

MATLAB Real-time Data Acquisition for Laboratory Froth Flotation Automation

Ivan Brezáni¹ and Fridrich Zeleňák²

In this paper analog input/output signals are used to control laboratory froth flotation machine using industrial panel PC. Data acquisition is provided by two Advantech PCI cards, while flotation control is based on connection of adjustable frequency drive further connected to AC motor for agitation control, mass flow controller for air flow control, pH and temperature transmitters with according probes for pH and temperature logging respectively. Process control is provided by graphical user interface programmed in MATLAB.

This practical engineering design allows researchers to set parameter values prior to flotation test as well as log and save measured values for further processing.

Keywords: froth flotation, automation, data acquisition, signal processing, MATLAB

1 Introduction

Froth flotation is a separation method widely used in mineral processing because of its universal utilization. It can be used in primary as well as secondary raw materials processing, biotechnologies and waste water cleaning, allows separation of complex ores with similar physicochemical properties and all of this in size ranges allowing good mineral liberation, where most of the other physical separation methods has limited or no efficiency. Froth flotation uses difference in surface properties of raw materials represented by floatability, wettability, natural or chemically altered hydrophilicity and hydrophobicity. Separation takes place on three phase contact of water, air and mineral particle surface where hydrophobic particles of valuable mineral inside of an agitated and aerated pulp tend to form stable connection with air bubbles and report to overflow - froth product, while hydrophilic particles of gangue sink and report to tailings. This simplified description of froth flotation separation process is depicted in Fig. 1.

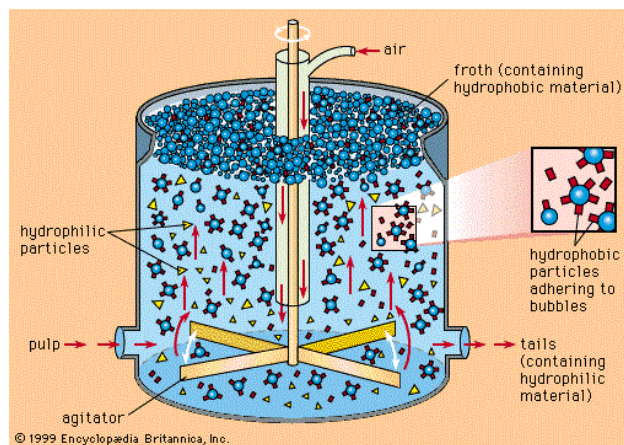


Fig. 1. Schematic diagram of a continual froth flotation separation cell [1].

Although the above description may sound simple, this complex process is not yet fully understood and none of the many models proposed over the decades to describe flotation phenomena and to predict results of froth flotation separation can be used as a single and universal one [2].

¹ Ing. Ivan Brezáni, Technical University of Košice, BERG Faculty, Institute of montanneous sciences and environmental protection, Park Komenského 19, 043 84 Košice, ivan.brezani@tuke.sk

² doc. Ing. Fridrich Zeleňák, PhD., Technical University of Košice, BERG Faculty, Institute of montanneous sciences and environmental protection, Park Komenského 19, 043 84 Košice, fridrich.zelenak@tuke.sk

As it has been already noted, froth flotation process is very complex and is affected by following parameters and properties:

- slurry properties (density, solids content),
- slurry flow rate (retention time),
- hydrodynamic properties inside of a cell,
- electrochemical parameters / potentials (pH, Eh, conductivity),
- chemical reagents and their addition rate (frothers, collectors, depressants, activators),
- pulp levels in cells,
- air flow rates into cells,
- froth properties (bubble size distribution, froth stability),
- particle properties (size distribution, shape, degree of mineral liberation),
- mineralogical composition of the ore,
- mineral concentrations in the feed, concentrate and tailings (recovery, grade) [3, 4].

