Training Simulator for Flotation Process Operators

Timo Roine*, Jani Kaartinen*, Pertti Lamberg**

**Aalto University, Department of Automation and Systems Technology*
Espoo, Finland (e-mail: timo.roine@tkk.fi, jani.kaartinen@tkk.fi).
***Luleå University of Technology, Division of Sustainable Process Engineering, Luleå, Sweden (e-mail:pertti.lamberg@ltu.se)*

Abstract: This paper presents a novel simulation concept for operator training in the field of mineral processing. The simulations are carried out with a dynamic process simulator HSC Sim® of HSC Chemistry® developed by Outotec Research Oy. The simulator is fitted to mimic an existing copper flotation circuit as accurately as possible by using metallurgical models and then integrated into a larger simulation environment, providing the operator trainees a realistic experience of the process. The simulation environment is designed to be scalable and very flexible, allowing many different usage scenarios and thus aiding in the transfer of the tacit knowledge from operator generation to the next. Concurrent work is being done on higher level analysis, utilizing the results reported in this paper.

Keywords: process simulators, operators, process models, training, process automation

1. INTRODUCTION

Mineral flotation is a complex separation process that typically contains several stages and multiple feedback loops (i.e. circulating loads). Also, the reagents that are used vary and often have opposing effects. This makes the process difficult to control, at least in an optimal manner (for further details about flotation and mineral processing in general, see e.g. Wills and Napier-Munn, 2006, Finch and Dobby, 1990, and King, 2001). For these reasons, the actions of the process operators and differences in their operating behaviour play a significant role in the performance of the flotation plant.

Training of the operators in mineral processing has traditionally been carried out by teaching the basics of the process to the students and then letting them follow more experienced operators at work. Due to the increase in computing power and decrease in prices of computer hardware, training simulator software is becoming an important factor in different application areas. This type of simulation software has been in use, for example, in nuclear power plants and in aviation for a long time. However, in mineral flotation, the use of such simulators has been limited. This is not to say that simulation has not been utilized in the mineral industry; there are many simulation based applications in common use, for example: *JKSimMet®* (McKee and Napier-Munn, 1990)*, Dynafrag* (Desbiens et al., 1997, Flament et al., 1997), *JKSimFloat®* (Schwarz and Alexander, 2006)*, USIM-PAC®* (Brochot et al. 2002) *HSC Sim® 7.0* (Outotec, 2006, Roine and Kotiranta, 2007, Lamberg and Bernal, 2009), but they are typically focused on aiding in design or control of the process rather than helping in operator training. Furthermore, flotation models in these solutions have typically been empirical and have allowed only steady-state analysis capabilities.

Modelling of the flotation process is very difficult due to the complex physicochemical reactions and feedbacks of the process. The micro processes that can be identified in flotation are: 1) particle-air bubble collision, 2) particle-bubble attachment, 3) rise of the bubble, 4) detachment of particle from bubble, 5) froth processes (King, 2001). It is very challenging to create a physical model even in simple controlled systems (Miettinen et al., 2010). Flotation, however, involves chemistry, too. To make minerals hydrophobic, i.e. floatable, they are treated with collector chemicals which change the surface of mineral particles. Chemically these reactions are complex and their measurement in industrial applications is difficult. Finally, there is a challenge from the complexity of the material. True flotation plant feeds have wide size distribution, complex mineralogy and wide range of different liberated, binary and multi-mineral particles. Therefore, for flow sheet development and process improvement, empirical and more practical approaches are used (Runge et al., 1997). It is common to bind all pulp sub processes under a simple kinetic flotation model and call this part of the model true flotation. The froth processes are combined under the froth recovery models. The third important component in the empirical models is to handle water and entrainment. Entrainment is defined as the unclassified part of solid material that is carried by water into the concentrate.

To improve the training of process operators, a training simulator environment has been created and is described in this paper. It consists of 1) flotation process simulations performed in HSC Chemistry® (Outotec, 2006), 2) process logic emulation by means of software developed in Matlab®, and based on Outotec's Proscon® automation system, and 3) Proficy/HMI Cimplicity® automation software for control and visualization. In addition, supervisory teacher software has been developed to manage the student training environments.

