

Real Time Simulator for the Runout Table of Hot Strip Mills

R. Krishna Kumar, S.K.Sinha
Department of Electrical Engineering
Indian Institute of Science
Bangalore - 560 012
INDIA

A.K.Lahiri
Department of Metallurgy
Indian Institute of Science
Bangalore - 560 012
INDIA

Abstract

In a hot strip mill, the strip is cooled by spraying water from top and bottom on the RunOut Table (ROT), before the strip is coiled. The desired mechanical properties and metallurgical structure of the strip is achieved by controlling the cooling rate and temperature of the strip on the ROT. This paper describes a novel real time simulator for ROTs. The simulator has been implemented in parallel on a multi-transputer system. The proposed simulator can interact with a controller in real time and can serve as a valuable design tool for evaluating the performance of controllers for ROTs. In addition, the simulator can be used as a tool for the design of new ROT layouts as well. **Keywords:** Hot strip mill, Runout table, Real time simulator, Parallel simulator, Interactive simulator.

1 Introduction

In a hot strip mill, the strip is cooled by spraying water from top and bottom on the runout table (ROT) before the strip is coiled. Figure 1 shows a typical runout table layout with four water zones. The ∇ symbol in figure 1 represents a temperature sensor. The strip enters the ROT at a temperature of about 900° C (typically) and is cooled down to about 650° C. The desired metallurgical and mechanical properties of the strip are achieved by controlling the rate of cooling and the final temperature of the strip. Cooling rate on the ROT is controlled by regulating the flow of water through the water valves. The velocity at which the strip traverses the ROT ranges from 3 m/s to 10 m/s for strips of thickness ranging from 1cm to 1mm.

In order to accurately control the cooling rate and final temperature of strip on the ROT, the water valves have

to be adjusted after the strip enters the ROT and its initial temperature is known (sensed). Further, as the strip moves on the ROT at the speeds mentioned above, the strip temperature measured at each temperature sensor would have to be compared with the expected temperature and any corrective action required for the subsequent water zones must take place before the strip reaches the next water zone. A fast online controller would, therefore, be necessary to ensure consistent quality of the strip on the ROT.

Development of such controllers is a topic of current interest [1,2,3,4,5,6]. However, no appropriate tools are available to evaluate the performance of controllers for ROTs and at present controllers for ROTs can be evaluated only by trials on the actual plant. On site trials are expensive and not many steel mills are willing to offer their plants for such experimentation.

In this paper, a *real time* simulator for the runout table of hot strip mills is proposed which can be used to evaluate controllers meant for ROTs. The proposed simulator sends strip temperature information to the controller online and responds to the control signals from the controller in real time. A visual front end has also been developed which provides a pictorial view of the ROT and shows the changes that take place in strip temperature and water valve settings on the ROT during a simulation run. One run of the simulator corresponds to the movement of one strip on the ROT. The simulator has been written in Occam and implemented in parallel on a network of transputers.

To the best of the knowledge of the authors no *real time* simulator for ROTs is reported in literature. The simulators for ROTs reported in literature [6,7,8] can predict the temperature of the strip given the water flow rates apriori and have no provision for communication of data and control signals from the sensors and controllers. Con-

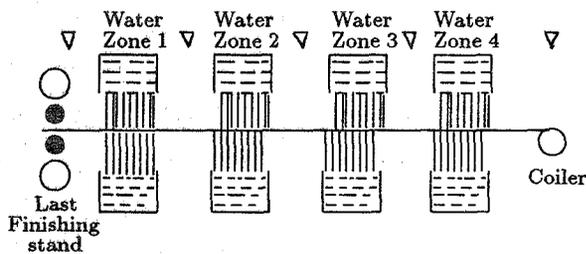


Figure 1: Layout of a ROT

sequently, such simulators cannot be used to evaluate the controllers.

Real time simulation of the cooling process on the ROT requires modelling of the cooling process. A mathematical model of the cooling process and parallel solutions of the model equation have been developed and are reported in a companion paper titled, "Modelling of the Cooling Process on the Runout table of a Hot Strip Mill - A Parallel Approach" [9].

2 The Proposed Simulator

In the proposed simulator, the ROT to be simulated is divided into as many simulation regions as the number of water zones on the ROT. Each simulation region contains a water zone and the following temperature sensor, if any, before the next water zone. In case no temperature sensor exists between two water zones, the simulation region ends at a pre-defined point between the two water zones. For example, the ROT of one of the SAIL plants, which has four water zones and three temperature sensors, is divided into four simulation regions as shown in figure 2.

