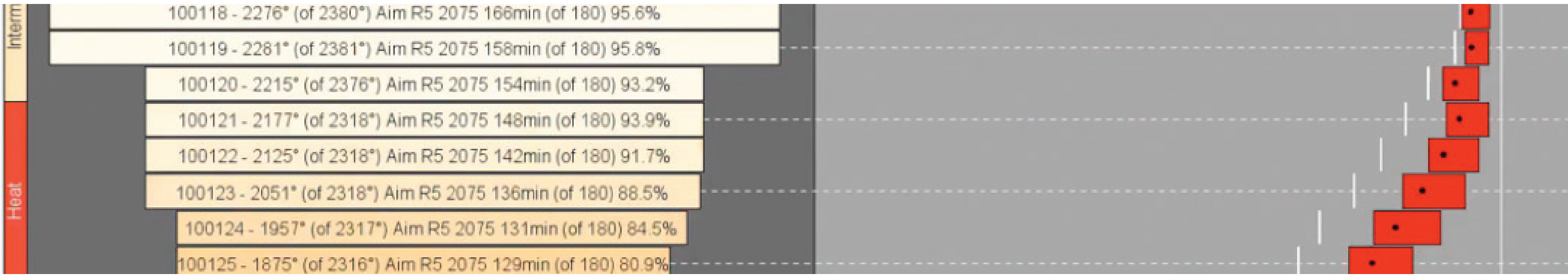

Temperature Setpoint Optimization in Steel Reheat Furnaces Using Open Architecture Neural Network and Modeling Software

Ricky Vesel

Griffin Open Systems
Phone: (440) 286-1319
Email: ricky@griffinopensystems.com

Justin Isaacs

Neundorfer Inc.
4590 Hamann Parkway, Ohio, USA, 44122
Phone: (440) 942-8990
Email: justini@neundorfer.com

GRIFFIN

NEUNDORFER

Keywords: reheat furnace, neural network, optimization, R5 temperature, efficiency, heat rate

Introduction

Steel reheat furnaces are widely used in the steel and metallurgical industries to heat metal slabs to the proper temperatures for rolling, forging, or extruding. While there are many different types and designs of reheat furnaces, they all face common challenges such as inefficient energy usage, low granularity or difficult to use control systems, and higher than desired reject rates due to unacceptable slab temperatures after extraction. Some of the difficulties in achieving the desired operating characteristics for reheat furnaces can be attributed but are not limited to:

- diverse product types often adjacent to each other in the furnace
- inconsistent burner performance and unstable flame location
- limited temperature feedback from inside the furnace
- process unpredictability such as delays for maintenance and other issues
- generally poor correlation between temperature setpoints and measured temperatures in the furnace
- furnace degradation leading to heat loss and difficulty meeting temperature setpoints

In order to overcome these difficulties and optimize reheat furnace operations, a new temperature setpoint control system was developed and tested in a four-month long study. The subjects of this study are three multi-zone walking beam type steel reheat furnaces like the one depicted below. Each furnace contains a preheat, heat, intermediate, and soak zone, where the temperature in each zone is controlled by a number of opposing burner pairs. Burner pairs are either controlled together or each side can be controlled individually, which provides some degree of lateral temperature control. Each burner pair is located on the top or bottom half of the furnace in order to provide heat distribution and some top-to-bottom temperature control.

Steel slabs ranging from 9" to 11" thick and weighing approximately 40,000 lbs are loaded from the charge side and traverse through the furnace over a period of time (approximately 3-4 hours) until they are extracted, at which point they enter the rolling mill. Each slab can have varying geometry (thickness, length, and width) and desired rolling temperatures depending on the product type. This variability further complicates the task of optimizing furnace temperatures for heating.

