

IMPLEMENTATION OF A WORKFLOW MODEL TO STORE AND ANALYZE MEASURED DATA AT THE ESS-BILBAO ION SOURCE TEST STAND

Z. Izaola*, M. del Campo, M. Eguiraun, I. Bustinduy (ESS-Bilbao, Bilbao, Spain)

Abstract

In order to fully characterize the experimentally measured beam in any accelerator facility, both diagnostics measurements and operating parameters need to be stored and correlated. Generating thus, a substantial amount of data. To address this problem in the ESS-Bilbao Ion Source Test Stand (ITUR), we have developed a software toolkit. This software stores Pepperpot, Faraday-Cup, Retarding Potential Analyzer, ACCT and DCCT measurements in a relational database associated with the operating parameter values at the time of measurement. Furthermore, the toolkit stores in the same database the beam transverse dynamics parameters processed from the pepperpot device. This allows to connect easily the beam physics with the accelerator running parameters. MySQL[®] has been used as database back end and Matlab[®] as programming language.

INTRODUCTION

The rationale behind the ITUR project [1] is to perform a comparison between different kinds of hydrogen ion sources using the same beam diagnostics setup. In particular, a direct comparison will be made in terms of the emittance characteristics of Penning-type sources such as those currently being used in ISIS (UK) and those of microwave type such as CEA-Saclay and INFN.

To fully characterize the performance of an ion source it is essential to correlate the measurement of different beam diagnostic instruments with the operating parameters of the ion source. It is very important to store as many operating parameters and diagnostic data as possible while the ion source is running to, analyze any possible problem. The data of some diagnostic instrument, *e.g.* transverse phase space measured by a pepperpot, need a careful post-processing; and thus, its result can not be stored on real time with the rest of the data.

The aim of the software presented in this work is to store the output of the pepperpot data analysis in a relational database with the relevant operating parameters and diagnostic instrumentation measurements. This objective will allow to characterize the beam distribution of the ion source at ITUR for different operating parameters.

DATA COLLECTED AT ITUR

The main operating parameter of a Penning-type source are the platform, plasma and extraction voltages (their

working duty factor and reference and actual waveform), the Cs boiler temperature, the H₂ flow rate and the pressure in the vessel. Any change on one of these parameters may lead to different ion distribution in the extracted beam.

For diagnostic purposes ITUR is equipped with a Pepperpot, a Faraday-Cup, a Retarding Potential Analyzer (RPA), an AC and DC current transformer (ACCT and DCCT). At the moment, the RPA is being reworked for improvements and therefore it is not used.

The integrated current measurement of the ACCT and DCCT, and the actual voltage and current waveforms of the various power supplies are stored as arrays of floating values, while the rest of the parameter are scalar variables of double floating type. The size of the arrays is 8000 points for the ACCT and 4000 points for the DCCT and the Faraday Cup. This difference is due to the used sampling rate (200 kHz and 100 kHz respectively) which is limited by the sensors' bandwidth. Two consecutive working periods are sampled. As a result the waveform is 40 ms long. At the present time scalar parameters are stored every 5 seconds, while array values are stored every 5 minutes. The total amount of data is close to 10 MB/hour.

The control system is based on LabVIEW[®] Real Time, installed on a National Instruments' embedded controller equipped with a 2.53 GHz Intel Core 2 Duo T9400 dual-core processor. The system is managed from a Windows[®] 7 PC running a LabVIEW[®] program that saves the data directly on the MySQL[®] database.

THE PEPPERPOT

The pepperpot [2] consists of 2 main elements: an intercepting head and a high speed CCD camera. The head consists basically of a 0.5 mm tungsten foil with a grid of $100 \pm 10 \mu\text{m}$ diameter holes, sandwiched between two copper plates with an identical array of 2 mm of diameter holes. The beamlets crossing the tungsten plate holes are intercepted by a 1 mm thick scintillating quartz screen. The hole grid is a square array of 41×41 holes, with a 9 ± 0.025 mm pitch, giving a total imaging area of $120 \times 120 \text{ mm}^2$.

