BUILDING A FeNi SMELTER SIMULATOR

H. Oterdoom[1], R. Degel[1]

[1]SMS SIEMAG AG, Eduard Schloemann-Strasse 5, 40237 Duesseldorf, Germany,
Harmen.Oterdoom@sms-siemag.com

ABSTRACT

At the Cu2007 SMS Siemag presented a copper slag cleaning simulator, which had been developed by Professor A. Warczok of the University of Chile. The now presented simulator is for a 6-in-line FeNi smelter, similar to the ones now under construction by SMS Siemag in Brazil for Vale Inco and Anglo American Brasil.

This simulator has been developed with the target to control the FeNi furnace, in particular to maintain operation smooth and the energy and mass balances in equilibrium within a realistic range of operating parameters:

Furnace should be running smoothly
Electrical energy input should be balanced with smelting calcine and heat losses (energy input=energy output=material input)
Charging calcine should be balanced with smelting calcine (material input=production rate)
Tapping should be balanced with smelting calcine (production rate=material output)

In practice interruptions can happen unexpectedly, which leaves the operator to decide if the furnace load should be reduced or not, often without knowing how long the unplanned stop may last.

The simulator contains linked models for the charging, electrical power input and tapping combined into a mass and energy balance. A fast forward option makes it possible to focus on balancing these items, without having to wait for long time spans.

Building a simulator forces developers to describe the actual furnace operation in a mathematical and logical model. Some lessons learned on how to start such a project are described at the end that may be helpful for others aiming at developing a similar kind of model.

1 HISTORY OF THE SIMULATOR

At the Cu2007 a paper was presented regarding a copper slag cleaning simulator, developed by Professor A. Warczok [1]. Targets of this simulator were:

- Training of operators
- Simulating slag cleaning behaviour of the furnace
- Understanding operational fundamentals

Based on this development it was decided to look at other applications (as mentioned at the time) and, due to the FeNi furnaces under construction, developing a six in line FeNi simulator was chosen.

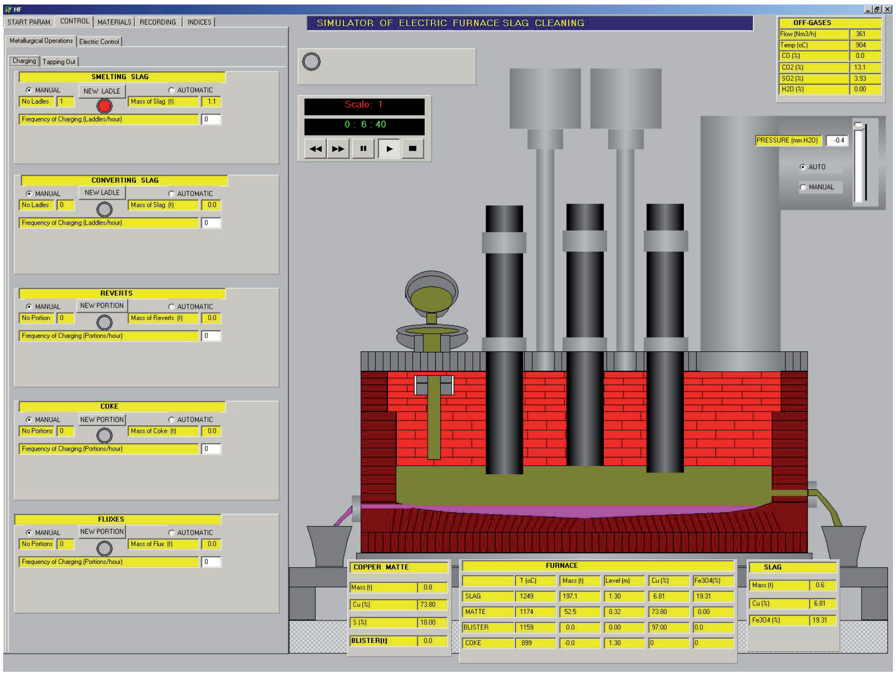

Figure 1: Copper slag cleaning simulator screenshot

2 CHOICE OF PROGRAM

The choice is usually a compromise between accuracy, “in house knowledge”, availability and targets. SMS Siemags choice for Delphi is briefly explained in the following paragraphs

2.1 Accuracy

This simulator is absolutely not intended to compete with programs like Fluent. Secondly, the Cu-slag cleaning furnace simulator, developed by professor A. Warczok, demonstrated that no complicated thermodynamical models are required to achieve accurate results with Delphi. However, it demands some “old school” writing down of formulas in a computer language.

2.2 In house knowledge

Both the experience with the Cu-slag simulator as well as SMS Siemags own experience with Delphi and Simusage [2,3], a Delphi add-in for thermodynamical modelling based on Factsage[4], made Delphi the program of choice. Fluent[5] could also used, but was considered to exhaustive and lacked the interactive possibilities Delphi offers. This was also the reason not to work with Femlab [6].

2.3 Availability

A number of programs are suitable for building a simulator. Ideas[7], Unisim[8] or Matlab[9] in combination with Simulink[10] could probably all do similar things as presented here. Availability of the program licenses for Delphi within SMS Siemag, lack of licenses and funds for the others, made the choice based on availability an easy one.