Different scenarios can be used in the training simulator to train inexperienced operators, as well as to improve process knowledge of senior operators. The environment can also be used to collect information of the operator actions and analyse and compare the performance of different operators.

In addition, the system and the collected data can serve as a valuable means to convey important silent knowledge to following operator generations. Another valuable asset of the system is that it can be used to train operators even before the construction of the actual plant is completed.

Although the HSC Chemistry® simulation software can be used in a variety of different applications, in this particular scenario it is used as an integral part of a virtual training environment in order: 1) to get the trainees acquainted with metallurgical unit processes, 2) to provide a realistic feel and response to the changes in metallurgical parameters and to control actions made by the trainees, and 3) to provide a tool for improving strategies and scenarios for process control and development.

The training simulator presented in this paper utilizes the same simulation engine as the Virtual Experience of Outotec (Moilanen and Lamberg, 2010), but with a completely different design. Firstly, the simulation model has been fitted to match the copper flotation circuit of Inmet Mining Corporation's Pyhäsalmi mine in central Finland (details in Section 2). Secondly, the simulation environment is designed to be flexible enough to comply with different usage scenarios, scalable in the number of concurrent simulations, and distributed so that simulation speed can be increased by running CPU intensive tasks simultaneously in several computers. Furthermore, the distributed nature of the system allows also physical distribution, meaning that teaching can be done via Internet. One example of the several possible setup scenarios is shown in Fig. 1.

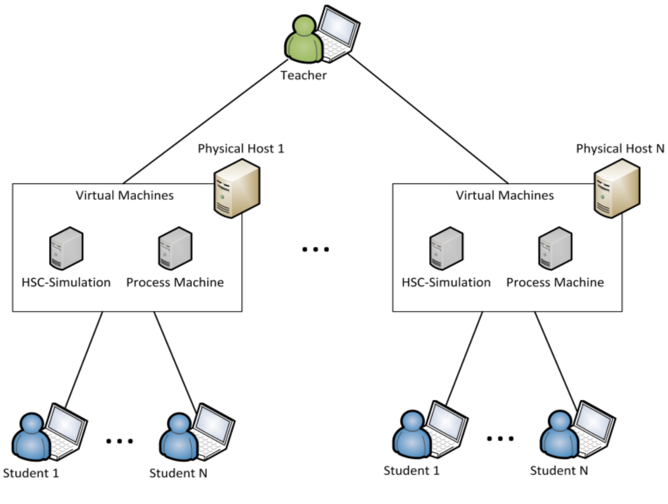

Fig. 1. Example setup for the training environment.

As it can be seen, one teacher can control several simulations and each of them is realized with two virtual machines (Smith and Nair, 2005). The virtual machines can be distributed into one or several physical machines, depending on the performance requirements as well as on the capabilities of the physical machine(s). In this case, each simulation is run in a separate physical machine and an arbitrary number of students can connect to each simulation, making it possible to have dedicated simulations for each student, or to let the students work as a team. The structure, implementation, and communication aspects of the training environment are covered in Section 3.

The basic idea in the training system presented in this paper is to mimic the operational behaviour of an existing flotation circuit as closely as possible and then use the generated model with copies of the existing displays being used in the plant. This makes it possible for the operator trainees to get very realistic experience with the simulated process. Another usage scenario is to run the simulation model in parallel with the actual process and use it to give foreknowledge of – say – the implications of a given control action. These and few other usage scenarios are described further in Section 4.

2. SIMULATION MODEL

Dynamic model of flotation built in HSC Sim® is largely based on the AMIRA P9 models (Vera, 1999, Zheng et al., 2006, Welsby et al., 2010). As the P9 models have been developed for steady-state simulation the dynamic model uses differential equations with small (1 to 5 seconds) simulation time steps. The entrainment and froth recovery models have been adopted from Neethling (2003) and Neething and Cilliers (2002a, b, 2009). These are described in more detail later in the text. Flotation cell is divided in two mass balance areas: pulp and froth. Particles flow from pulp to froth by two mechanisms: true flotation and entrainment. With current model the entrainment passes directly through the froth into the concentrate. Flux by true flotation is modelled using first order kinetic equation and the flux from the froth to the concentrate with the froth recovery model.