The front copper plate absorbs a significant fraction of the incident beam and the middle plate provides the 10 mm drift length and prevents the beamlets from overlapping; both provide improved cooling to the tungsten screen. The scintillation light from the quartz plate is imaged with a PI-MAX:1K high speed camera with 1024×1024 pixels, a 16-bit monochrome CCD sensor and a Nikon Sigma 70 mm f/2.8 EX DG lens. The camera-to-screen distance is cur-

* zunbeltz.izaola@essbilbao.org

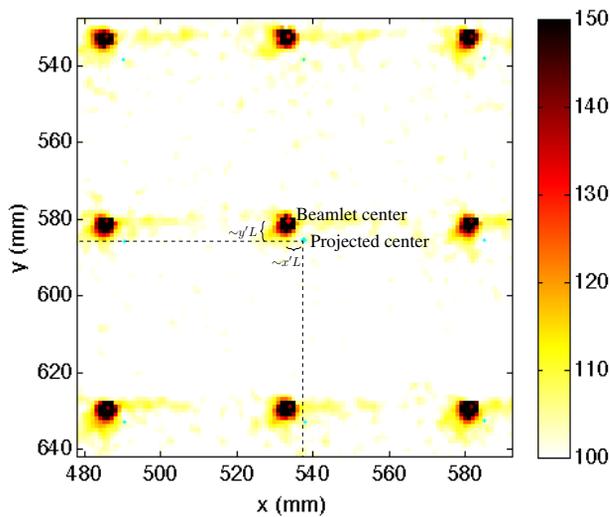


Figure 1: Center of the light spot produced by the beamlet and the projection of the apertures' center.

rently fixed at 110 mm.

At the moment the pepperpot is controlled from an external laptop via an usb port with its proprietary software.

PEPPERPOT DATA ANALYSIS

The data from the camera is written directly in a multi-image TIFF file and analyzed with Matlab[®]. The analysis software was developed within the FETS project at RAL and it has been generalized to positive ion beams and other pepperpot grid geometries. The pepperpot creates produces 1681 beamlets; one for each hole. From the positions and angle of the centers of each beamlet, the emittance is calculated as

$$\epsilon = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle xx' \rangle^2},$$

where averages are weighted with the number of particles, e.g. $\langle x^2 \rangle = \sum x^2 c(x, x') / \sum c(x, x')$.

The beamlet produces a light spot, on the scintillator, displaced from the aperture projection. The angle is calculated with this displacement and the distance, $L = 10$ mm, between the grid and the scintillator (see Figure 1). The projected hole positions are generated bearing in mind the calibration parameters, *i.e.* the possible offset and tilt of the pepperpot head with respect to the CCD sensor and the spot-to-spot distance in mm. These parameters are calculated finding the marks in the four corners of the last copper plate, in a calibration image taken with an external light source.

First, an approximated emittance distribution, *average calculations*, is calculated from the weighted center of each light spot. Then, from the x' and y' values of every single pixel with an intensity above a certain cut threshold, the proper emittance values, *single pixel calculations*, are produced. The total pixel intensity for each spot is summed to obtain the beam intensity information.

06 Beam Instrumentation and Feedback

T22 Reliability, Operability

The analysis calculates the emittances for both dimensions x and y , and the rms values of x , y , x' and y' (also the related velocities β_x and β_y) in the average and pixel approximations. The Twiss parameters in both phase-space planes are also derived from the single pixel calculations:

$$\alpha = -\langle xx' \rangle / \epsilon, \quad \beta = \langle x^2 \rangle / \epsilon, \quad \gamma = \langle x'^2 \rangle / \epsilon.$$

The software can also generate a particle distribution with the calculated beam characteristic to use it in particle dynamic tracking simulations, such as GPT, MAFIA and LINTRA.

SYNCHRONIZATION OF THE PEPPERPOT IMAGE WITH THE CONTROL SYSTEM

The present control of the pepperpot prevents us to trigger the photographs using the signal from the ion source. Therefore, the process is manual and the exact moment when the image was taken can not be determined. The only information available is the time stamp given to the image file by the operating system. This time stamp depends on the internal clock of the laptop and it only has a second scale accuracy.

To overcome this problem, a program running in the laptop is constantly communicating with the main control system. It writes, in a text file, the time on the laptop and the time when data from diagnostics was acquired. This allows to set the time stamp of the image file to the ones from the control system

WORKFLOW AND SOFTWARE DESCRIPTION

The designed workflow for the pepperpot measurements follows these steps:

1. Take an image using the pepperpot.
2. Store the image with the related ion source operating parameters and the diagnostic data.
3. Analyze the pepperpot image.
4. Store the calibration parameters and output data of the analysis.

The software presented in this work deals with steps 2 and 4.