2.4 Targets

In this case the target of training operators required a method that

- Would demonstrate the primary job of furnace operators: balancing energy and material input with slag and metal output.
- Would interact with the user: doing any action always has a certain positive or negative impact on furnace operation.
- Would not spend too much time in calculating details that do not matter significantly.
- Would demonstrate some basic other situations like electrode breakage or arcing and their effect on the operation.

- Would demonstrate these without costing operational downtime. It would be a very expensive training to deliberately disturb a furnace running stably by breaking electrodes, or damage the roof by arcing.
- Would do all of this faster than in real time. A lot of effects take a long time to develop in reality (like imbalances in tapping and smelting). It is therefore practical to be able to have a fast forward option

Due to the experience gained from the copper simulator, and looking at future expansion of the model, it was rapidly decided to go for Delphi. It must be remembered that different programs have different features and purposes. The choice of program should therefore be a careful one, with awareness of all advantages and disadvantages, strengths and weaknesses. It is also acknowledged not all known software tools or platforms available are mentioned.

3 MAIN INTERFACES

The first interface is a top view of the furnace including the kiln, calcine transport cars and calcine containers, the locations for the crane, the furnace bins, the furnace and the charging chutes. Optional is to show more details for tapping slag and metal as well as details on the furnace bunkers or controls for tapping.

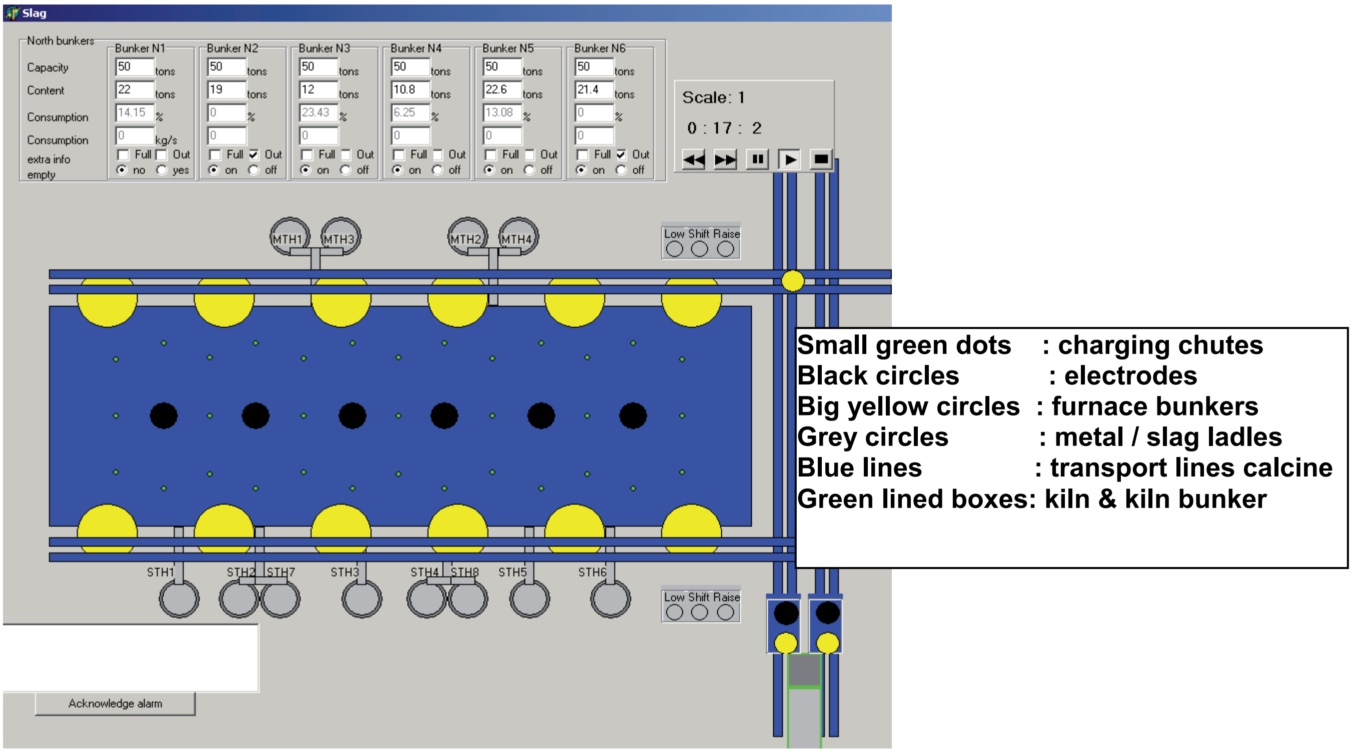

Figure 2: Main screen 1 of the FeNi simulator

In the second main screen, a cross section of the furnace is displayed. It includes all electrode columns with slipping and regulating cylinders, as well as metal, slag and calcine present within the furnace. The one electrode having a red "holding ring", is in the middle of a slipping sequence.

