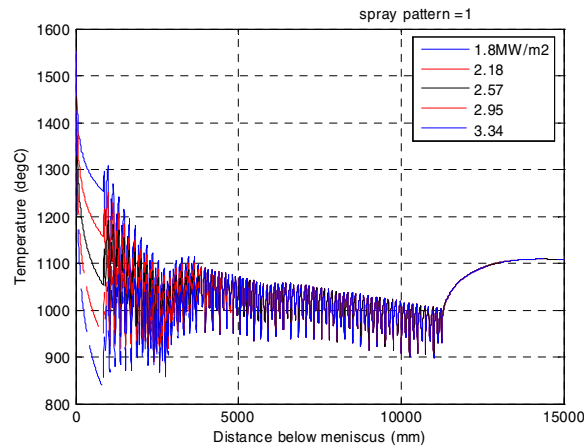


Fig. 14. The 5 temperature setpoint curves for spray pattern 1

SIMULATION RESULTS

The monitor program can produce animations to illustrate the caster response to arbitrary scenarios involving changing casting conditions. Initial efforts have focused on evaluating the accuracy of the new control system, especially in comparison to the old control system of fixing spray water flow rates with casting speed. For example, Fig. 15 compares the surface temperature histories extracted from the software sensor predictions at 4 points down the caster for a sudden drop in casting speed from 3.5 m/min to 3.0 m/min (at $t=0$). The simulations were run for 0.048 % carbon steel. To improve visibility, the particular points are chosen to have decreasing temperature with distance down the caster, although this is not generally the case. The wiggles in the temperature histories are due to the interpolation approximations for these 3 real-time simulations, which are taken for realistic operating conditions. The initial increase in temperature at $t=0$ in all three graphs is due to a decrease in the mold heat removal rate from 2.55 MW/m² at the higher casting speed to 2.37 MW/m² at the lower speed. In online mode, the control system receives mold heat flux data from the caster every second. In the offline mode used to produce these simulations, the system estimates this data based on the casting speed.

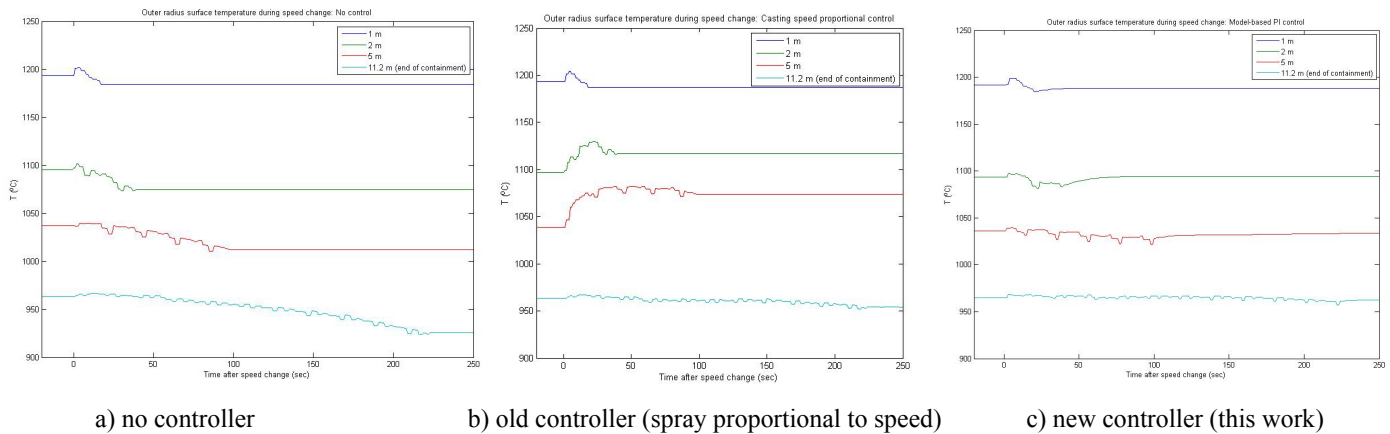


Fig. 15. Variation in shell surface temperature predicted for three different spray water control systems