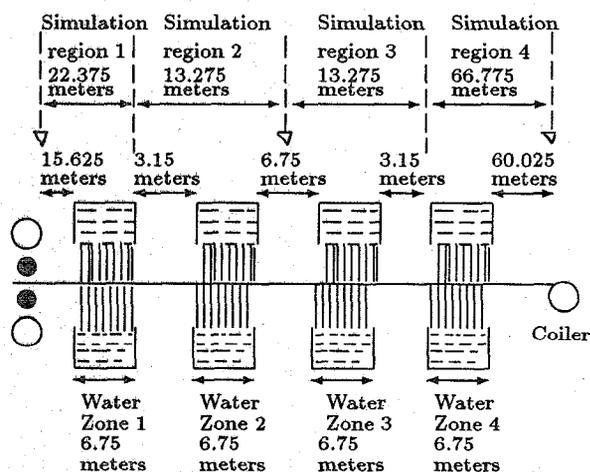


Figure 2: A SAIL Plant ROT and its Division into Simulation Regions

2.1 Functional Description of the Simulator

The functional modules of the proposed simulator are shown in figure 3. Information regarding the plant layout, physical properties and dimensions of the strip and the velocity at which the strip moves on the ROT are entered into the simulation system through the user interface prior to a simulation run.

A number of Real Time Temperature Predictor (RTTP) modules, one for each simulation region of the ROT, constitute the core of the proposed simulator. Each RTTP module i ($i \neq 1$) receives information about the temperature of the strip from the preceding RTTP module ($i - 1$) when the strip enters the simulation region i . The RTTP module 1 receives the temperature of the strip from the user interface when the strip enters the ROT. The initial temperature module is programmed at the beginning of a simulation run through the user interface. In addition to the temperature information, the RTTP modules receive control information regarding the water valve settings. Initially, the water valves are set as defined by the user but the valve settings may get modified during the simulation run under controller action. The communication modules serve as the interface between the controller and the RTTP modules.

Each RTTP module i solves the model equation in parallel [9] to compute the new expected temperature of the strip at the exit point of the simulation region i (which is the entry point of the simulation region $i+1$) as soon as the module senses any change in the input temperature of the strip and/or the water valve setting. This computed temperature is sent to RTTP module $i+1$ when the strip reaches the start of simulation region $i+1$. RTTP module i sends the temperature of the strip to the controller through the communication module as well as to the user interface when the strip reaches the end of simulation region i . However, the temperature information is sent to the controller only if a temperature sensor is present at the end of simulation region i .

For ease of comprehension, a visual front end has been provided which presents a pictorial view of the simulated cooling process to the metallurgist. The strip movement on the ROT, changes in the water flow and temperature of the strip at each temperature sensor point are displayed on the screen.

2.2 Hardware for the implementation of the proposed simulator

The proposed simulator has been implemented on a network of transputers. The IMS B008 board [10] with T800 transputer modules has been employed for the purpose. For simulation regions of upto 60 meters in length and strip velocities of upto 10 m/s two transputers are able to carry out the required computations in real time.