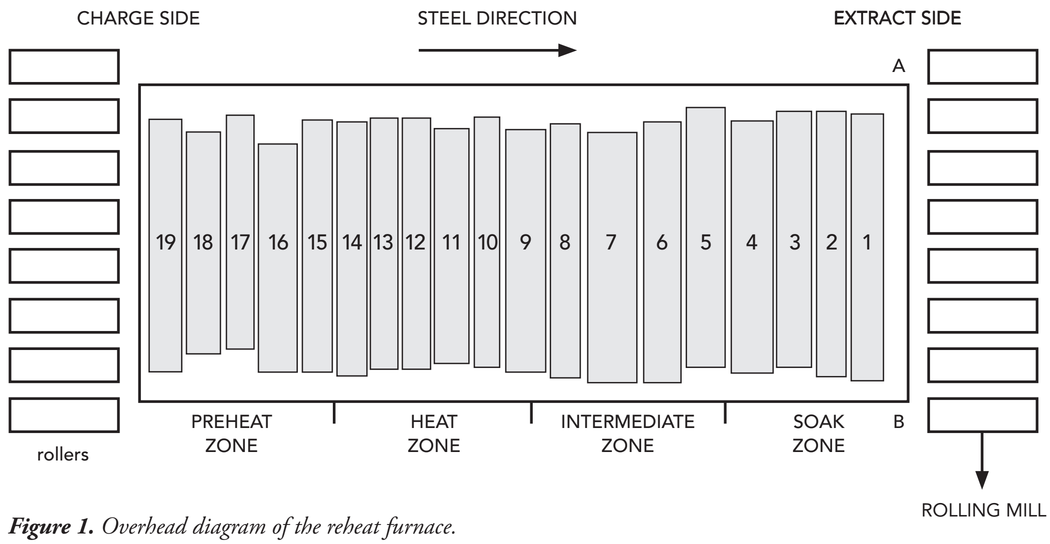

Figure 1. *Overhead diagram of the reheat furnace.*

There are multiple primary objectives for controlling temperatures in reheat furnaces. One objective is to ensure that the steel slabs are heated such that the measured steel temperature at a certain point in the rolling process, usually after the roughing stage (referred to as the "rougher exit temperature"), is within an appropriate window. If it is too hot, the steel may have experienced excessive slagging or surface melting. Excessive temperature also correlates directly with fuel waste. If temperatures are too low, increased rolling power is required and slabs may be rejected as the desired metallurgical properties are not obtained.

A second objective is to create an appropriate temperature distribution within the steel slab by the time it is extracted from the furnace, both through the slab's thickness and across the length of the slab. This ensures that the core of the slab is sufficiently heated, and a proper temperature difference exists between the top and bottom slab surfaces, in order to prevent the slab from curling up or down during rolling.

The third objective is to accomplish the prior two stated objectives with minimum energy input into the furnace, and when possible, minimize slab residence time in the furnace. There are additional objectives to furnace control, such as ensuring internal furnace temperatures do not exceed safe levels or shorten the life of the furnace and burners, in addition to the prevention of melting on the slab surface.

Discussion

The control system optimizer implemented in this study was built using commercially available software that runs on Windows-based PCs. The software is a graphical programming toolkit designed for rapid prototyping and testing of applications using a variety of built-in tools for control, modeling and optimization. It includes features such as data averaging and filtering, curve lookups, interpolation, database interfacing, and expert logic. It also includes interfaces for building neural networks and leveraging those networks for optimal control. The software's graphical programming environment allows deployment without the need for software development or writing any code. However, customization is also possible, and the construction and addition of new components to the graphical interface (coded in Java™) was used in this project to create a behind-the-scenes model of furnace state.

One of the goals of the software in an industrial environment is to augment and/or replace the traditional ad-hoc manual interventions used in process control with more consistent programmatic adjustments based on established best practices. In this case, engineers were able to create program logic to automatically and repeatably mimic the beneficial actions of furnace operators who heretofore made manual temperature adjustments based on temperature feedback they received from the rolling mill. Logic and models were also implemented to anticipate conditions and take actions earlier than if left in a manual mode of operation.

The customizability of the software package leads to the characterization of the software as "open architecture." Additionally, the user-friendly graphical programming interface makes it possible for plant engineers to view and modify the control logic in real-time and without interruption of control. This contrasts with many industrial process optimization solutions which function as a "black box."

Several features of the control optimization toolkit were utilized in the implementation of the final solution tested in this study, including physical modeling, neural networks, extensive use of simulation, customization, and the ability to easily incorporate a large amount of expert operator knowledge and best practices into the real-time control system logic.

Description of Solution

The primary goal of the control system is to achieve desirable rougher exit temperatures while minimizing fuel use and avoiding alarm conditions. The final solution includes three main components: a temperature control optimizer, a "pacing" feedback signal, and a graphical furnace map. The temperature control optimizer is the heart of the system and uses available information to determine the optimal temperature setpoints for each zone of the furnace at each time step. The pacing signal is a feature that has the ability to place the rolling mill in a Hold state, meaning that no slabs will be extracted from the furnace until the Hold is released. This prevents the rolling mill from extracting a slab that the system predicts will be at high risk for temperature rejection. The furnace map is the element of the control optimizer solution with the most visibility to furnace operators. It is a convenient visual representation of furnace state, including every slab in the furnace, slab temperatures, and their heating progress relative to the desired extract temperature. The individual components are described further below.

Temperature Control Optimizer

The temperature control optimizer has the overall goal of selecting the temperature setpoints for the various zones of the furnace to accomplish the following objectives: **1)** sufficient temperature of the slabs at extract time without overheating or surface melting; **2)** proper temperature distribution throughout the slabs (laterally and through the thickness); **3)** minimal fuel use; and **4)** avoiding temperature alarms or other undesirable furnace behaviors.