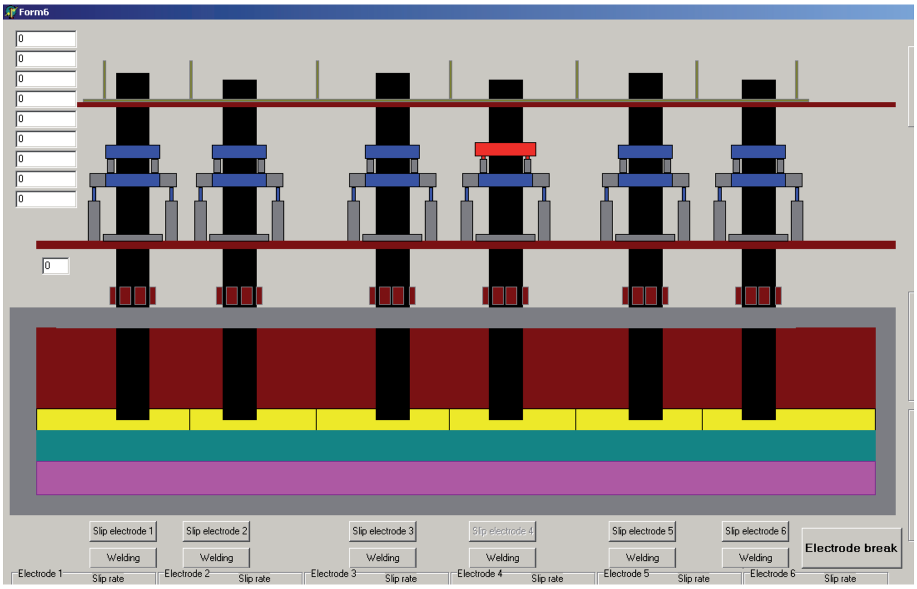

Figure 3: Main screen 2 of the FeNi simulator

4 SIMULATOR MODELS

Where in real life the mass and energy balances are back-calculated based on the data collected, in a simulator, data need to be created first to be used in calculations.The description of the models starts with a brief explanation how the energy input was simulated followed by the models for smelting, charging and tapping. The results from these models are combined in the energy and mass balance.

4.1 Electrical

Where in reality a transformer is installed and supplies energy and data, now every single value has to be created by the program. These values have to interact with other program parts that calculate the bath levels, which again are linked to tapping and charging. As the new generation of FeNi furnaces typically operates with a covered arc, the algorithms describing power input must cover a wide range of conditions:

- short circuiting/high currents if the electrodes approach the metal level
- resistance heating if the electrode is immersed in the slag bath
- arcing when the electrode is raised out of the slag

Covering these three areas is achieved by having different formulas depending on the electrode position:

- Electrode tip close to the metal bath where the impedance rapidly decreases
- Electrode tip in slag where the impedance slowly increases
- Electrode tip above slag where the impedance rapidly increases until contact is lost

These interactions can be observed during simulated operation when raising or lowering the electrodes.

In the model following items have been addressed, among others:

- The transformers can operate in manual, current and impedance mode, where settings can be adjusted as desired.
- If set points are selected, then the electrode regulation will adjust the electrode position if a certain threshold in passed where thyristor control can not compensate the offset.
- As the electrodes approach the metal bath, the impedance will decrease up to the point where the transformer will trip on high current.
- Above 20 mOhm per electrode two random functions are used to add instability to the power input, simulating arcing. The arcing instability is reduced by the thyristors. The level of stability is mathematically related to the impedance, creating more instability at higher impedance.
- In manual mode the electrical values will change with rising/decreasing slag and metal levels.

- In automatic mode the electrodes will move up or down depending on the bath levels and settings.

4.2 Smelting

Whether the furnace is operating in arc mode (raised electrodes) or in resistance mode (immersed electrodes) has a significant impact on the smelting pattern within the furnace. Ratios for calcine consumption in the centre of the furnace compared to the side may well be up to 9 to 1 when the furnace is operated in arc mode, but is much lower in resistance mode: the electrical settings will influence the calcine consumption for each m^2 in the furnace. Positioning of the charging chutes also plays an important role. Both have an impact on how much calcine needs to be charged per chute in a certain area to maintain the surface covered, and the calcine consumption per chute determines when the connected bin needs to be refilled. To have a model that calculates the calcine consumption inside the furnace in relation to the electrical settings, following set up was chosen:

1. the furnace is divided into grid cells of 250 x 250 mm, which means a matrix of 140 x 45 cells for a 35 x 11.25 m furnace
2. each electrode has a power input, divided in arc and bath power
3. all grid cells of the furnace get a fraction of the arc and bath power from each electrode depending on the distance to that electrode
4. arc and bath power of all electrodes are added up for each grid cell
5. each grid cell thus has a power available that is used for smelting calcine

$$k_{i,j} \rightarrow k_{x,y} \qquad \begin{matrix} k_x = (i-0.5)*\dfrac{W}{I} & i>0 \\ k_y = (j-0.5)*\dfrac{L}{J} & j>0 \end{matrix} \quad *) \qquad (1)$$

*) some simplification in these formulas as W/I and L/J must give integer values

Where: $k_{i,j}$ is any cell within in the grid in row i and column j;

i = row number of the grid cell;

I = total of row numbers;

j = column number of the grid cell;

J =total of column numbers;

L = furnace length in millimetres;

W = furnace width in millimetres;

$k_{x,y}$ = any grid cell now with coordinates in x and y in millimetres;

With the grid cell row and column number transferred to coordinates based on the furnace dimensions, and knowing the position of the electrodes, the distance between each grid cell and each electrode can be calculated.