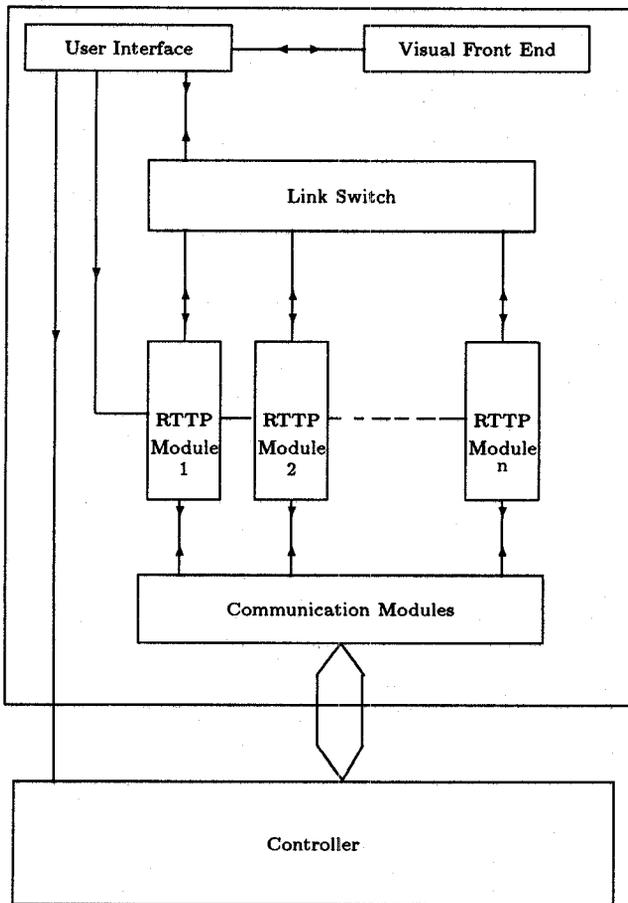


Figure 3: Functional Modules of the Proposed Simulator

Hence each RTTP module in figure 3 (and the corresponding communication module) has been implemented on a pair of transputers. However, provision has been made for deployment of upto four transputers for implementing a RTTP module and the corresponding communication module if the length of a simulation region is more than 60 meters or if the strip velocities are higher, requiring faster computation. The four-transputer implementation of a RTTP module i and the associated communication module is shown in figure 4.

The complete hardware arrangement for the simulation of the example ROT of figure 2 is shown in figure 5. A pair of transputers P_{i1}, P_{i2} ($1 \leq i \leq 4$) implement one RTTP and communication module. Communications from processors P_{11}, P_{21}, P_{31} and P_{41} to the user interface take place through the IMS C004 link switch [11] and a transputer P_{com} . P_{com} polls the RTTP modules in a round robin fashion and on receipt of any message from the polled transputer, P_{com} passes the message to the user interface.

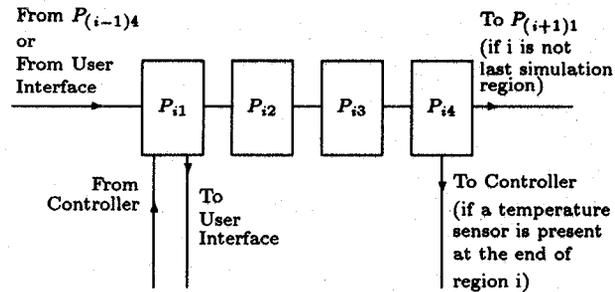


Figure 4: Four Processor Implementation of RTTP Module i and Associated Communication Module

3 Simulator Software

The simulator software is explained with respect to figure 5. As explained earlier, information regarding the plant layout, physical properties and dimensions of the strip and the velocity at which the strip moves on the ROT are entered into the simulation system through the user interface prior to a simulation run. An *user_process* executes on the user interface and downloads the above parameters of the ROT and strip, and initiates the simulation run. During simulation, information regarding the temperature of strip and water flow rates are communicated to the *user_process* by the RTTPs through P_{com} .

Each RTTP and communication module i ($1 \leq i \leq 4$) has four parallel processes - *solution_process1*, *solution_process2*, *output_process* and *communication_process*. *Solution_process1* and *solution_process2* run on P_{i1} and P_{i2} (fig. 5), respectively, at low priority and compute in parallel the expected surface temperature of the strip at the end of region i by solving the model equation with the temperature of the strip and water flow rate prevailing in region i . The temperature of the strip at the entry point of region i ($i \neq 1$) is communicated to the RTTP module i by the RTTP module $(i - 1)$. RTTP module 1 receives the initial temperature of the strip from the user interface. Information regarding valve settings are communicated to the RTTP modules by the controller. The solution processes are invoked only when the temperature of the strip at the beginning of a simulation region changes by more than the user defined value (1°C in the present implementation) and/or a control signal for a change in water flow rate by more than a user defined value ($0.0001 \text{ m}^3/\text{s}$ in the present implementation) is received.