Solid material is described as particles, each representing a particle class, and having properties like size, composition and specific gravity. In principle the model is capable to handle multiphase particles but as a first approximation a floatability component approach has been used. Each mineral is divided in each size class into three components: fast floating, slow floating and non-floating – in the Pyhäsalmi case a total of 75 particle classes (5 minerals x 5 size classes x 3 components). Liquid phase includes water and reagents: collector and frother. Collector reacts in the conditioning stages immediately and resets the mass proportions of components for each mineral by size class. Frother follows liquid phase but in a flotation cell it is divided with a fixed ratio between the froth (concentrate) and liquid (tail).

In HSC Sim® the unit model is a DLL file and the main program takes care of material transport between the units. Pyhäsalmi copper circuit simulation consists of 17 flotation cells, 2 conditioners, 14 pump/sumps and two on-line analysers. Delay caused by pipes is currently ignored. In each flotation unit the calculation within a simulation step goes: 1) take the new input into the cell and mix it totally with the pulp, 2) calculate the flux of each particle type into the froth by true flotation, 3) calculate, according to froth recovery, the flux of each particle class from the froth to the concentrate and, through drainage, back to the pulp phase, 4) for the current pulp calculate water flux into the concentrate and flux of each particle class into the concentrate by entrainment, 5) for the remaining pulp calculate the flux of each particle class and water into the tail according to tailing valve opening, 6) for the remaining pulp calculate the pulp level in the cell, 7) calculate the new value for the tailing valve opening according to pulp level PID control.

2.1 Equations for recovery by true flotation

In the dynamic flotation model the mass flow rate (flux) of a particle (class) i transferred from the pulp to the concentrate ($m_{i,t}$) in a time interval (dt) is calculated according to equation

$$m_{i,t}{=}m_{i,p}k_{\mathrm{i,c}}\mathrm{dt} \tag{1}$$

where $m_{i,p}$ is the mass of particle class i in the pulp and $k_{i,c}$ is the flotation rate constant of particle class i in the collection zone.

The flotation rate constant of a particle class i, $k_{i,c}$ is a function of floatability of the particle class (P_i) and bubble surface area flux (S_b) (Gorain 1997, 1999):

$$k_{i,c} \ = \ P_i S_b \tag{2}$$

The bubble surface area flux S_b is related to airflow speed in the flotation cell (J_g, m/s) and bubble diameter (d_{32}, Sauter average, m) as follows (Finch and Dobby, 1990)

$$S_b = \frac{6J_g}{d_{32}} \tag{3}$$

Superficial gas velocity J_g (m/s) can be calculated from the air flow rate Q_{AIR} (m^3/s) and cross sectional area of the flotation cell (A, m^2) using equation (Gorain, 1997, 1999)

$$J_g = \frac{Q_{AIR}}{A} \tag{4}$$

2.3 Froth recovery

Froth recovery $R_{i,f}$ for particle class i, i.e. mass proportion of particles passing through the froth of the true floated ones is:

$$\begin{cases} R_{i,f} = \left(\frac{\alpha(1-\alpha)v_g}{v_{set}}\right)^{\frac{f}{2}} \left(\frac{r_{in}}{r_{out}}\right)^f & ,if\ \alpha < \frac{1}{2} \\ R_{i,f} = \left(\frac{v_g}{4v_{i,set}}\right)^{\frac{f}{2}} \left(\frac{r_{in}}{r_{out}}\right)^f & ,if\ \alpha \geq \frac{1}{2} \end{cases} \tag{5}$$

where f is the fraction of attached material that becomes detached from lost surface area during coalescence, and v_g is vertical gas velocity, r_{in} and r_{out} are the bubble sizes in the bottom and top of the froth, respectively (Neethling, 2008).