$$d_{i,j,m} = \sqrt{(k_x - E_x^m)^2 + (k_y - E_y^m)^2} \qquad (2)$$

Where: $d_{i,j,m}$ is the distance of any cell in row i and column j to electrode m in millimetres;

E^m = Electrode with number m;

m = electrode number;

$E^m_{x,y}$ = x and y coordinates of electrode m in millimetres;

Next step is to use a function f to describe the decrease of smelting power related to increasing distance between any grid cell and a certain electrode m. This results in a relative distribution of power around each electrode.

$$k^v_{i,j,m} = \frac{a * P_m}{(d_{i,j,m})^r} \qquad (3)$$

Where: $k^{v}_{i,j,m}$ is a value assigned to a grid cell based on power and distance between electrode and grid cell;

P_m = power (arc or bath) released at electrode m;
r = constant 1;
a = constant 2;

The constants r and a are chosen differently when describing bath and arc power and can be fine-tuned when actual smelting data are available

Adding all values $k^{v}_{i,j,m}$ up and dividing the available power for an electrode by this number, gives the factor $k^{factor}_{i,j,m}$ by which $k^{v}_{i,j,m}$ has to be multiplied to obtain the power available in each grid cell per electrode.

$$k^{factor,type}_{i,j,m} = \frac{P^{type}_{m}}{\sum_{i=1}^{I}\sum_{j=1}^{J} k^{v}_{i,j,m}} \quad (4)$$

Where: $k^{factor,type}_{i,j,m}$ is the factor to multiply each grid cell value $k^{v}_{i,j,m}$ with to obtain the power per grid cell;

type = either arc or bath power;

Calculations for arc and bath power are presently similar except for different constants in formula (4)

Adding up arc and bath power from 6 electrodes in each individual cell gives the total power input that is used for smelting divided over all grid cells of the furnace:

$$P^{total}_{i,j} = \sum_{m=1}^{6} (P^{arc}_{i,j,m} + P^{bath}_{i,j,m}) \quad (5)$$

Where: $P^{total}_{i,j}$ is the arc and bath power available for smelting per grid cell, depending of the distance between the grid cell, all electrodes and electrical settings of that electrode;

Results for different charging patterns as a result of different electrical parameters are given in graphs under the paragraph “Results of running the simulator”.

4.3 Charging

For the four furnaces under construction in Brazil, charging of calcine into the furnace is controlled automatically by the radar system. The moment the level of calcine is below the lower set point, a batch is charged. Is the calcine above the upper limit, then calcine discharging is interlocked. This means that if no or small amounts of calcine are being charged compared to the furnace load, a charging tube is blocked or the slag level is too high and more slag tapping (or lowering furnace load) is required. This can all be simulated.

Based on the consumption per m^2 in the furnace and the positioning of the charging chutes, it is calculated when a batch of calcine needs to be charged.

A typical operator problem is to know how to balance available power with available calcine. The production of FeNi is done in a chain of unit operations (mining, drying, calcining, smelting, refining), and any unit can encounter problems, which will have an impact on the furnace operation. For example, in case not enough calcine is available, the operator can reduce power to stabilise the furnace or take the risk of running out of feed at some point.

The simulator provides for a kiln, 2 calcine transfer cars, 2 cranes and 3 calcine transfer containers. At the furnace a total of 12 feed bins and 33 chutes have been installed, each chute equipped with radar. Each piece of equipment can be switched off, showing how this will impact the operation. The kiln can be set to any production level within a fixed range. It is thus possible to see what happens if calcine production and smelting are unbalanced. One of the tab sheets within the simulator allows verifying if these are in balance, and, if not, how much time is left before the feed bins are either filled or empty.

4.4 Tapping

To add tapping to the simulator, it took three stages to reach a realistic furnace operation:

1. Step 1 was to open a tap hole and have a material flow coming out. To follow how much material was tapped from where, have a good mass balance and to be able to make tapping automatic (based on time or weight) more detailed programming had to be done in stage 2
2. By registering in detail the number of taps, the quantity tapped, the time of tapping per tap hole, linking this to the bath levels and introducing the options to have automated tapping sequences, bath levels could be maintained in fast forward better than could be with "manual" tapping only. This opened the door to was more realistic furnace operation in stage 3
3. With all information from stage 2, multiple situations that can occur in operation can be simulated, for example using a tilting launder, spillage if a ladle is missing, interrupting tapping, wear and tapping slag from the metal tap hole.