The *output_process* and the *communication_process* execute on P_{i1} and P_{i2} , respectively, at high priority. The *output_process* maintains four queues; *valueq1*, *valueq2*, *timeq* and *delayq*. *valueq1* stores the temperature values of the strip at the end of region i and *valueq2* stores the water flow rates in region i , received from the solution processes as the strip moves on the ROT. The time instants when a new temperature value of the strip at the start of region i or when a control signal for a change in the valve setting for region i is received by the solution processes are

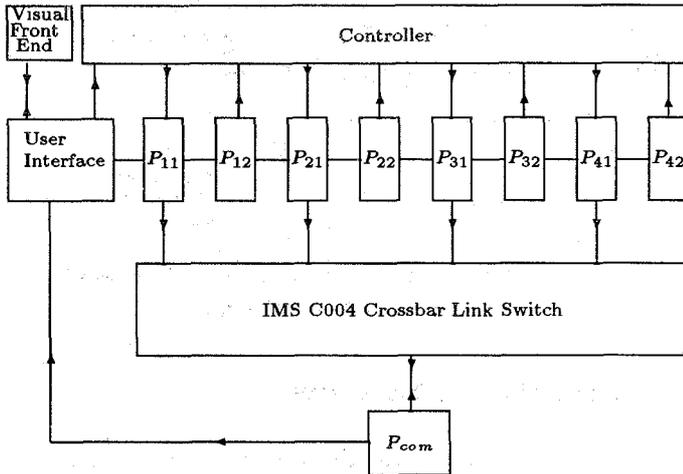


Figure 5: Simulator Hardware for the ROT in figure 2

recorded in the *timeq*. If a new temperature at the start of region *i* is received, a value equal to the time taken by a point in the strip to traverse the entire length of region *i* is stored in the *delayq*. On the other hand, if a control signal for a change in the water flow rate in region *i* is received, a value equal to the valve reaction time plus the time taken by a point in the strip to move from the water zone in region *i* to the end of region *i* is stored in the *delayq*.

The *communication_process* has only three queues; *valueq1*, *timeq* and *delayq*. Information is stored in these queues in a manner similar to that explained for the *output_process*.

The *output_process* communicates the expected temperature of the strip at the end of region *i* stored in *valueq1* to the *user_process*. The same information is sent to the controller by the *communication_process*. However, this information is communicated to the controller only if a temperature sensor is present at the end of region *i* in the plant being simulated. The *communication_process* also sends the temperature information of the strip stored in *valueq1* to RTTP module *i+1* provided *i* is not the last region. The *output_process*, in addition, communicates information regarding changes in the water flow rates stored in *valueq2* to the *user_process*. The transfer of information from *valueq1* and *valueq2* to the *user_process*, controller and RTTP module (*i+1*) take place in real time and hence the *output_process* and *communication_process* are executed at high priority.

The manner in which the software responds to the changes in the temperature of the strip and the water flow rates during a simulation run is explained below with an example sequence of events occurring in a region *i*.

Figure 6 shows a portion B of the strip entering the simulation region *i* with a temperature $T_i = T_1$ at time $t = t_0$ with the water flow in the region $w_i = w_1$. All the subsequent time instants in the example are measured

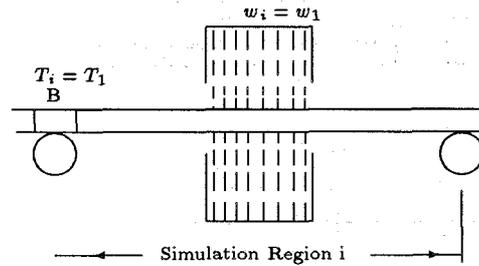


Figure 6: Conditions in Simulation Region *i* at $t = t_0$

from $t = t_0$.

After t_0 , as the strip continues to move, let the following events take place in sequence:

1. A control signal for a change in the water flow w_i to w_2 is received at time instant t_1 .
2. At time instant t_2 , a portion C of the strip enters the region *i* with a temperature $T_i = T_2$ as shown in figure 7. The location of portion B of the strip at time instant t_2 would depend on the speed at which the strip moves and the ROT layout.
3. At time instant t_3 , a control signal for a change in the water flow to w_3 is received.

In the above example, the solution processes would be invoked at t_0 and the expected temperature of portion B of the strip when it reaches the end of region *i* (T_{ie}) would be computed with $T_i = T_1$ and $w_i = w_1$. The computed value of temperature T_{ie} is sent to the *output_process* and *communication_process* and is appended to the *valueq1* in both the processes. The value of w_1 is sent to the *output_process* and is appended to the *valueq2*.

At t_1 , when a control signal to change w_i to w_2 is received, the expected temperature T_{ie} of the strip is computed by the solution processes with $T_i = T_1$ and $w_i = w_2$. As before, the computed value of T_{ie} is sent to the *output_process* and *communication_process* and appended to the *valueq1* in both the processes. The value of w_2 is sent to the *output_process* and appended to the *valueq2*.