2.2 Entrainment

Mass flow of particle class i directly from the pulp through the froth to the concentrate by entrainment $m_{i,E}$ (i.e. unclassified particles coming with water) is calculated as

$$m_{i,E} = E_i J_w \tag{6}$$

where E_i is the degree of entrainment of a particle class i and J_w is water flux into the concentrate (Savassi et al., 1998). The degree of entrainment is calculated for each particle class using the approach of Neethling & Cilliers (2002a, b, 2009).

$$\begin{cases} E_i = \exp\left(-\frac{v_{i,set}^{1.5} h_{froth}}{D_{Axial}\sqrt{J_g}\alpha(1-\alpha)}\right) & ,if\ \alpha < \frac{1}{2} \\ E_i = \exp\left(-\frac{2v_{i,set}^{1.5} h_{froth}}{D_{Axial}\sqrt{J_g}}\right) & ,if\ \alpha \geq \frac{1}{2} \end{cases} \tag{7}$$

where $v_{i,set}$ is particle settling velocity, h_{froth} is froth height, D_{Axial} is axial dispersion coefficient and α is air recovery. Water flux is calculated using the formula (Neethling, 2003)

$$\begin{cases} J_l = \frac{J_g^2 \lambda_{out}}{k_1}(1-\alpha)\alpha & ,if\ \alpha < \frac{1}{2} \\ J_l = \frac{J_g^2 \lambda_{out}}{k_1}(1-\alpha)\alpha & ,if\ \alpha \geq \frac{1}{2} \end{cases} \tag{8}$$

where J_l is water flux, λ_{out} is the length of Plateau borders per volume of froth calculated using the average overflowing bubble size and k_1 is a physical parameter combining particle density, Plateau border drag coefficient and liquid viscosity.

2.4 Tail flow rate

In the steady-state process all material coming into the flotation tank is sent to concentrate and tailing. In the dynamic process the tailing flow rate will be dictated by the tailing port. Tailing port is adjusted with PID control to keep the froth depth in set-point (Lamberg et al., 2009).

2.4 Defining the model parameters

The dynamic flotation model equations (1-8) include a large number of parameters, many of them difficult to measure or define. In the first generation simulations we have adopted reasonable fixed values for many of them (like bubble size, axial dispersion coefficient, length of Plateau borders and even air recovery). However, we have found out that even with such a simplification, simulation is capable to produce realistic responses and trends, and therefore is suitable for operator training to learn the principles. Model calibration and validation remains a work to be done in the future.

3. TRAINING ENVIRONMENT

As shown in Fig. 1, each simulation is comprised of two virtual machines communicating with each other. One is running the simulation engine (HSC-Simulation) and the other (Process Machine) is running the rest (Cimplicity automation server, logic emulation, database connections, etc.). This section describes the inner workings of a single simulation and shows how the distributed nature of the environment has been realized. As the system is built into virtual machines, the scalability is made easy; new simulation environments are created by just copying the virtual hard drives of the two virtual machines and by setting new IP-addresses for them. Everything else remains the same.

3.1 General structure

The training environment consists of four separate parts, which are divided into virtual machines, or VMs (see Fig. 2). The four parts are HSC-Simulation, process machine, operator interface and teacher user interface. The actual simulations take place in the HSC-Simulation VM using HSC Simulation software and an accurate model of the process. The process machine includes the server part of the Cimplicity automation software and the logic emulator, and is thus playing the part of the automation system. The Cimplicity server includes an OPC server and an SQL database. In addition, an analysis component can be included into process machine for calculating different statistics about operator performance (Pietilä and Haavisto, 2010 and Pietilä et al. 2011). The operator VM includes only the Cimplicity Viewer software for accessing process screens. The teacher software for managing the simulations and training sessions is in the fourth virtual machine. The teacher can control one or several simulation environments, and use Cimplicity Viewer to view the screens of the simulations of the students.

The division into VMs was done to ensure equal computer settings for the automation software and communication links, and to make the parts easy to replicate and distribute.