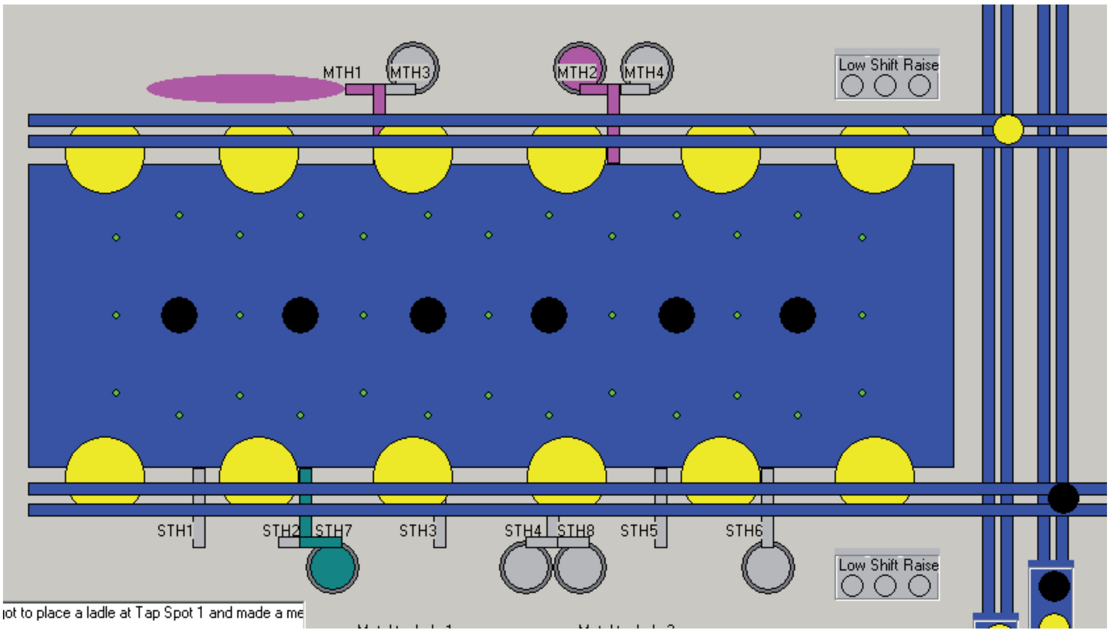

Figure 4: Tapping slag (green) and metal (pink), including spillage at Metal Tap Hole 1

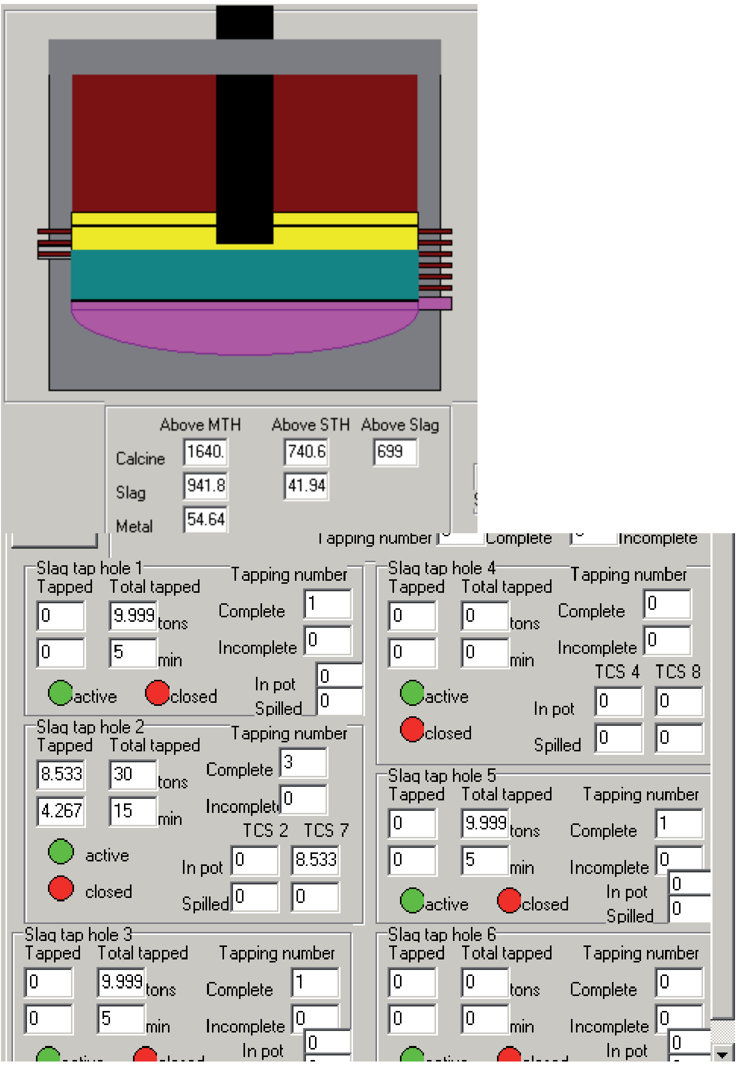

Figure 5: Cross section of the furnace and detail of slag tapping data

4.5 Mass & energy balance

The mass balance is based on a certain calcine coming in with a Ni content, on the calcine/ton of FeNi ratio and on the Ni content in the FeNi. The metallurgical energy consumption per ton of FeNi can be varied. This allows fro showing the effects on the operation of differences in calcine or Ni content in the produced metal, without having to make detailed elemental mass balances.

The mass- and energy balances of this model were kept simple for a number of reasons. First of all, programming a six-in-line furnace takes enough time as it is, and doing the mass balance in detail did not seem a good value for effort at the time. A separate Delphi model exists that does take detailed mineralogical compositions into account, yet this proved so elaborate that this operational simulator would turn unworkable if incorporated. Software like Simusage could bypass this problem.

Secondly, as its main use is to give operators a basic idea of the interaction between power input, smelting and tapping, it was considered out of scope to have them consider the effects of minor changes in calcine composition. Variations in, for example calcine temperature, can easily be introduced by increasing the metallurgical energy required per ton of FeNi.