The logic within the temperature control optimizer consists of three main sections: physics-based slab temperature and enthalpy calculations, a neural network-based "temperature loss" model, and an MPC-like setpoint optimization loop based on the consultation of the heat and temperature loss models.

For each slab in the furnace, a 1D temperature model is maintained, which estimates the temperature throughout the slab using five nodes. The temperature model is based on radiative and convective heat transfer between the furnace and furnace gases and the top and bottom surfaces of the slab. The furnace is instrumented with an array of thermocouples, and these temperatures are interpolated in order to estimate the top and bottom temperatures experienced by the slab at various depths into the furnace. There is no modeling of the lateral temperature variation of the slab. Instead, lateral temperature variation is measured by temperature feedback from the rolling process and addressed with heuristic control adjustments.

When the slab is extracted from the furnace, information about the slab, including geometry, weight, metallurgy, and the calculated slab average temperature is fed into a neural network which then predicts the difference between this average temperature at extraction time and the rougher exit temperature measured later during the rolling process. This is the temperature loss model. The temperature loss model is used in the setpoint optimization loop, and is continually updated as new data is received.

Because this temperature loss model is rather simplistic (a more rigorous model would include data about the time-history of furnace state during the slab's residence), it is augmented with an error-correction algorithm. This algorithm compares the actual rougher exit temperature feedback to that predicted by the temperature loss model and calculates an error term which is updated every time new temperature feedback is received. This error term is then incorporated into the temperature loss model predictions during the setpoint optimization loop.

Setpoint Optimization Loop

The setpoint optimization loop is the process that determines, for each slab individually, what the temperature setpoints should be for each furnace zone in order to achieve the desired rougher exit temperature, given that particular slab's current temperature distribution and anticipated rate of progress through the furnace. These optimal setpoints are recalculated at regular intervals, hence the MPC-like nature of the system. When combined with accurate furnace temperature data, slab heat models, and temperature loss model, this process allows for optimal control of furnace temperature setpoints.

The setpoint optimization loop is essentially a binary search for optimal temperatures, where at each step the slab is simulated as it traverses through the furnace and experiences the selected temperatures. At the end of the slab's simulated residence, its rougher exit temperature is predicted via the temperature loss model including the current error term. If the predicted temperature is too high or low, the setpoints are lowered or raised, respectively. The search ends after a set number of iterations where the anticipated rougher exit temperature almost exactly matches the desired value. The final setpoints are stored as data corresponding to each slab. In addition, the expected profile of slab average temperature versus time for the slab is also stored. This temperature profile information informs the hold logic and furnace map (described below).

Rather than allowing each setpoint to vary individually, setpoints for all zones are selected as a group from a predefined table of possible zone temperature setpoints, covering low to high temperature operation. This table is based on engineering judgement and the recommendations of staff as to the expected temperature distribution in the furnace at various operating conditions. The setpoints are selected such as to produce an appropriate rate of heating, temperature saturation of the slab, and top to bottom temperature differential. This lookup table is made available to site engineers via tuning screens and can be modified at their discretion. Values from the table are interpolated according to the value of the binary search parameter. Use of the table ensures that a similar furnace temperature profile and thus heating strategy is employed for all slabs.

When optimal setpoints for each slab have been determined, the actual temperature setpoints in each zone are selected such that the slab with the highest setpoint requirement is satisfied. This is due to the emphasis on meeting rougher exit temperature, where too-low temperatures are less acceptable than slab overheating. A diagram of the setpoint optimization and selection process is shown below.