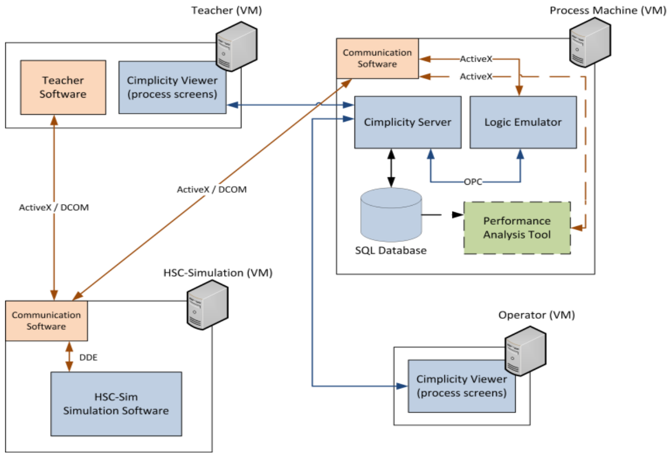

Fig. 2. Structure and communications in the training system.

3.2 Communications

The communications between the different parts of the training environment have been realized with ActiveX, Dynamic Data Exchange (DDE) and Object Linking and Embedding for Process Control (OPC). As shown above in Fig. 2, the teacher software communicates with the HSC Simulation VM through special communication software. The teacher software connects to the HSC Simulation VM with ActiveX over Distributed Component Object Model (DCOM) and starts remotely the communication software. In a similar way, the teacher software starts the communication software also in the process machine. The three parts in the communication software form a network in which data and commands are transferred. DCOM allows the environment to be established over TCP/IP between virtual machines in different locations; only an internet connection is required. The connection to the Cimplicity server, which provides data to the process screens, can also be established over TCP/IP.

Inside the virtual machines more communication links are needed. The communication software in process machine starts and communicates with the logic emulator using ActiveX. Similar procedure can also be used with the performance analysis tool. The logic emulator communicates with the Cimplicity software with OPC, which allows the logic to almost instantly receive an event when an operator changes any value. Also, large amounts of data can be efficiently updated from the logic to the automation software using OPC. In the HSC-Simulation VM the communication software uses DDE to communicate with the HSC software.

Generally, the process machine and the HSC-Simulation VMs are located within the same subnet to allow fast connection for data transfer. Students and teachers can then access the simulation network from anywhere using the operator or teacher VM on their own computers. It is also possible to have all the virtual machines running on a server computer and have students and teachers connect to the virtual machines using Remote Desktop Protocol from around the world. This way, no software needs to be installed on the computers of the students or teachers.

3.3 Integration to the automation software / Logic emulator

Once the simulation engine was working, one of the biggest remaining challenges was the logic emulation. In order to get the flotation process to run, a large number of I/O, calculated variables, unit controllers, interlocks, etc. had to be modelled. The challenge was solved by programming a separate component that carries out all the same tasks as the real logic controllers in the plant. The logic emulator was programmed with Matlab® so that the actual configuration files from the process logic (Proscon® Configuration Manager) could be used in configuration of the logic. The files can be saved in multiple sheet Excel format, and contain most information needed to configure the logic. The functions and relationships of the different variables in the process logic are programmed into another spreadsheet file. Based on these two files, the logic emulator generates its database of points (i.e. variables) and their internal relationships. The database can also be edited manually, should there be any need to configure certain points differently. Furthermore, the structure of the logic emulator allows the environment to be easily adapted to new processes with the same automation system.

So far, the generation of the static files that comprise the logic database has been discussed. The following explains how the logic emulator works when a new simulation is started. Before the first run, the logic emulator connects to the Cimplicity OPC server (see Fig. 2) and verifies that all the points defined in the logic exist also in the automation software. Then, it iterates an optimal execution order for the points based on their dependencies. This is to ensure that the point values are updated in correct order so that the latest information is used when calculating new values (e.g. when a point value depends on one or more other point values).