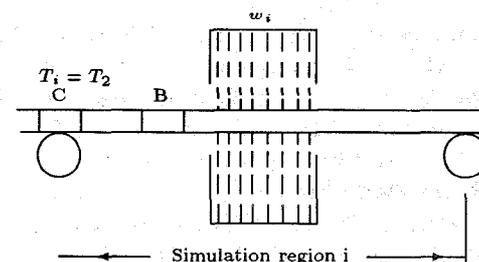


Figure 7: Conditions in Simulation Region *i* at $t = t_2$

The steps executed by the software at time instants t_2 and t_3 shall depend on the relative values of the quantities defined below:

$t_a = (t_2 + \text{time taken by a point in the strip to reach the water zone from start of region i})$

= The instant of time at which the portion C of the strip with $(T_i = T_2)$ reaches the water zone

$t_b = (t_1 + \text{valve reaction time})$

= The time at which the valve setting $w_i = w_2$ is effective

$t_c = (t_3 + \text{valve reaction time})$

= The time at which the valve setting $w_i = w_3$ becomes effective

At t_2 , t_a may be greater than t_b , equal to t_b or less than t_b . Similarly at t_3 , t_a may be less than t_c , equal to t_c or greater than t_c . The steps executed by the software for each of the above six situations are explained below:

$t_a > t_b$

- Compute T_{ie} with $T_i = T_2$ and $w_i = w_2$.
- Append T_{ie} to *valueq1* in *communication_process*.
- Append T_{ie} to *valueq1* in *output_process*.
- Append w_2 to *valueq2* in *output_process*.

$t_a = t_b$

- Compute T_{ie} with $T_i = T_2$ and $w_i = w_2$.
- Overwrite the last entry in *valueq1* of *communication_process* with T_{ie} ¹.
- Overwrite the last entry in *valueq1* of *output_process* with T_{ie} ¹.
- Delete the last entry in *timeq*.
- Delete the last entry in *delayq*.

$t_a < t_b$

- Compute T_{ie} with $T_i = T_2$ and $w_i = w_1$.
- Overwrite the last entry in *valueq1* of *communication_process* with T_{ie} ².
- Overwrite the last entry in *valueq1* of *output_process* with T_{ie} ².

¹The entries being overwritten would correspond to T_{ie} computed with $T_i = T_1$ and $w_i = w_2$

²The entries being overwritten would correspond to T_{ie} computed with $T_i = T_1$ and $w_i = w_2$

- Overwrite the last entry in *valueq2* of *output_process* with w_1 ³.

- Recompute T_{ie} with $T_i = T_2$ and $w_i = w_2$.

- Append the new T_{ie} to *valueq1* of *communication_process*.

- Append the new T_{ie} to *valueq1* of *output_process*.

- Append w_2 to *valueq2* of *output_process*.

- Interchange the last two entries in the *timeq*.

- Interchange the last two entries in the *delayq*.

$t_a < t_c$

- Compute T_{ie} with $T_i = T_2$ and $w_i = w_3$.

- Append T_{ie} to *valueq1* in *communication_process*.

- Append T_{ie} to *valueq1* in *output_process*.

- Append w_3 to *valueq2* in *output_process*.

$t_a = t_c$

- Compute T_{ie} with $T_i = T_2$ and $w_i = w_3$.

- Overwrite the last entry in *valueq1* of *communication_process* with T_{ie} ⁴.

- Overwrite the last entry in *valueq1* of *output_process* with T_{ie} ⁴.

- Overwrite the last entry in *valueq2* of *output_process* with w_3 ⁵.

- Delete the last entry in *timeq*.

- Delete the last entry in *delayq*.

$t_a > t_c$

- Compute T_{ie} with $T_i = T_1$ and $w_i = w_3$.

- Overwrite the last entry in *valueq1* of *communication_process* with T_{ie} ⁶.

- Overwrite the last entry in *valueq1* of *output_process* with T_{ie} ⁶.

- Overwrite the last entry in *valueq2* of *output_process* with w_3 ⁷.

- Recompute T_{ie} with $T_i = T_2$ and $w_i = w_3$.