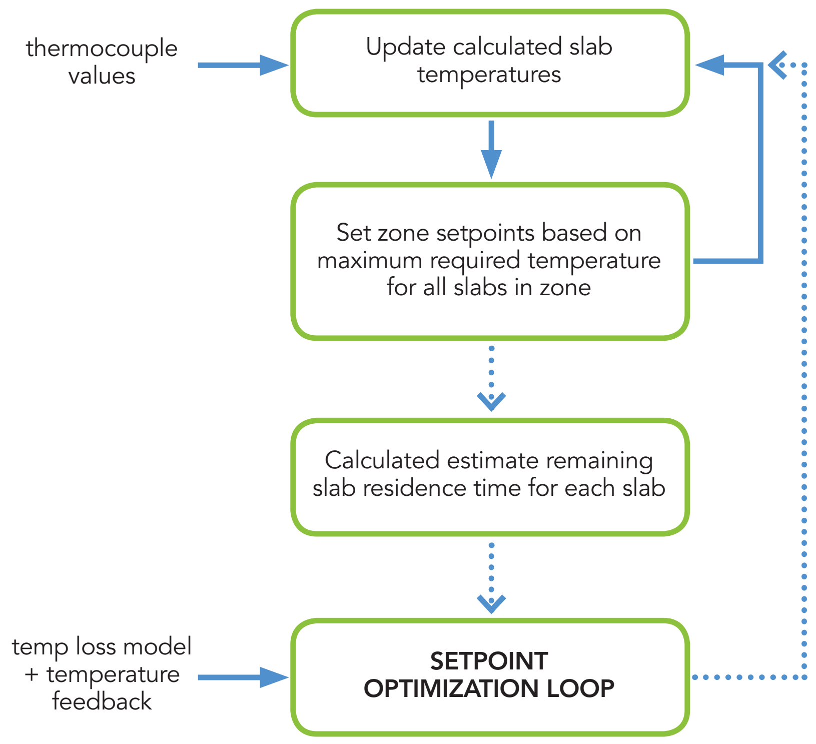

Figure 2. *Temperature setpoint control process. Solid lines are carried out each time step. Dashed lines are carried out on a larger time interval.*

Temperature Corrections

Temperature feedback often reveals a heating imbalance from left to right across the furnace. A subsystem of the temperature control optimizer attempts to correct this imbalance by increasing setpoints on the cool side. There are also critical thermocouples which have the ability to trip the unit if a certain temperature is exceeded. If these values reach a configurable warning level, the control system will gradually reduce the temperature setpoints in the relevant zones until the warning condition goes away.

Temperature Control During Holds

A substantial portion of the temperature control logic is dedicated to optimal fuel use during holds. Holds occur unpredictably and for a variety of reasons. Furnace operators have the ability to enter an expected hold duration, although this is not done for the majority of holds. During a hold, no slabs will be extracted from the furnace, and the slabs will remain stationary at their current location in the furnace. When a hold is released, the next slab is typically extracted immediately.

When a hold of unknown duration occurs, the temperature control system continues operating as if no hold had occurred, until each slab in the furnace has achieved the expected temperature given its current location and residence time in the furnace. When this condition has been achieved, the furnace enters a “waiting” state, where in the soak and intermediate zones, temperatures are selected to maintain slabs at their current temperature. In the heat and pre-heat zones, predetermined hold temperature setpoints are used.

When operators enter an expected hold duration, this gives the opportunity for deeper fuel savings, particularly during long holds of several hours or more. In this case, if the duration is long enough, temperature setpoints will begin reducing throughout the furnace, until they reach prescribed minimum values. This hold state may result in some cooling of the slabs. During such a hold, at every time step, a calculation is performed to estimate how much heating time would be required for the slabs in the soak zone to reach their extraction temperature. When the estimated remaining hold duration becomes less than or equal to this expected heating time (plus a user-defined safety margin), temperature setpoints are returned to normal control. This ensures that minimal time is wasted waiting for slabs to heat up when the rolling mill is again ready for operations.

Pacing Feedback

The control system has the ability to send its own hold signal to the rolling mill. This occurs when the temperature loss model predicts a rougher exit temperature for the next-to-extract slab that would put it at risk for temperature rejection. Under normal operation, this hold state persists until the slab achieves sufficient temperature for extraction. Prior to this feature, there were no programmatic mechanisms in place to prevent slabs from being extracted at cold temperatures. Furnace operators could use their judgement to place the mill on hold, but this would typically occur only after a first (or second) rejected slab had been lost.

Furnace Map

A custom furnace map display (shown below) was created to convey vital information about the state of the slabs in the furnace to furnace operators and facility personnel, a custom furnace map display was created, shown below. All graphics were created with the Java™ programming language. On the left, an overhead view of the furnace is provided, along with data for each slab including slab number, current temperature, aim extract temperature, current residence time, and percent heating progress. 100% heating progress means the average slab temperature calculated by the heat model is equal to the optimal extract temperature determined by the setpoint optimization loop. The slabs are also realistically colored according to their temperature.

To the right of the overhead view is the heating progress view. For each slab in the furnace, a rectangle is displayed, showing how close each slab is to its aim extract temperature. The black dots indicate the average slab temperature (which ideally reaches the 100% line right at the time of extraction), and the short white lines indicate the expected slab average temperature, given that slab's position and residence time in the furnace. These slabs are color coded, where red slabs are too cold to be extracted (and are high risk for temperature rejection if they are), yellow slabs are below aim temperature but are at low risk for temperature rejection, and green slabs are at or above aim extract temperature. If operators see red slabs about to be extracted, they know there is a risk of temperature rejection.

In the heating progress view, the slab width is determined by the minimum and maximum temperatures within the slab. The left edge of the rectangle corresponds to the lowest temperature (usually in the middle of the slab) as a percent of the aim extract temperature. The right edge corresponds to the highest temperature (usually on the top surface) as a percent of aim extract temperature.

On the right side of the furnace map, specific details about individual slabs are displayed, where clicking on a slab in either view populates the fields with information for that slab. Supplemental information about the state of the furnace and the last extracted slab are also displayed.