When the simulator completes its calculation round, all the information coming from it is processed by the logic emulator, i.e. there is no direct connection from the automation system to the process or vice versa (this is the case also in actual process; everything goes through logic). The logic emulator sets all the correct automation software variables for the operator to see on the screens. Also the control outputs for PID and ratio controllers are calculated based on the measured values received from the simulator. All the actions of the operators are first processed by the logic to find out, for example, possible interlock situations or to open valves that need to be opened when a motor is started. After the whole logic calculation round, the information is finally passed to the simulator. A typical operator screen in the Cimplicity automation software is illustrated in Fig. 3, showing e.g. an air flow PID controller.

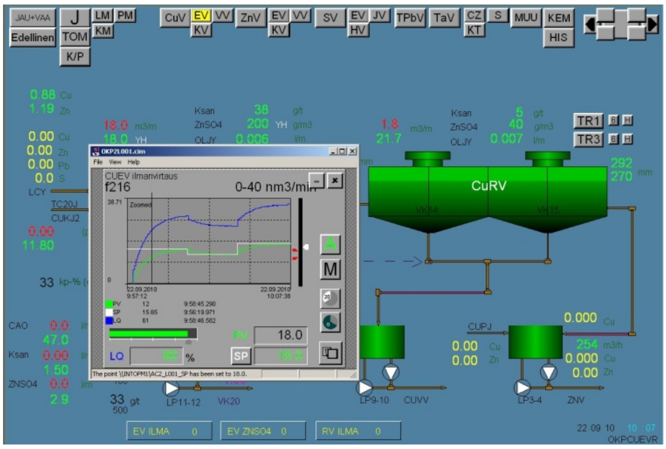

Fig. 3. Typical operator screen in the Cimplicity automation software (courtesy of Pyhäsalmi Mine Oy).

The logic emulator works with close cooperation with the simulation models in HSC. It receives simulation time in the communication message with the data during every round. Thus, simulation speed can be altered during simulation. Also, a fast forward command is going to be implemented to quickly simulate into steady state situation with the current model parameters. If (in the future) the simulator is run in parallel with the real process, this would allow the operator to quickly see the effect of the different control choices and make his final decision based on the simulated outcomes.

Outotec Oyj has also integrated the simulator into Siemens automation system to train operators at plants using it. This was more straightforward as a soft logic option was already present in the system and could be used with the simulator.

4. USAGE SCENARIOS

As already mentioned, there are many different usage scenarios for the training environment and the simulator. The main use is classroom training, but the simulator could also be used in parallel to the process by recursively updating the model with the real measurement available in the automation system. Thus, the system would be able to predict process responses for different operating actions. This requires, however, that a number of model parameters must be changed from static to dynamic (see Section 2.4).

The classroom training can be carried out in traditional manner, by selecting a common starting point and letting the trainees control the process as well as they can. The control goals are typically set by the experience level of the trainees; they may be just to keep the level of a single flotation cell or to control the entire flotation circuit. As the typical flotation process is relatively slow, the simulation environment allows the simulation speed to be increased (while maintaining continuous simulation) and there are also fast forward and rewind capabilities (jump between operating points).

Another teaching method is to let the trainees carry out pre-recorded process time series tasks that could happen (or have actually happened) in the process. An example of one such task could be a gradually descending input concentration requiring operator action to keep the process performing in an efficient manner. The tasks can reflect the skill level of the group and they can be used in teaching e.g. process specific events that are important to be handled correctly. Furthermore, the consistent set of tasks that are run by every student make it possible to use automated skill evaluation routines as depicted by Performance Analysis Tool in Fig. 2.

If needed, the classroom scenarios can be run with a rather coarse model since the accurate response is not as important as the correct look and feel of the system. Of course, the responses must be in the correct direction and of reasonable magnitude. This makes it possible to use generic process models at least for the generic classroom teaching.

A more advanced option, as done in this study, is to model the actual flotation process as accurately as possible. Of course, this requires much more work in the modelling phase, but it also brings interesting possibilities. For example, if the model is recursively kept up to date with the process, it is possible to run the simulation model in parallel with the process and use the simulator in prognosis, as illustrated in Fig. 4. In the example, the process operator is indecisive between two possible control actions and the simulator is used to check the response of the selected variables in advance. Note that the simulator is able to calculate also variables that are difficult to measure directly from the process, or it can substitute measurements for broken sensors. Also, as the simulation would be running concurrently with the process, it would be possible to construct tools for recording interesting operating points or sequences to be used in classroom training. They could be e.g. abnormal situations or those where the operator has managed exemplarily.