³The entry being overwritten with $w_i = w_1$ is $w_i = w_2$

⁴The entries being overwritten would correspond to T_{ie} computed with $T_i = T_2$ and $w_i = w_2$

⁵The entry being overwritten with $w_i = w_3$ is $w_i = w_2$

⁶The entries being overwritten would correspond to T_{ie} computed with $T_i = T_2$ and $w_i = w_2$

⁷The entry being overwritten with $w_i = w_3$ is $w_i = w_2$

- Append the new T_{ie} to *valueq1* of *communication_process*.
- Append the new T_{ie} to *valueq1* of *output_process*.
- Interchange the last two entries in the *timeq*.
- Interchange the last two entries in the *delayq*.

As can be seen from the above example, the simulator needs to solve the model equation repeatedly to override the previously computed values when $t_a = t_b$, $t_a < t_b$, $t_a = t_c$ and $t_a > t_c$. The strips move at speeds ranging from 3 m/s to 10 m/s depending on strip thickness, and the length of a simulation region may vary from 20 to 60 meters. The model equation may have to be solved several times for the entire length of the simulation region within the time a point in the strip traverses the simulation region. Although the cooling process for one strip takes several seconds (in some cases several minutes) for accurate real time simulation the model equation may have to be solved within milliseconds depending on the length of the simulation region, speed of the strip and the instants of time at which the control signals are received. The proposed parallel implementation meets the above time constraint even for the fastest cooling process presently in vogue.

4 Implementation and Results

The simulator has been implemented on the hardware arrangement shown in figure 5 and tested for the ROT layout shown in figure 2.

Trial runs of the proposed simulator have been carried out with and without control action for strips of thickness ranging from 1mm to 1cm, moving at velocities ranging from 10 m/s to 3 m/s. The maximum valve reaction time of the water valves has been taken as 1 second.

In order to test the real time operation of the proposed simulator the temperature of the strip as it entered the ROT (T_0) was deliberately made different at different time instants. For a trial run with a 7.7 mm thick strip moving at 8 m/s the instants of time at which the simulator effected the corresponding changes in the temperature of the strip at the end of each of the four simulation regions (T_i) are given in table 1.

Figures in parentheses are the surface temperatures of the strip as given out by the simulator. A comparison of these time instants with the time it takes for a portion of the strip to travel from the start of the ROT to the end of the four simulation regions confirms that the simulation is being carried out in real time. During the above trial run the controller was disabled and the water valves were set for average water flows of $3.5517 \text{ m}^3/\text{s}$ in water bank 1, $3.153 \text{ m}^3/\text{s}$ in water bank 2, $2.5108 \text{ m}^3/\text{s}$ in water bank 3 and $2.0749 \text{ m}^3/\text{s}$ in water bank 4.

The ability of the simulator to interact with an external controller has been tested by running the simulator in conjunction with a controller for ROTs developed by the present authors. The controller received temperature information of the strip at the temperature sensor points on the ROT from the simulator and in turn sent control signals to the simulator to regulate the water flow such that the temperature of the strip at the end of each simulation region is maintained within the specified range. The surface temperatures of the strip at the end of region 1 and the water flow rates in region 1, both with and without a controller, are presented in table 2. The figures presented in table 2 pertain to the same strip for which simulation results are presented in table 1.

The desired temperature of the strip at the end of region 1 was set at the nominal value of 753.76° C . As can be seen from table 2, the simulator responds to control signals by adjusting the water flow and returns the correct (desired) temperature of the strip when run in conjunction with the controller. On the other hand, in the uncontrolled mode the simulator does not alter the water flow rates set prior to the simulation run and consequently returns temperatures which depend on the temperature of the strip at the entry point of the ROT.

5 Conclusions

A real time parallel simulator for the runout table of hot strip mills has been proposed and implemented on a multi-transputer system. The implementation details and software of the proposed simulator have been discussed.

The simulator has been tested successfully for its real time performance with temperature changes occurring at 1 second time intervals as the strip moves on the ROT. The simulator has also been tried in conjunction with a controller and found to respond to the commands from the controller in real time. The proposed simulator can therefore be used to evaluate the performance of controllers of ROTs in real time.