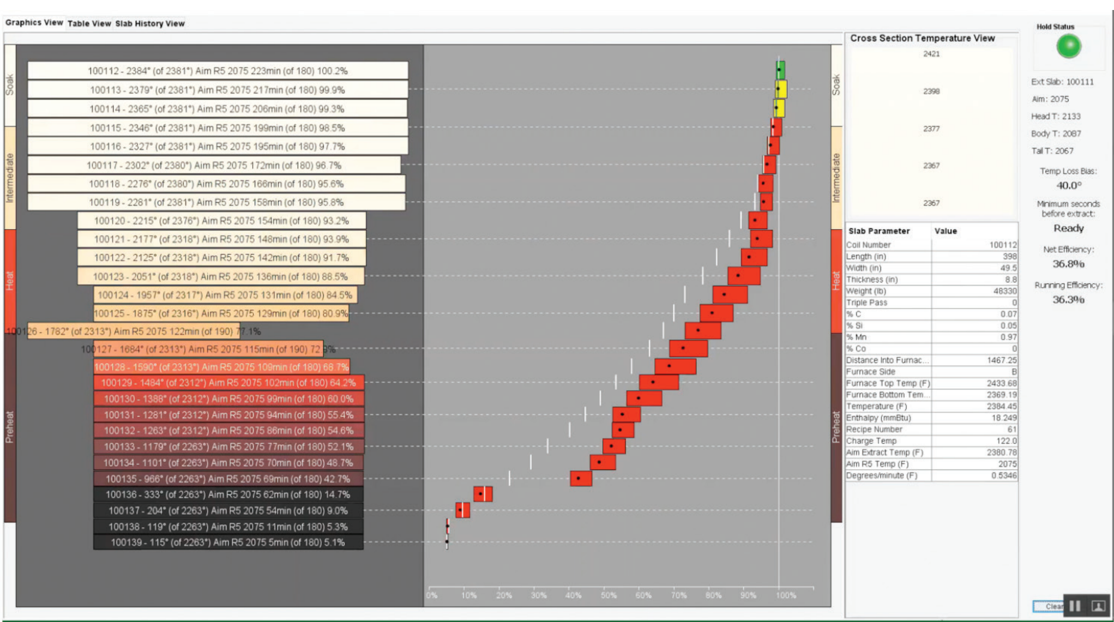

Figure 3. *Screenshot of the furnace map.*

Software

The software used to create the control system was summarized at the beginning of this section. Two features of the software are a graphical programming environment (pictured below) and the ability to augment the interface with custom components. A unique aspect of this control system is the in-memory furnace model containing the mathematical representation of the slab heat model as well as up-to-date information about each of the slabs in the furnace. Rather than attempting to build all of this using the graphical programming interface, it was decided that the furnace model would be a custom component. This also facilitated the creation of the custom graphical furnace map. A “Furnace” component was created, as well as several other custom components to represent events such as slab entering the furnace, slabs moving forward in the furnace, slab extraction, and the receipt of rougher exit temperature information. These custom components, when activated, send messages to the furnace component that allows it to update its internal state.

The ability of the software to allow real-time arbitrary modification of the control system without recompilation or interruption of control was critical in facilitating rapid iteration of the control system optimizer and the fast development of features in response to feedback from operators and other parties. Many features and logic changes were tested, tuned, and even created on the fly while the plant was conducting operations. It is expected that this would have been difficult or impossible with a non-graphical, non-real-time control optimization package.

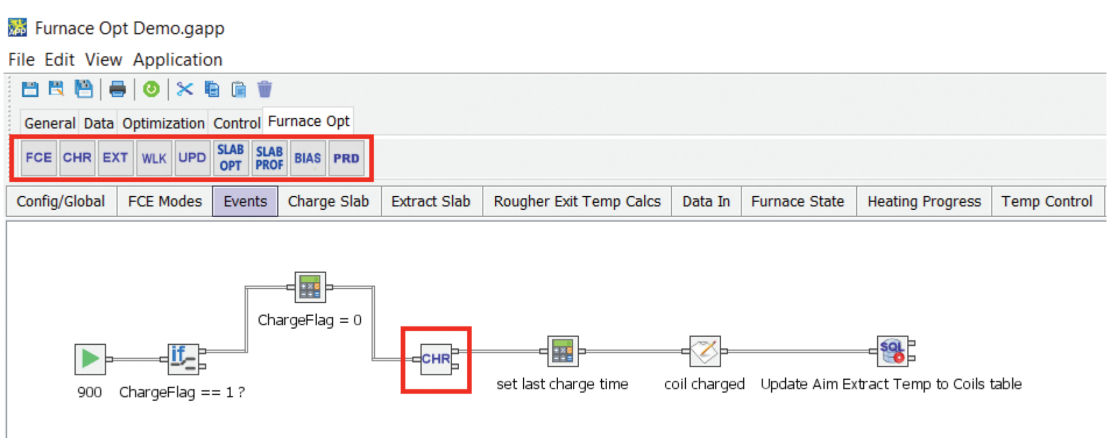

Figure 4. *Example screen capture from the graphical programming environment. The custom components are highlighted in red.*