In laboratory conditions, flotation test are commonly carried out as batch tests as opposed to continuous flotation process in industrial applications. This simplifies the tests and their evaluation, however makes it more difficult to predict industrial flotation performance based on laboratory test. Mineral processing simulation and modelling software such as MODSIM and USIM PAC can be used, allowing user to evaluate the effect of different flowsheet designs including cleaning and scavenging operations as well as grinding and regrinding circuits. Simplified models - first order and second order kinetic models are used as they are comprehensive enough to address many industrial problems, yet simple enough to be used with the data available from laboratory batch tests [5].

These models described i.e. in [6] require the dependency of valuable and gangue mineral recovery over time as a basis for parameter estimation. Different models are used in continuous and batch flotation conditions as can be illustrated on classical first order kinetic flotation model with its batch form shown as Eq. 1 and continual form shown as Eq. 2 [7].

$$R = R_{\infty} \left(1 - e^{-k\tau} \right) \quad (1)$$

$$R = R_{\infty} \left(1 - \frac{1}{1 + k\tau} \right) \quad (2)$$

Where: R - mineral recovery [-], R_{∞} - maximum possible recovery (at infinite flotation time) [-], k - flotation rate constant [1/min], τ - flotation time [min].

This means that Eq. 1 can be used to estimate the value of parameters R_{∞} and k for valuable mineral and gangue based on experimental laboratory data and can then be used to predict recovery of these components at different retention times in industrial flotation cells using Eq. 2. In order to obtain results with good reproducibility, parameters such as air flow or agitation speed affecting hydrodynamic conditions inside of a cell must be kept constant throughout a single test, but it is advised to have ability to change these individual parameters in order to find optimum set for each type of raw material.

Common laboratory flotation machines only allow users to manually adjust agitation speed and adjust air flow rate after the conditioning period (reagents are added prior to flotation test). Modernization and partial automation of these machines can be done by equipping them with new components that can be externally switched on/off and adjusted using PC. Connection with computer also allows logging of various parameters, such as pH and temperature along with agitation speed and air flow rate. Such a system was developed to simplify laboratory froth flotation tests and will be discussed further in this article.

MATLAB as high level technical computing language allows users to perform computational tasks easier and faster than traditional programming languages such as C, C++, and FORTRAN. MATLAB as a software product is a set of various toolboxes that extend its use for virtually any application. MATLAB was chosen over other similar products because of authors prior experiences with this product.

2 System description

2.1 Device description

DV-2 laboratory flotation machine (see Fig. 2) was used as a skeleton. Previous AC motor was replaced by inverter-duty capable one and connected to adjustable inverter. Air mass flow meter and controller was installed prior to machines air inlet. pH controller and temperature transmitter each with analog output were installed.

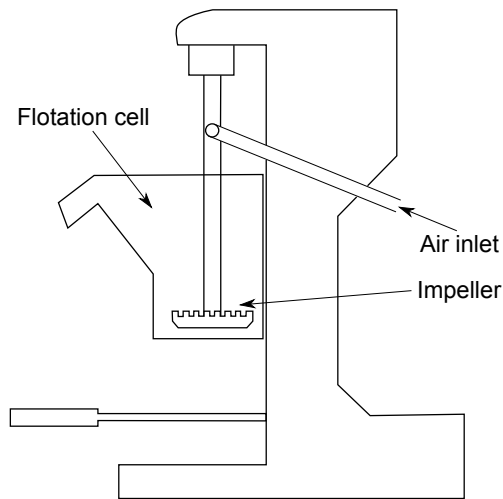


Fig. 2. DV-2 Laboratory flotation machine [8].