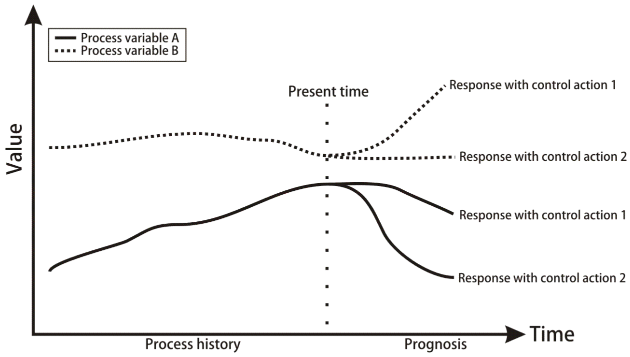

Fig. 4. An illustration of the prognosis option that is possible with an accurate simulation model.

ACKNOWLEDGEMENTS

This study has been financially supported by the Finnish Funding Agency for Technology and Innovation, Pyhäsalmi Mine Oy and Outokumpu Technology Minerals Oy. The research has been carried out in close collaboration with the aforementioned companies and the authors are indebted to many collaborators who have contributed to the work, especially to Mr Jarmo Huuskonen from Pyhäsalmi mine for his help on ins and outs of the Cimplicity/Proscon system.

REFERENCES

Brochot, S., Villeneuve, J., Guillaneau, Durance M.V., and Bourgeois, F. (2002). USIM PAC 3.0: Design and optimization of mineral processing plants from crushing to refining. In Mular, A.L, Halbe, D.N. and Barratt, D.J. *Mineral Processing Plant Design, Practice and Control*, 477-494. SME, Colorado, USA.

Desbiens A., Flament, F., and Pomerleau, A. (1997). Distributed Control at the Kidd Creek Grinding Plant – Part II: Implementation. *CIM Bulletin*, 90 (1008), 145-150.

Flament, F., Desbiens, A., del Villar, R., and Jakelski, D. (1997). Analysis and implementation of a new control strategy for the Kidd Creek grinding circuit, Part I. *CIM Bulletin*, 90 (1008), 139-144.

Finch, J.A., and Dobby, G. (1990). *Column Flotation*. Pergamon Press, Oxford, United Kingdom.

Gorain, B.K. (1997). *The effect of bubble surface area flux on the kinetics of flotation and its relevance to scale-up*. Ph.D. Thesis, University of Queensland, Australia.

Gorain, B.K. (1999). Prediction of bubble surface area flux in mechanical flotation cells from cell design and operating data. In *AMIRA P9L Project, FINAL REPORT*. JKMRC, Univesity of Queensland, Minerals Processing Research Group, Queensland, Australia.

King, R.P. (2001). *Modeling and simulation of mineral processing systems.* Butterworth-Heinemann, Oxford, United Kingdom.

Lamberg, P., and Bernal, L. (2009). Modeling and Simulation of SkimAir® Flash Flotation as a Part of Grinding-Flotation Circuit – Case Study in Esperanza, Chile. In *Proceedings of the 41st Annual Canadian Mineral Processors Conference* (CMP 2009), 30-38. Ottawa, Canada.

Lamberg, P., Paloranta, S., Aaltonen, A., and Myllykangas, H. (2009). A property based model of flotation and application in dynamic simulator for training purposes. In *Proceedings of the Conference of Metallurgists* (COM 2009), Ontario, Canada.

McKee, D.J., and Napier-Munn, T.J. (1990). The status of comminution simulation in Australia. *Minerals Engineering*, 3 (1-2), 7-21.

Miettinen, T., Ralston, J., and Fornasiero, D. (2010). The limits of fine particle flotation. *Minerals Engineering*, 23 (5), 420-437.