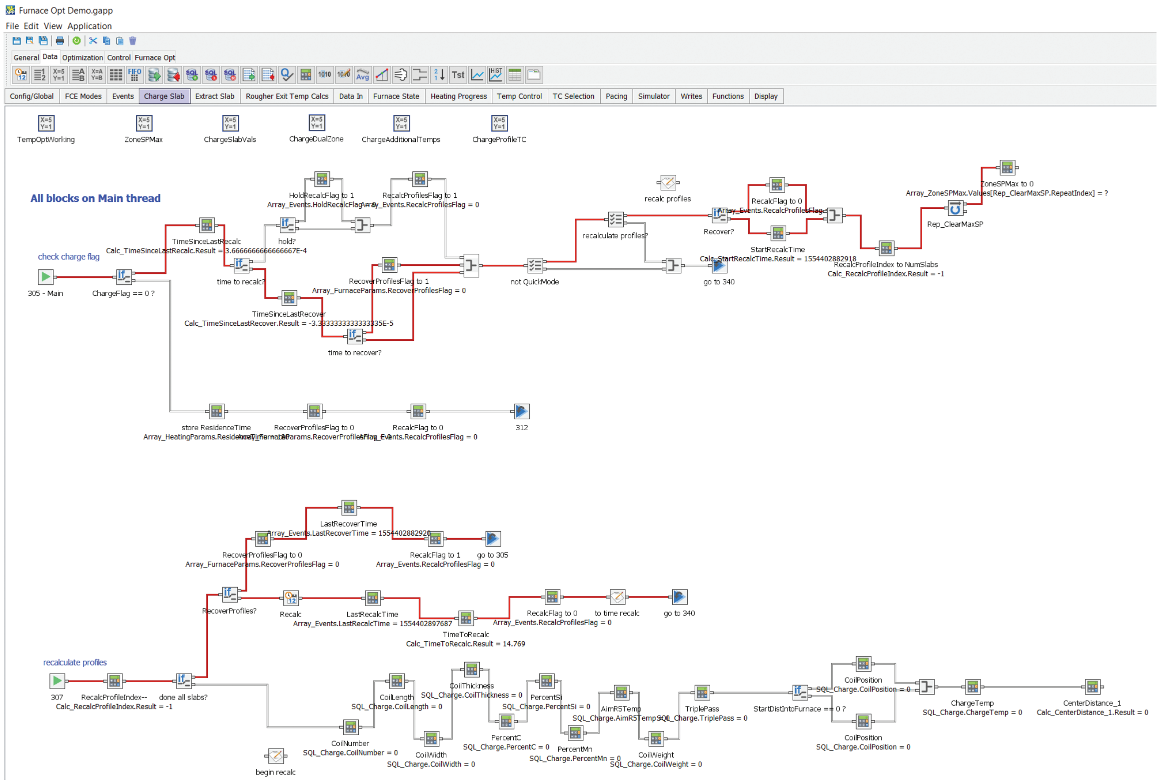

Figure 5. *Screen capture of the graphical programming environment showing more complex logic with execution path highlighting.*

Obstacles

The most significant obstacles throughout this study were poor or unpredictable furnace performance and operator intervention. The low NO_x burners installed in the furnace had exhibited erratic behavior, with visual inspections of the furnace interior during operation revealing a highly erratic flame pattern. The flame location in turn affected the temperature readings from the thermocouples, and quite often the temperature sensor associated with a given burner would have a poor correlation to the fuel flow at that burner.

Furnaces also exhibited significant temperature imbalances from one wing of the furnace to the other. These imbalances persisted despite many different mitigation strategies. The consequence was that slabs would have unequal heating across their length, and the hot end would have to be significantly overheated to ensure that the cold end was up to temperature, resulting in wasted fuel. It was discovered that following furnace maintenance the temperature imbalance problem was largely resolved.

Furnace degradation as a whole played a major role. Cost cutting measures resulted in decreased maintenance cycles and the furnaces showed worsening performance over time. This complicated control system development as the operating environment was unstable. At a certain point, one furnace had degraded to the point where it was unable to heat slabs to the extract temperature even given unlimited residence time. This also reflected poorly on the performance of the optimization system because the furnaces were wasting a large amount of fuel due to heat leakage.