Agitation Model: MVXF50A0-2 vector adjustable drive
 Accuracy: $\pm 3 \text{ rev.min}^{-1}$
 Analog input: 0-10V or 4-20mA
 Analog output: 0-10V
 Motor: 0.5hp, $1725 \text{ rev.min}^{-1}$

Mass flow Model: FMA-A2409-SS mass flow controller
 Air flow: $0-10 \text{ dm}^3 \cdot \text{min}^{-1}$
 Accuracy: $\pm 0.1 \text{ dm}^3 \cdot \text{min}^{-1}$
 Analog input: 0-5V
 Analog output: 0-5V or 4-20mA

pH Model: PHP-701 pH controller with pump
 Accuracy: $\pm 0.01\text{pH}$
 Analog output: 4-20mA
 Electrode: PHE-1304, 0-12pH

Temperature Model: TXDIN70 transmitter
 Accuracy: $\pm 0.15^\circ\text{C}$
 Analog output: 4-20mA
 Sensor: PRCU-10-2-100-3 fast response RTD

2.2 Data acquisition

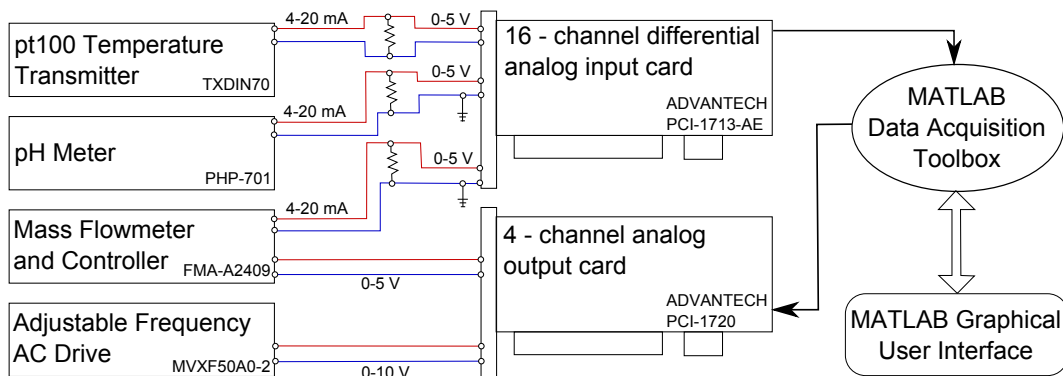


Fig. 3. Schematic diagram of device connections and control.

15" touch screen industrial panel PC, model IPPC-7157A equipped with two PCI data acquisition cards running Matlab R2010a under Windows XP Pro Embed SP3 was used for data acquisition and flotation control. Final connection of individual devices can be seen on Fig 3. 4-20mA current loop signal from individual devices was converted to 0.96-4.8V signal using precision 240 Ω resistor. Alternatively 250 Ω resistor can be used to convert the signal to 1-5V.

Conversion of 4-20mA analog output signal to voltage was preferred over direct voltage output, as this allows indication of loop disconnection when measured current is below 4mA or converted signal is below 0.96V respectively. From Fig 3 it is also evident that agitation speed is not measured, as the variation of agitation speed is only about $\pm 2 \text{ rev.min}^{-1}$ and the drive was set to achieve final speed in about 2 seconds.

Input Model: PCI-1713-AE

Accuracy: $\pm 0.04\%$

Channels: 32 single-ended or 16 differential input

Range: Bipolar: $\pm 10\text{V}$, $\pm 5\text{V}$, $\pm 2.5\text{V}$, $\pm 1.25\text{V}$, $\pm 0.625\text{V}$

Unipolar: 0-10V, 0-5V, 0-2.5V, 0-1.25V

Output Model: PCI-1720U-AE

Accuracy: $\pm 0.024\%$

Channels: 4x isolated analog output

Range: Unipolar: $0\pm 5\text{V}$, $0\pm 10\text{V}$

Bipolar: $\pm 5\text{V}$, $\pm 10\text{V}$

3 Graphical User Interface

Graphical user interface (GUI) (see Fig. 4) was programmed in Matlab R2010a in order to simplify parameter setting as well as overall machine control. It can be vertically divided into three sectors, where on the left side currently read values are shown, in the middle section there are plots with values measured for each second of flotation and on the right side, overall flotation time, remaining time in current flotation period as well as different control buttons can be found. After flotation test is over, the left section shows average parameter values for overall flotation time (excluding conditioning period). Horizontally there are four sections - one for each parameter (from top to bottom - pH, air flow, temperature and agitation speed).