Thirdly, once a mining plan is set and the plant is running, the input material will not vary significantly over short time spans. Calcine and energy consumption per ton of FeNi will remain relatively stable. In this model it would be possible to follow how the change in ore composition or Ni content effects the operation roughly and its economics. The effect of minor changes on energy consumption, impurity behaviour, smelting behaviour, liquidus temperature and effects on the side wall refractory are better served by specialist software like Factsage or Fluent. Because modelling of the interaction between slag and sidewall is out of scope, detailed compositional calculations have been bypassed (see choice of program).

4.6 Accuracy

It is acknowledged that the models used are not comparable to those available in Fluent. However, as stated in the targets, the model was developed to show the interaction between tapping – charging – power input. It is the authors believe this has succeeded to a large extent using basic equations and assumptions. The model will allow for fine tuning making it more realistic once the real furnaces are on line and actual smelting patterns, electrical parameters and further specific operational values can be implemented.

5 RESULTS OF RUNNING THE SIMULATOR

Below is a given situation with 3 transformers equally balanced, with electrical values as can be seen in the right side of figure 6. The model calculation shows that 4 hot spots exist below the centre electrodes (the lower of the 2 small rectangular surface views), whereas the outer electrodes consume less material. This is typical for rather deeply immersed electrodes, or low impedance values.

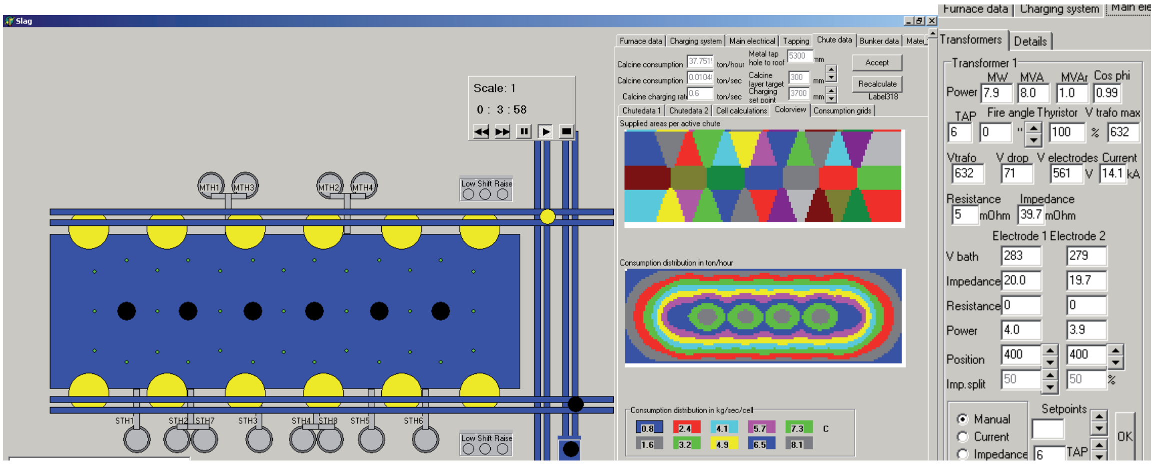

Figure 6: Furnace smelting pattern with immersed electrodes

In the following screen shot, the electrodes have been raised until all were above the slag level, and the effect of the arc on increased smelting behaviour is visible. The upper right drawing with the triangles depicts the areas of influence per charging chute. Combining these two sources of data, makes it possible to calculate the calcine consumption per charging chute, and thus per bunker.

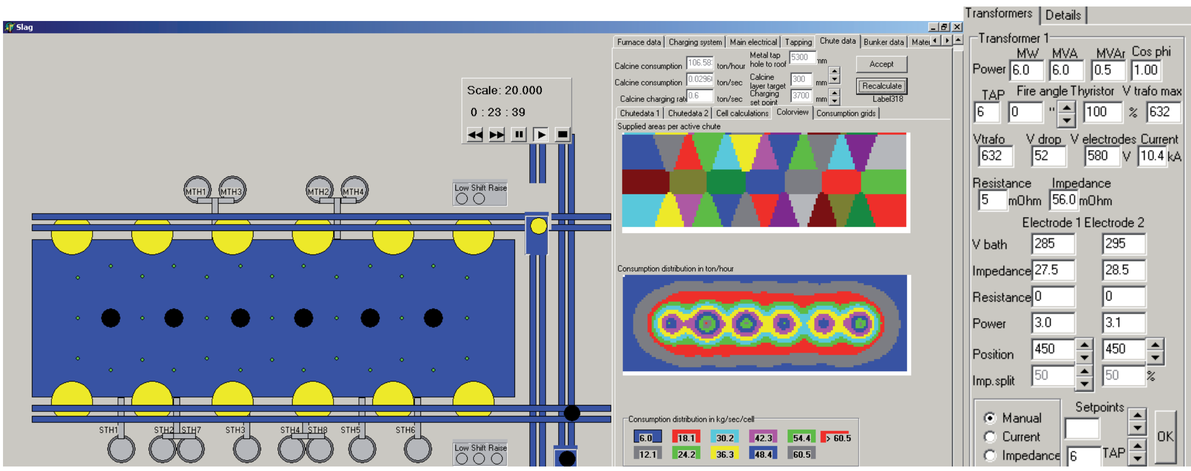

Figure 7: Furnace smelting pattern with raised electrodes

The last screen shot is an image, where some of the charging chutes have been deactivated. As can be seen, some bunkers get larger areas of influence, which automatically means that calcine consumption for some bunkers will increase, as would happen in real operation.