Finally, one furnace was taken out of service and had extensive maintenance performed. When it returned to service, the control system demonstrated extraordinary efficiency. This confirmed the viability and effectiveness of the control system given a reasonable level of furnace maintenance. A graph demonstrating fuel use for this furnace before/after maintenance is shown below.

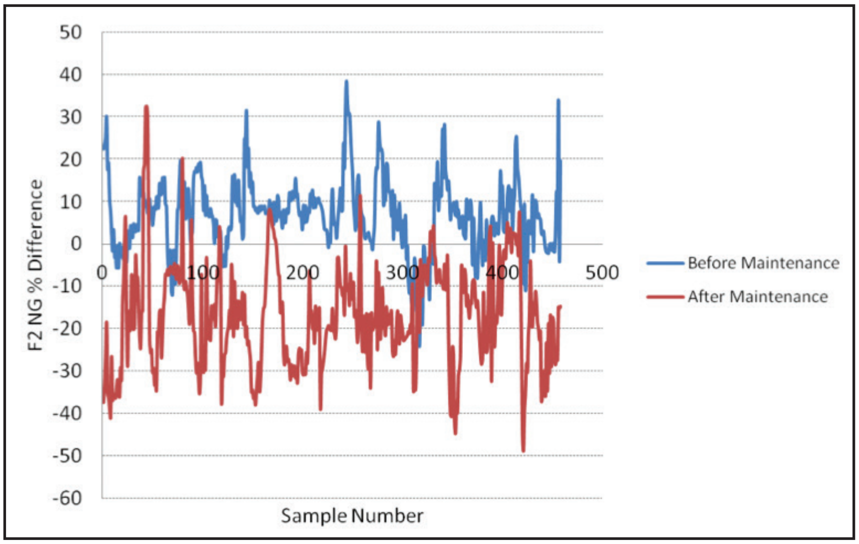

Figure 6. *Graph demonstrating reduced fuel use after maintenance. The x-axis represents each 4-minute interval. The y-axis represents the ratio of instantaneous fuel use of Furnace 2 compared to the average of Furnaces 1 and 3. Before maintenance, F2 was averaging around 12% more fuel use than the other two furnaces. After maintenance it averaged 9% less fuel use. All furnaces were using the new temperature control optimizer.*

The other major obstacle was operator intervention with the control system. A culture shift is necessary when implementing a complex automation system on top of a process that has historically been controlled with manual adjustments. Operators were accustomed to frequent manual tweaking of the system in response to the actual measured rougher exit temperatures. With the optimizer running, any operator adjustment to temperature setpoints required putting that point into “manual” mode, where it often remained until operators were reminded to return it to optimizer control. Operator interference also made it hard to judge the efficacy of the control system as it was difficult to separate operator decisions from those of the control optimization system. Over time, however, operators became less prone to intervention as the reliability of the system was established. The graphical furnace map played an important role in operator acceptance because it gave them confidence that slabs were heating as expected and that the control system was doing its job.

Particularly beneficial to operator acceptance was the elimination of reject slabs, resulting from a combination of better temperature optimization and implementation of the pacing feedback signal.

Results

The temperature control optimizer provided several benefits to the plant. The exact fuel use reduction proved difficult to quantify, as many factors contribute to fuel use, including furnace maintenance, total time under hold, and material type. As such, it was difficult to make a direct comparison to the production period preceding the implementation of the temperature control optimizer. However, fuel use reduction was ultimately estimated to be from 5-12%. After coming back from an outage and with furnaces in a good state of maintenance, under the new control system the plant demonstrated the lowest fuel use across the entire corporate fleet.

The actual fuel use results were compared to a 10% fuel use reduction criteria. Some descriptive plots showing heat rate in mmBTU/ton of steel are provided below. The results are divided into "total" and "running" heat rate, where running heat rate refers only to fuel used when the hold light is off, whereas total heat rate includes all fuel used during a given production run starting from when the first slab is extracted.