Fig. 4. Graphical user interface during a flotation test.

3.1 Source code

Only few relevant pieces of source code will be described in this article, as there are over 1500 lines of code in the main function. Source code in the form of Matlab m-files is freely available for download under the BSD License at MATLAB Central File Exchange webpage [9].

Matlab Data Acquisition Toolbox was used to communicate with analog input PCI card (advantech device "0") and analog output PCI card (advantech device "1"), where the `daqwinfo` function was used for identification of the two PCI cards. Default input/output channels are defined in main function preamble as follows:

```

1  % Define hardware channels of installed DAQ cards
2  handles.ai_card = 0; % 'advantech' PCI-1713
3  handles.ao_card = 1; % 'advantech' PCI-1720
4  % Define analog input channels
5  handles.pH_ai_ch = 0; % Default channel for reading pH
6  handles.lpm_ai_ch = 1; % Default channel for reading air flow
7  handles.temp_ai_ch = 2; % Default channel for reading temperature
8  % Define analog output channels
9  handles.rpm_ao_ch = 0; % Default channel for setting air flow
10 handles.lpm_ao_ch = 1; % Default channel for setting stirring speed

```

When flotation test starts, three input channels and two output channels are automatically created. Input channels are set to measure voltage in the range of 0 – 5V. Output channel for agitation speed control is set to send voltage in the range of 0 – 10V and output channel for air mass flow control is set to produce 0 – 5V output.

```

1  % Create output channels
2  handles.ao = analogoutput('advantech', handles.ao_card);
3  addchannel(handles.ao, handles.rpm_ao_ch);
4  addchannel(handles.ao, handles.lpm_ao_ch);
5  set(handles.ao.Channel(1), 'OutputRange', [0 10]);
6  set(handles.ao.Channel(2), 'OutputRange', [0 5]);
7  % Create input channels
8  handles.ai = analoginput('advantech', handles.ai_card);
9  addchannel(handles.ai, handles.pH_ai_ch);
10 set(handles.ai.Channel(1), 'InputRange', [0 5]);
11 addchannel(handles.ai, handles.lpm_ai_ch);
12 set(handles.ai.Channel(2), 'InputRange', [0 5]);
13 addchannel(handles.ai, handles.temp_ai_ch);
14 set(handles.ai.Channel(3), 'InputRange', [0 5]);

```

Sample rate of input signal is set to 10Hz, which ensures measurement of ten voltage values each second. Flotation time is calculated as a sum of individual used period times (see Sec. 3.3), while the length of the flotation test is determined by the number of measurement to take, which is the flotation time multiplied by ten.

```

1  set(handles.ai, 'SampleRate', 10);
2  flot_time = sum(handles.period_time.*handles.period_use);
3  set(handles.ai, 'SamplesPerTrigger', 10*flot_time);

```

Voltage measurement on analog input card is started using the `start` command, while process control using analog output card uses the `putsample` command. `putsample` command needs two arguments, where the later one is a vector consisting of voltages to produce, where `V` is a calculated voltage for controlling agitation speed and `V2` value controls air mass flow.

```

1  start(handles.ai);
2  putsample(handles.ao, [V V2]);

```

As it is not possible to access measured voltages using standard `getdata` command until the number of samples specified in the `SamplesPerTrigger` property of analog input object are taken, `peekdata` command is used to access last ten measurement for each second of flotation.

```

1  data = peekdata(handles.ai, 10);