With no controller, spray water flow rates remain constant with time, so the decrease in casting speed causes higher heat extraction at any given distance down the caster, and the surface temperatures all eventually drop. The time delay for the transition to the new lower steady-state temperature varies with distance down the caster. Steady state is not reached until steel starting at the meniscus at time $t=0$ finally reaches the given point in the caster after being cast entirely under the new conditions. Thus, points near to mold exit react quickly to the change, while points lower in the caster are affected by the changing upstream temperature history for a long time.

With a controller that increases spray water in proportion to casting speed, the temperature response varies in time, magnitude, and direction. Compared with the original temperatures, the new steady-state temperatures at long times are sometimes lower (e.g. at 1 m), sometimes higher (e.g. 2 m and 5 m), and sometimes almost the same (11.2 m), depending on how well the proportional drop in water flow compensates for the actual drop in heat extraction. Moreover, the temperature change is sudden, as the sudden large change in spray water takes immediate effect, causing sharp temperature gradients and stress. Temperature overshoots, as the strand is too hot for the lower water flow rates. Eventually, the temperature drops to reach steady state, with a time delay similar to that for no controller.

With the new controller developed in this work, the temperature transition is much smoother. The magnitude of the decrease in spray water flow rates are controlled to vary with spray zone, in accordance with the predicted changes in heat extraction predicted by the fundamental, calibrated CONONLINE model to maintain the surface temperature. Moreover, the changes in water flows are applied gradually, according to the local position. The result is almost constant temperature with time, neglecting the numerical glitches, which are not experienced by the steel. This validates that the controller is doing an accurate job. The quality of the control system now depends on the accuracy of the software sensor calibration to match the real caster. Work is proceeding to measure heat transfer, both with fundamental lab experiments, and with optical pyrometers and other experiments in the commercial steel thin-slab caster.

SUMMARY

Maintaining the shell surface temperature profile under transient conditions by spray water cooling in continuous casting of steel is important to minimize surface cracks. For this purpose, a real-time spray-cooling control system is being implemented on a commercial caster that includes 1) a software sensor for accurate estimation/prediction of shell surface temperature, 2) control algorithm and data checking subroutines for robust temperature control, 3) TCP/IP Server and Client programs for communicating between these two software components and the caster, and 4) a real-time monitor to display the predicted shell surface temperature profiles, water flow rates, and other important operating data. Simulation results demonstrate that the new control system achieves better temperature control performance than conventional systems.

ACKNOWLEDGEMENTS

Ron O'Malley, Matthew Smith, and Terri Morris from Nucor Decatur are gratefully acknowledged for their unwavering support of this work. National Science Foundation support under Grant DMI 05-00453 and the Continuous Casting Consortium is acknowledged.