Once the image has been taken (step 1), it must be transferred from the laptop of the pepperpot to the data analysis computer. A Matlab[®] function takes the image, as sole input; reads the image file time stamp and calculates its value in the control system reference time. Using this new time stamp, the function retrieves the operating parameters and diagnostic data from the main database. Because the control system's precision is higher than a mere second, the software preforms an average of the values available within the time stamp of the image. Then, this averaged data and the image are stored back in the pepperpot_measurement table (see Figure 2) of the relational database.

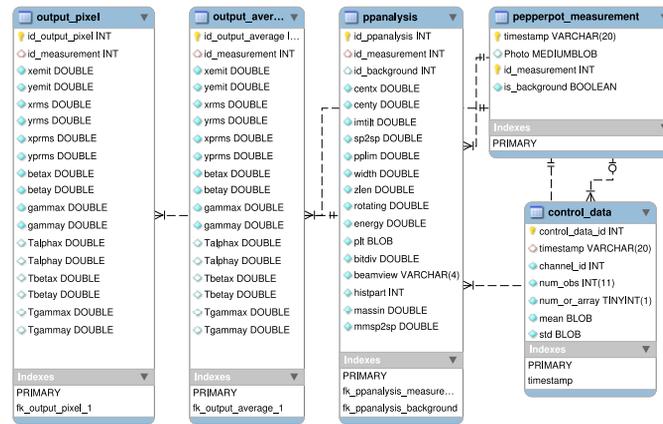


Figure 2: Database scheme for the tables that stores the pepperpot image (pepperpot_measurement), image analysis parameters (ppanalyse) and analysis results (output_pixel and output_average).

The next step is to analyze carefully the image as described before. A new calibration image should be used if any of the parameters of the pepperpot is changed.

Then, a second function with the image time stamp and calibration data as input parameters is used. This function runs the analysis program automatically and records the calculated quantities. The input of the analysis is stored in the table ppanalyse, the output of the average calculation in the output_average table and the single pixel calculations in the output_pixel table (see Figure 2).

Because all table are in the same database it is possible to relate all the calculation with the original data by means of the time stamp.

Implementation details

The software is written using Matlab[®]. The connection with the MySQL[®] database is done with the MyM package [3]. Altogether there are ~700 lines of code, from which ~250 are for testing proposes.

Using the unit test method, small units of source code (functions,..) are tested to determine if they are fit for use. The test are run automatically [4] using the xUnit Test Framework [5].

The source code is fully documented using the literate programming system matlabweb[6], which is based on CWEB [7].

FUTURE WORK

For the future, two main working areas are foreseen:

- The integration of the pepperpot within the general control system. To build a custom program to trigger the camera with the ion source timing signal will eradicate the synchronization problem. It will also make possible to automatically upload the image to the database.
- A faster and more reliable system using a new data

storage back end based on Hypertable [8], will be deployed [9]. The software should change accordingly.

CONCLUSIONS

The presented workflow will allow to manage the big amount of data involved in the characterization of ion sources. The pepperpot based transverse phase-space beam distribution can be easily related to the ion source operating parameters.

ACKNOWLEDGEMENTS

The authors would to thank S. Jolly for providing the image analyzing software. This work is supported by the Basque Government and Spanish Ministry of Science and Innovation.

REFERENCES

- [1] Ibon Bustinduy et al. First Simulation Tests for the Bilbao Accelerator Ion Source Test Stand. In *1st International Particle Accelerator Conference: IPAC'10, Kyoto, Japan, 2010*.
- [2] S. Jolly, D. Lee, J. Pozimski, P. Savage, D. Faircloth, and C. Gabor. Beam Diagnostics for the Front End Test Stand at RAL. In *Proceedings DIPAC07, 2007*.
- [3] Dimitri Shvorob. MySQL wrapper for MATLAB. <http://sourceforge.net/projects/mym/>.
- [4] S. Eddins. Automated Software Testing for MATLAB. *Computing in Science Engineering*, PP(99):1, 2010.
- [5] Steve Eddins. MATLAB xUnit Test Framework.
- [6] Mark Potse. <http://www.ctan.org/tex-archive/web/matlabweb>.
- [7] Donald E. Knuth and Silvio Levy. The CWEB system of structured documentation.
- [8] Hypertable. <http://hypertable.org/>.
- [9] Maria del Campo, Mauro Giacchini, Loris Giovannini, and Josu Jugo. EPICS HyperArchiver: initial tests at ESSBilbao. In *this proceedings (WEPC155)*, 2011.