Table 1: Time Instants at which T_0 is Changed and the Corresponding changes in T_i are Effected

At the start of ROT	At the end of region 1*	At the end of region 2*	At the end of region 3*	At the end of region 4*
0 secs ($T_0 = 878^\circ \text{C}$)	2.797 secs (753.76°C)	4.456 secs (763.13°C)	6.116 secs (724.5°C)	14.463 secs (693.9°C)
1 secs ($T_0 = 908^\circ \text{C}$)	3.797 secs (759.06°C)	5.456 secs (768.75°C)	7.116 secs (730.56°C)	15.463 secs (696.35°C)
2.5 secs ($T_0 = 888^\circ \text{C}$)	5.297 secs (755.84°C)	6.956 secs (765.34°C)	8.616 secs (728.98°C)	16.963 secs (694.32°C)
4 secs ($T_0 = 868^\circ \text{C}$)	6.797 secs (751.37°C)	8.456 secs (760.6°C)	10.116 secs (723.09°C)	18.463 secs (692.6°C)
6 secs ($T_0 = 858^\circ \text{C}$)	8.797 secs (748.69°C)	10.456 secs (757.75°C)	12.116 secs (718.99°C)	20.463 secs (689.74°C)
10 secs ($T_0 = 848^\circ \text{C}$)	12.797 secs (745.71°C)	14.456 secs (754.59°C)	16.116 secs (716.64°C)	24.463 secs (687.23°C)

Note * For the strip speed of 8 m/s and given lengths of the simulation regions (fig 2) a change in the temperature of the strip at the start of ROT at t, should be reflected at (t+2.797) secs at the end of region 1, (t+4.456) secs at the end of region 2, (t+6.116) secs at the end of region 3 and (t+14.463) secs at the end of region 4 for real time simulation.

Table 2: Simulator Output for region 1 with and without Controller

Start of ROT	Time in seconds →	0.0	1.0	2.5	4.0	6.0	10.0
	Temperature in $^\circ \text{C}$ →		878.0	908.0	888.0	868.0	858.0
Region 1	Time in seconds →	2.797	3.797	5.297	6.797	8.797	12.797
	Water flow rate without controller → in m^3/s	3.5517	3.5517	3.5517	3.5517	3.5517	3.5517
	Water flow rate with controller → in m^3/s	3.5517	3.7957	3.647	3.4435	3.3231	3.1913
	Temperature in $^\circ \text{C}$ without → controller	753.76	759.06	755.84	751.37	748.69	745.71
	Temperature in $^\circ \text{C}$ with → controller	753.76	753.76	753.76	753.76	753.76	753.76

References

- [1] G.V. Ditzhuijzen. The controlled cooling of hot rolled strip: a combination of physical modelling, control problems and practical adaptation. *IEEE Trans. on Automatic Control*, Vol. 38(No. 7):pp 1060-1065, July 1993.
- [2] G. Uetz, G. Woelk, and T. Bischops. Influencing the formation of the steel structure by suitable temperature control in the runout sections of hot strip mills. *Steel Research*, Vol. 62(No. 5):pp 216-222, May 1991.
- [3] K. Yaniro, J. Yamasaki, M. Furukawa, et al. *Development of Coiling Temperature Control System on Hot Strip Mill*. Technical Report No. 24, Kawasaki Steel, Apr 1991.
- [4] A.G. Groch, R. Gubernat, and E.R. Birstein. Automatic control of laminar flow cooling in continuous and reversing hot strip mills. *Iron and Steel Engineer*, Vol. 67(No. 9):pp 16-20, Sep 1990.
- [5] M.D. Leltholf and J.R. Dahm. Model reference control of runout table cooling at LTV. *Iron and Steel Engineer*, Vol. 66(No. 8):pp 31-35, Aug 1989.
- [6] E.N. Hinrichsen and G.P. Petrus. Hot strip mill runout table cooling - a system view of control, operation and equipment. *Iron and Steel Engineer*, Vol. 53(No. 10):pp 29-34, Oct 1976.
- [7] J. Filipovic, R. Viskanta, F.P. Incropera, and T.A. Veslocki. Thermal behaviour of a moving steel strip cooled by an array of planar water jets. *Steel Research*, Vol. 63(No. 10):pp 438-446, Oct 1992.
- [8] K. Yanagi. Prediction of strip temperature for hot strip mills. *Transactions of the Iron and Steel Institute of Japan*, Vol. 16(No. 1):pp 11-19, Jan 1976.
- [9] Krishna Kumar R., S.K. Sinha, and A.K. Lahiri. Modelling of the cooling process on the runout table of a hotstrip mill - A parallel approach. Companion paper.
- [10] *IMS B008 User Guide and Reference Manual*. INMOS, 1988.
- [11] *The Transputer Databook*. INMOS, 1991.