Moilanen, J., and Lamberg, P. (2010). Virtual Experience for Operator Training. In *Proceedings of the 7th International Mineral Processing Seminar* (PROCEMIN 2010), Santiago, Chile, in press.

Neethling, S.J., and Cilliers, J.J. (2002a). The entrainment of gangue into a flotation froth. *Int. J. Mineral Processing*, 64 (2-3), 123-134.

Neethling, S.J., and Cilliers, J.J., (2002b). Solid Motion in Foams. *Chemical Engineering Science*, 57, 607-615.

Neethling, S.J., Lee, H.T., and Cilliers, J.J. (2003). Simple Relationships for Predicting the Recovery of Liquid from Flowing Foams and Froths. *Minerals Engineering*, 16 (11), 1123-1130.

Neethling, S.J. (2008). Simple Approximations for Estimating Froth Recovery. *Int. J. Mineral Processing*, 89 (1-4), 44-52.

Neethling, S.J., and Cilliers, J.J., (2009). The entrainment factor in froth flotation: Model for particle size and other operating parameter effects. *Int. J. Mineral Processing*, 93 (2), 141-148.

Outotec, (2006). *HSC Chemistry – Chemical reaction and equilibrium software with thermochemical database and simulation module*, ISBN 978-952-9507-12-2, Pori, Finland.

Paloranta, S. (2008). *Dynamic model and training simulator for flotation process*, M.Sc. Thesis. Helsinki University of Technology, Finland.

Pietilä J., and Haavisto O. (2010). Comparison of operator performance in a mineral processing plant, In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* (SMC 2010), October 10-13, Istanbul, Turkey.

Pietilä J., Remes A., and Haavisto O. (2011). Process Operator Skill Exploration by a Data Analysis Software, 18th IFAC World Congress (IFAC 2011), August 28 - September 2, Milan, Italy, submitted.

Roine, A., and Kotiranta, T. (2007). Development of Sustainable Processes with new HSC Chemistry 6.0 Software. In *Proceedings of European Metallurgical Conference,* 1877-1888. Dusseldorf, Germany.

Runge, K.C., Frew, J., Harris, M.C., and Manlapig, E.V. (1997). Floatability of streams around the Cominco Red Dog Lead Cleaning Circuit. In *Proceedings of the AusIMM 6th Mill Operators Conference*, 157-163. Australasian Institute of Mining and Metallurgy, Madang, Papua New Guinea.

Savassi, O.N., Alexander, D.A., Franzidis, J.P., and Manlapig, E.V. (1998). An empirical model for entrainment in industrial flotation plants. *Minerals Engineering*, 11 (3), 243-256.

Savassi, O.N. (1998). *Direct estimation of the degree of entrainment and the froth recovery of attached particles in industrial flotation cells*, Ph.D. Thesis. University of Queensland.

Schwarz, S., and Alexander, D. (2006). JKSimFloat V6.1 Plus: Improving flotation circuit performance by simulation. In Yalcin, T. and Shang, H. (ed.), *Mineral Process Modelling, Simulation and Control - Conference Proceedings*, 35-48. Laurentian University, Sudbury, Canada.

Smith, J., and Nair, R. (2005). *Virtual Machines: Versatile Platforms for Systems and Processes.* Morgan Kaufmann, San Francisco, USA.

Vera, M. (1999). Calculation of froth recovery by varying froth depth. In *AMIRA P9L Project, FINAL REPORT.* JKMRC, University of Queensland, Minerals Processing Research Group, Queensland, Australia.

Welsby, S.D.D., Vianna, S.M.S.M., and Franzidis, J.-P. (2010). A continuous pilot-scale flotation rig for the systematic study of flotation variables. *Minerals Engineering*, 23 (10), 826-829.

Wills, B.A., and Napier-Munn, T.J. (2006). *Wills' Mineral Processing Technology*, Seventh Edition. Butterworth-Heinemann, Oxford, United Kingdom.

Zheng, X., Franzidis, J.P., and Johansson, N.W. (2006). An evaluation of different models of water recovery in flotation. *Minerals Engineering*, 19 (9), 871-882.