```

Average voltage is calculated as a mean of ten measurements for air mass flow and temperature. As there was some fluctuation (noise) in read pH voltages, filtering was implemented as a simple rejection of voltage values outside of the mean voltage ± 1.15 times calculated standard deviation. Conversion of individual voltage values to pH, air flow and temperature is described in Sec. 3.2. The same principle of mean voltage and following conversion is also used at the end of the flotation test to prepare data for saving to external file.

```

1  % Calculate mean and standard deviation of pH value
2  stdev_pH = std(data(1:10,1));
3  mean_pH = mean(data(1:10,1));
4  % Remove outsider values
5  for J = 1:10
6      if data(J,1) > mean_pH + 1.15*stdev_pH
7          data(J,1) = 0;
8      elseif data(J,1) < mean_pH - 1.15*stdev_pH
9          data(J,1) = 0;
10     end
11 end
12 % Calculate average voltage values
13 data(1,1) = sum(data(1:10,1))/(10-sum(data(1:10,1)==0));
14 data(1,2:3) = sum(data(1:10,2:3))/length(data(1:10,2:3));

```

After flotation test is over and number of measurement specified in the SamplesPerTrigger property are taken input and output channels are deleted and data is cleared from buffer and memory.

```

1  delete(handles.ao);
2  clear handles.ao;
3  delete(handles.ai);
4  clear handles.ai;

```

3.2 Voltage conversion and calibration

Parameter values are calculated from measured voltage using Eq. 3 (shown as example for pH value):

$$pH_{val} = pH_{min} + \left(pH_{volt} - 4 \frac{pH_{\Omega}}{1000} \right) \frac{pH_{max} - pH_{min}}{20 \frac{pH_{\Omega}}{1000} - 4 \frac{pH_{\Omega}}{1000}} \quad (3)$$

where pH_{val} stands for actual (calculated) pH value, pH_{volt} is the analog voltage (converted current loop signal) read by the analog input card, pH_{min} and pH_{max} stands for pH value of 4mA or 20mA transmitter output respectively and pH_{Ω} is the resistance of resistor used.

Use of Eq. 3 allows full calibration of measured signals as can be seen on Fig 5. Configuration can be saved as *.conf* file and loaded after the GUI was restarted (it is not possible to store configuration settings from last session).

Air flow and agitation speed can be set along with minimum and maximum values to be shown on the graphs, while pH and temperature is not adjusted and therefore only minimum and maximum values to be shown on graphs can be set.

3.3 Flotation periods

As it has been already noted, in order to estimate flotation kinetics constants, dependency of mineral recovery over flotation time is needed. Therefore it is advised to divide overall flotation time into several time periods in which froth product is sampled separately. This can be done in the GUI (see Fig. 6) along with optional conditioning period with agitation but no air flow. Although remaining period time is shown in GUI, sound indication warns user before each period end.

Flotation times in seconds are stored in the *period_time* vector. Up to ten flotation periods can be specified plus the conditioning period. The decision whether to calculate with period is stored in the *period_use* vector consisting of values 0 and 1 for the case when period is not used and is used respectively.

After flotation test is over, parameters measured for each second of flotation can be saved as text *.txt* or Microsoft Office *.xls* file for further processing or archivation.

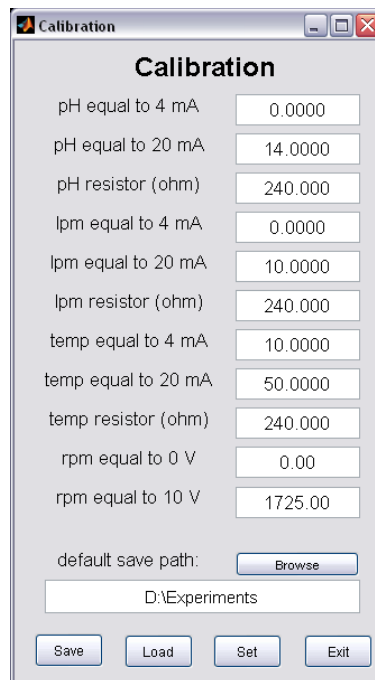


Fig. 5. Window for device calibration.