The reason that the smelting patterns have changed as well, is due to the fact that an electrode break had been triggered: even though the electrode did move down, as the transformer was in impedance mode, no contact could be re-established. As a result, smelting only takes part in 2/3 of the furnace.

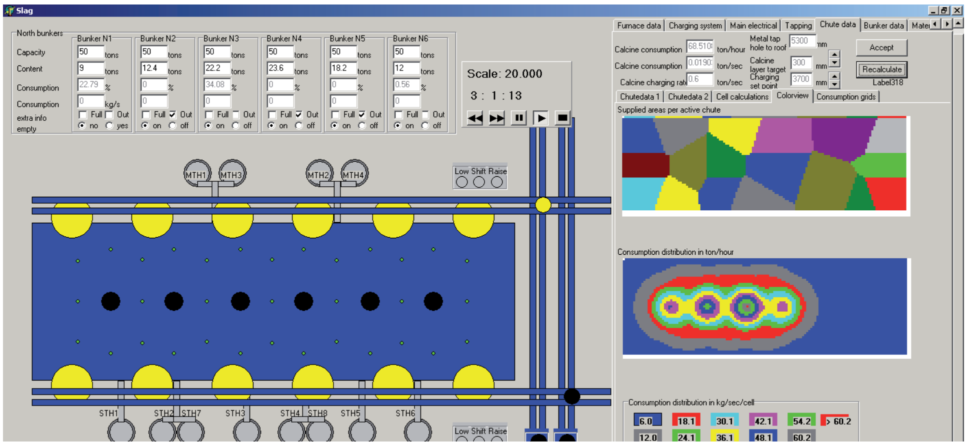

Figure 8: Furnace pattern with 1 broken electrode; feed bin charging pattern with multiple empty bunkers

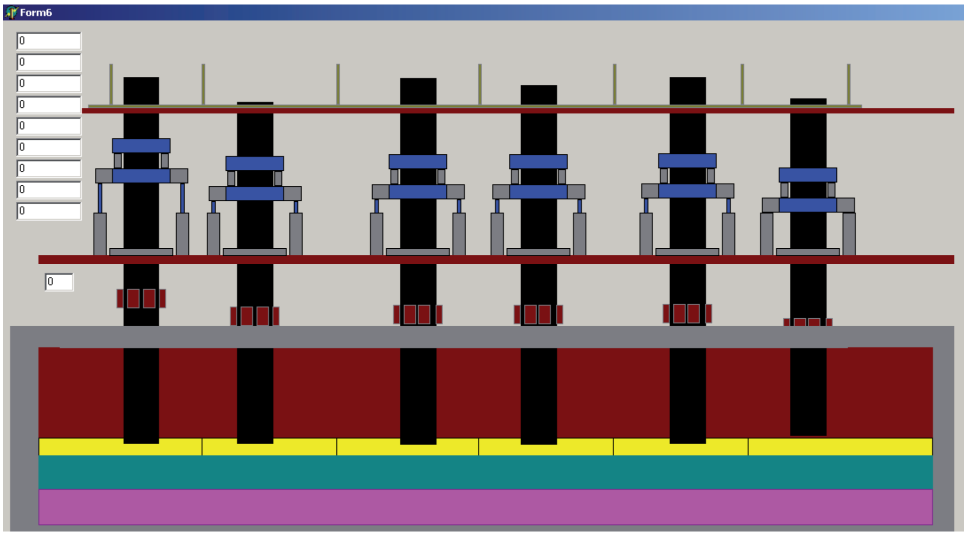

Figure 9: Cross section of the furnace with 1 broken electrode (nr 6)

6 FUTURE DEVELOPMENTS

6.1 Mass & energy balance improvement

The mass and energy balance in this simulator are relatively simple, for mentioned reasons. An advantage is that there are relatively few data that would need to be replaced by a thermodynamically more accurate program like Simusage or with another Delphi model that does dynamically follow calcine compositional changes (but is not linked to a furnace yet).

6.2 Parameter optimisation

Once the real furnaces are running, data from site can be used to adjust the various model parameters in such a way that the simulator is even more realistic. Especially the smelting pattern may prove an interesting possibility.

6.3 Actual HMI screens.

If it is known what the design of the actual HMI screens is, these could be integrated into the simulator to make it more realistic. If real HMI's should be applied for operator training, other software may be more appropriate due to flexibility in equipment modelling (cooling systems, interaction of equipment, control room possibilities).

7 RECOMMENDATIONS

To realise a project like the one presented here, is not an easy task. A lot of time can be saved and problems prevented if the project is started correctly with a proper preparation. Some advice is given in the following paragraph.

1: spend time setting targets & deadlines

Reflect on what the purpose of the model is. Then reflect again. And again.
Important questions at the beginning are:

- What are the targets?
- Who will it be for?
- Who will have to work with it now as a user?
- Who will have to work with it in the future as a programmer?

A detailed time schedule may help, but in the authors' experience the development comes in waves. Sometimes a minor issue can take weeks to program, sometimes huge progress is made in a day. The time needed for fine tuning the model and debugging should not be underestimated. This is more in the order of weeks than days, even if it is possibly to focus on the project continuously.