REFERENCES

1. K. Okuno, H. Naruwa, T. Kuribayashi, T. Takamoto, *Iron Steel Eng.*, Vol. 12 (4), 1987, 34-38.
2. K.-H. Spitzer, K. Harste, B. Weber, P. Monheim, K. Schwerdtfeger, *Iron Steel Inst. Jpn*, Vol. 32 (7), 1992, 848-856.
3. S. Barozzi, P. Fontana and P. Pragliola, *Iron Steel Eng*, Vol. 11, 1986, 21-26.
4. B. Lally, L. Biegler and H. Henein, *Met. Trans. B*, Vol. 21B, 1990, 761-770.
5. K. Dittenberger, K. Morwald, G. Hohenbichler, U. Feischl, "DYNACS Cooling Model - Features and Operational Results," in *VAI 7th International Continuous Casting Conference*, Voest-Alpine Ind., (Linz, Austria), 1996, 44.1-44.6.
6. R.A. Hardin, K. Liu, A. Kapoor, C. Beckermann, "A Transient Simulation and Dynamic Spray Cooling Control Model for Continuous Steel Casting," *Metal. & Material Trans.*, Vol. 34B (June), 2003, 297-306.
7. S. Louhenkilpi, E. Laitinen and R. Nienminen, *Metal. & Material Trans.*, Vol. 24B, 1999, 685-693.
8. S. Louhenkilpi, J. Laine, T. Raisanen, T. Hatonen, "On-Line Simulation of Heat Transfer in Continuous Casting of Steel," in *2nd Int. Conference on New Developments in Metallurgical Process Technology*, (Riva del Garda, Italy, 19-21 Sept, 2004), 2004.
9. T. Raisanen, S. Louhenkilpi, T. Hatonen, J. Toivanen, J. Laine, M. Kekalainen, "A Coupled Heat Transfer Model for Simulation of Continuous Casting," in *European Congress on Computational Methods in Applied Sciences and Engineering*, 2004.
10. M. Jauhola, E. Kivela, J. Kontinen, E. Laitinen, S. Louhenkilpi, "Dynamic Secondary Cooling Model for a Continuous Casting Machine," in *6th International Rolling Conference*, Vol. 1, Verein Deutscher Eisenhüttenleute, P.O. Box 105145, Dusseldorf, D-40042, Germany, (Dusseldorf; Germany, 20-22 June, 1994), 1994, 196-200.
11. Y. Meng and B. G. Thomas, "Heat-Transfer and Solidification Model of Continuous Slab Casting: CON1D," *Metallurgical and Materials Transactions*, Vol. 34B, Oct. 2003.
12. C. Edwards, I. Postlethwaite, "Anti-windup and Bumpless Transfer Schemes", *UKACC International Conference on CONTROL*, 1996, 394-399.
13. J.K. Brimacombe, P.K. Agarwal, S. Hibbins, B. Prabhaker, L.A. Baptista, "Spray Cooling in the Continuous Casting of Steel," Vol. 2, *Continuous Casting*, 1984, 109-123.
14. T. Nozaki, "A Secondary Cooling Pattern for Preventing Surface Cracks of Continuous Casting Slab," *Trans. ISIJ*, Vol. 18, 1978, 330-338.
15. R.A. Hardin, H. Shen and C. Beckermann, "Heat Transfer Modeling of Continuous Steel Slab Caster Using Realistic Spray Patterns," *Modeling of Casting, Welding and Advanced Solidification Processes IX*, 2000, 190.
16. J.K. Brimacombe, P.K. Agarwal, S. Hibbins, B. Prabhaker, L.A. Baptista, "Spray Cooling in Continuous Casting," in *Continuous Casting*, Vol. 2, J.K. Brimacombe, ed. Iron and Steel Society, Warrendale, PA, 1984, 105-123.
17. E. Mizikar, *Iron Steel Engineer*, Vol. 47, 1970, 53-60.
18. L.K. Chiang, in *57th Electric Furnace Conference*, Iron & Steel Society, Warrendale, USA, (Pittsburgh, USA), 1999.
19. K. Tanner, "Comparison of Impact, Velocity, Dropsizes, and Heat Flux to Redefine Nozzle Performance in the Caster," in *MS&T Conference Proceedings*, B.G. Thomas, ed. Association for Iron and Steel Technology, Warrendale, PA, (New Orleans, Sept 24-28, 2004), 2004.
20. K. Kasperski, "Spray Cooling Results of Air/Mist Spray Nozzles with Reduced Air Volumes," in *MS&T Conference Proceedings*, B.G. Thomas, ed. Association for Iron and Steel Technology, Warrendale, PA, (New Orleans, Sept 24-28, 2004), 2004.
21. S.-M. Lee and S.-Y. Jang, *ISIJ International (Japan)*, Vol. 36 (Supplementary on Science and Technology of Steelmaking), 1996, 208-210.