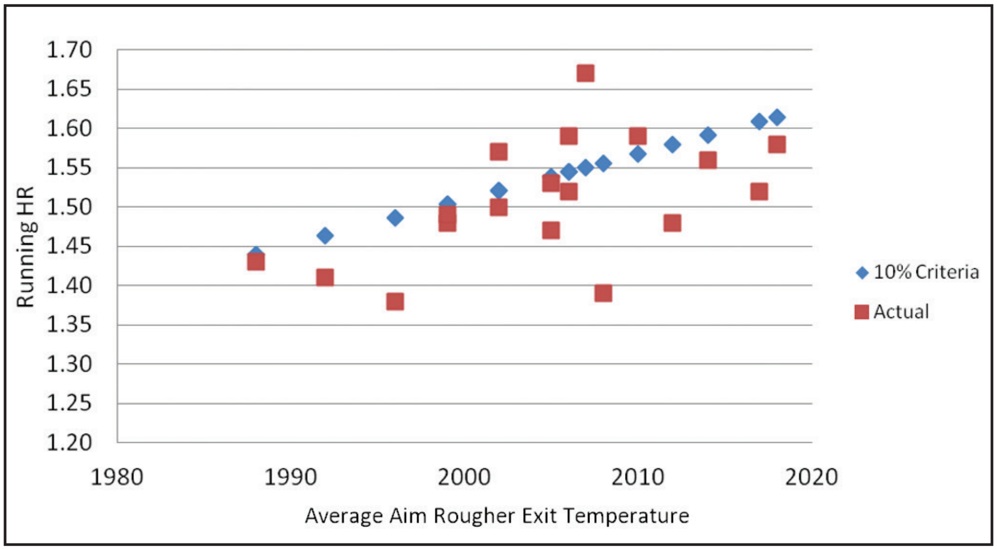

Figure 7. *Actual running heat rate (ignoring fuel used during holds) compared to 10% fuel use reduction criteria. Heat rate is mmBTU/ton of steel.*

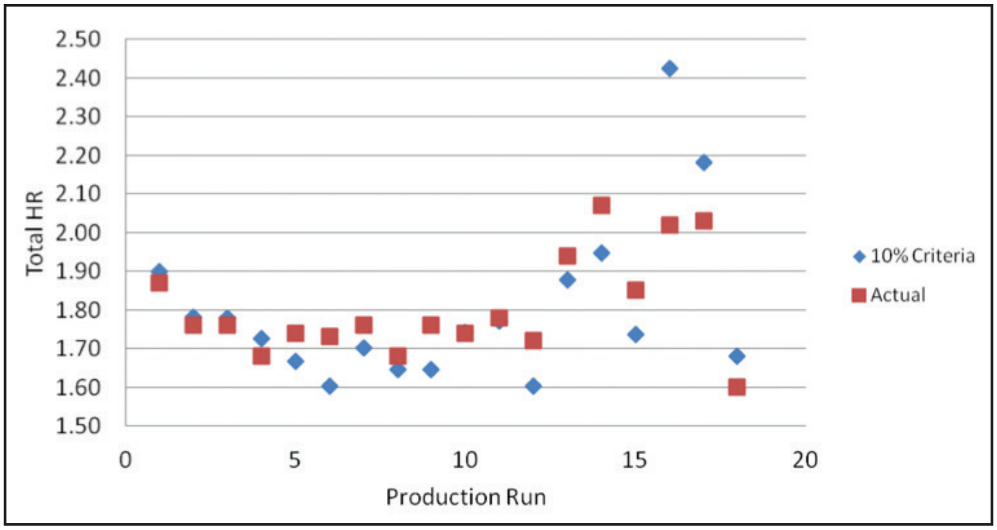

Figure 8. *Actual total heat rate compared to 10% reduction criteria*

Major secondary benefits were the elimination of temperature reject slabs which previously resulted in the scrapping of the entire piece, and the reduction of slag formation as a result of lower temperatures and lower excess air in the furnace. Slag reduction contributed to a roughly 1% increase in coil weight after the rolling process as slabs lost less material to slagging than previously, providing a direct contribution to plant revenue.

Ultimately, operations and engineering acceptance of the control system was very high, with the graphical furnace map becoming integral to operations. Both furnace operators and plant engineers rely on the furnace map to provide an accurate and detailed view of the heating progress of slabs inside the furnace.

Conclusion

A control system optimizer was created using an open-architecture graphical programming software toolkit with built-in neural network and optimization capabilities. Substantial process improvements were demonstrated including reduced waste and reduced fuel consumption.

Future Work

The most obvious improvements to the control system optimizer would address two principal areas of uncertainty in the temperature optimization process. The first is the temperature loss model. A more rigorous model of rougher exit temperature would include the time history of the slab in the furnace, including factors like thermocouple data, fuel use, and total furnace enthalpy. Fluctuations in the temperature loss model error term continue to be a source of process instability.

The second area of uncertainty is the calculation of remaining slab residence time. This is currently an estimation based on several factors, primarily the current furnace extract rate. However, the mill rolling system has internal calculations that decide precisely when slabs will be extracted. Access to these calculations or external duplication of them would enhance the remaining residence time calculation and improve stability of the optimized temperature setpoints. Alternatively, a model could be developed to estimate time-between-extractions based on the characteristics of the last-extracted slab.

Acknowledgement

The authors would like to acknowledge the support of Taber International and Griffin Open Systems for their participation in this study.

NEUNDORFER

4590 Hamann Parkway
Willoughby, OH 44094
(440) 942-8990
Fax: (440) 942-6824
www.neundorfer.com

© 2018 by AIST. Distributed by Neundorfer Inc. with permission of AIST. All other distribution is strictly prohibited.
Trademarks used herein are property of their respective owners.