2: select the proper program

Many different software packages exist with many different applications, strengths and weaknesses. Select the tool most suitable for your needs and then still make sure to be aware of the weaknesses it has. Black box models may save a lot of time, but may make it difficult to adjust the model to your needs. Standard simulation software may have the advantage of offering more realistic and standardised interfaces, whereas with software like Delphi, all interfaces must be created, which is a very time consuming exercise and may be far from reality.

3: spend more time setting limits

In the process of writing a program and seeing the model grow, more ideas and possibilities rise. Some of these ideas can be brilliant, funny, interesting, simply tempting, or all at the same time. The coloured smelting pattern was one of these side tracks for example. Stay on track. Getting distracted will normally cost tremendous amounts of time and the deadline.

4: plan a structure for the simulator:

A big risk of beginning to produce code without having a frame in which the code is placed, is that at some point the text expands to thousands and thousands of lines, where it is impossible to see a logic structure and make changes when later required. It is recommendable to use available options within the software used. Delphi offers the possibility to structure the program in separate units. Within each unit procedures and functions can be created, that can be called upon in the other units. This makes it easier to adapt parts later. Used units and documents were for example given in table 1, the first 2 columns.

5: list the variables required:

Create lists with variables that will all be required (see table 1, 4^{th} and 5^{th} column). Talk to experts from various departments (electrical, metallurgical, etc) to collect as many variables, parameters limits and experience. As an example, for a 6-in-line furnace, 3 transformers and 6 electrodes are needed. These 6 electrodes will have a current per transformer, but individual impedance and resistance depending on the tip position. These data combined give voltages for the arc and the bath. Setting the limits and using old data (from a previous time step for example) increase the quantity of variables per unit tremendously.

Once it is known what variables will be used, these can be structured as well, which can save a lot of programming time. Also to be considered are mechanical, electrical and physical limits, as these all need to be written into the program as well.

Table 1: Example of structure within a program

Unit	Procedures		Procedure	Variable
Electrical	MainElectrical Slipping WeldACase ElectrodeBreak		Slipping	SlippingDistance HoldingTime ElectrodeTop etc
Charging	StatusCheck BunkerFilling ChargeLeft ChargeRight ChargeFirst ChargeLightest LadleCar1 LadleCar2		LadleCar1	Position1 Position2 Position3 Active Waiting Speed1 Speed2 etc
Unit4	DrawInFurnace Grid cellCalculation ConnectGridToChutes ChuteCalculation CalculateHeight SmeltingProducts MetalTapping SlagTapping		MetalTapping	TapHole[i].Open TapHole[i].LadlePresent Taphole[i].Flow TapHole[i].Weight TapHole[i].Time TapHole[i].TargetWeight TapHole[i].TargetTime etc

6: combine ideas for structure and variables

Try to place all variables (point 5) into the structure (point 4). Can all variables be logically placed under one of the units? If not, are the variables necessary or is there something missing in the structure?

7: repeat

Go through this list with recommendations a number of times to fine tune the structure. While doing this, talk to specialists from various departments (electrical, process control, metallurgists, operators, etc) to have a structure as complete as can be. Building such a model is an iterative process. The more thought is given before starting to program, the less work will have to be done while programming.

8: build models first in Excel and other accessible software

Excel is a very practical tool to test is a certain calculation could work or to verify if calculation models are correct. At all times it is possible to see what is happening within a calculation, and there is probably no easier way to change parameters and follow up on the impact. It is also possible to create tables to see what should happen in time. For quick verification of mass and energy balances, especially when the code for them has been written your self, software like HSC[11] or Pyrosim[12] is recommendable.

8 CONCLUSIONS

A simulator has been built based on models for power input, smelting patterns and charging sequences. Operational actions like tapping and electrode management are implemented as well. Though it is not designed to be academically exact, it is the authors' believe that the simulator can be made to match reality very close once the 6-in-line FeNi furnaces in Brazil are operating. It is already capable of teaching the interaction between power input, charging and tapping and can be used as a training tool.
Delphi is a program that can very well be used for exercises as described above. One must always be aware of strengths and limitations of a certain software package. Incorporating Simusage/Factsage thermodynamical features could improve the metallurgical accuracy.

9 REFERENCES

[1] Warczok, A., Riveros, G., Degel, R., Kunze, J., Oterdoom, H.,Computer simulator of slag cleaning in an electric furnace, Proceedings of the Carlos Diaz Symposium, Book 2, Volume III, 2007, pp367-378

[2] http://www.gtt-technologies.de/simusage

[3] http://gttserv.lth.rwth-aachen.de/~cg/Software/SimuSage/DynaFrame.htm?/~cg/Software/SimuSage/Brochure/GeneralConcept.htm

[4] http://www.factsage.com/

[5] http://www.fluent.de/

[6] http://www.femlab.de/

[7] http://www.ideas-simulation.com/

[8] http://hpsweb.honeywell.com/Cultures/en-US/Products/ControlApplications/simulation/default.htm

[9] http://www.mathworks.de/

[10] http://www.mathworks.de/products/simulink/

[11] http://www.outotec.com/pages/Page.aspx?id=21783&epslanguage=EN

[12] http://www.pyrometallurgy.co.za/Mintek/Pyrosim/Pyrosim.htm