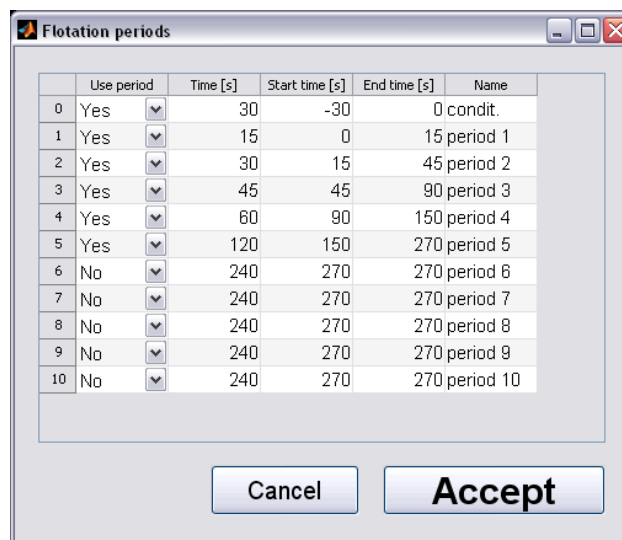


Fig. 6. Window for flotation periods setting.

4 Conclusion

Option for common laboratory froth flotation machine automation using analog signal input/output devices and PC data acquisition is here proposed. Graphical user interface was programmed in Matlab and made available for free download. This system can be further extended and modernized by connection of other measuring devices, as i.e. camera evaluating various other parameters - pulp levels, froth stability, bubble size distribution or device for continuous measurement of slurry density, conductivity, etc.

Source code can be used without alternation also with other pH and temperature transmitters with 4-20mA or 0-5V analog output, adjustable frequency drives with 0-10V input and air flow controllers with 4-20mA or 0-5V analog output and 0-5V analog input. Researchers are therefore not limited by use of the exact models as mentioned in Sec. 2. Minor change in source code will allow user to connect virtually any other analog input/output measuring or controlling device.

Acknowledgement

This work was supported by the Scientific Grant Agency (VEGA) of the Ministry of Education of the Slovak Republic and the Slovak Academy of Sciences, Grant No. 1/0752/10.

References

- [1] Flotation separation cell. Art. Encyclopedia Britannica Online. Web. 19 Sep. 2011.
<http://www.britannica.com/EBchecked/media/1524/Schematic-diagram-of-a-flotation-separation-cell>.
- [2] J.B. Yianatos, Fluid Flow and Kinetic Modelling in Flotation Related Processes: Columns and Mechanically Agitated Cells - A Review, Chemical Engineering Research and Design, Volume 85, Issue 12, pp. 1591-1603 (2007).
- [3] H. Laurila, J. Karesvuori and O. Tiili, Strategies for instrumentation and control of flotation circuits, Mineral Processing Plant Design, Practice and Control, Volume 1, pp. 2174-2195 (2002).
- [4] B.J. Shean, J.J. Cilliers, A review of froth flotation control, International Journal of Mineral Processing, Volume 100, Issues 3-4, pp. 57-71 (2011).
- [5] J. Villeneuve, J.-C. Guillaneau, M.-V. Durance, Flotation modelling: A wide range of solutions for solving industrial problems, Minerals Engineering, Volume 8, Issues 4-5, pp. 409-420 (1995).
- [6] A. Gupta, D. S. Yan, Flotation, Mineral Processing Design and Operation. Elsevier Science, Amsterdam, pp. 555-603 (2006).
- [7] Caspeo, USIM PAC Unit Operation Model Guide, Technical Manual, Version 3.1.1.0, pp. 195-196 (2006).
- [8] M. Sisol, Flotation separation of plastics (in Slovak), Dissertation thesis, Technical University of Košice, (2006).
- [9] I. Brezáni, Laboratory froth flotation machine automation GUI, Available online: <http://www.math-works.com/matlabcentral/fileexchange/32985>, (2011)