

# PROCESSING AND VISUALIZATION OF EPICS DATA WITH MATLAB APPLICATIONS

E.Tikhomolov, TRIUMF, Vancouver, Canada

## Abstract

To conserve control system resources it is often desirable to run compute-intensive real-time data processing applications on a dedicated host computer. In the EPICS-based control system of the ISAC radioactive beam facility, the Extensible Display Manager tool (EDM) is used for the operator interface. EDM screens control data acquisition, processing and provide visualization of the processed data. Matlab is used as the data processing engine. A number of Matlab applications were created in collaboration with the beam physics group. These applications are running on a dedicated Linux host, using EPICS Matlab Channel Access (MCA) to obtain raw data from beam diagnostic IOCs (Linux-based) and store the processed results in the IOC. The raw data are provided to the IOC by fast data acquisition applications through a shared memory interface.

## INTRODUCTION

Matlab is very popular tool, which is used for processing and visualization of physics data. Interfacing it to EPICS initially was done at SLAC and then many authors contributed to its development [1]. Matlab applications, which use MCA, are able to get data from EPICS process variables (PV) and then process and visualize them. Matlab applications consume a lot of computer resources, thus it is better to run them on some dedicated computer to avoid interference with work done by operators and experimenters. In ISAC the distributed system consists of EPICS Input/Output Controllers (IOC) that are used for data acquisition/control, EPICS clients running on user hosts, and Matlab applications running on a dedicated computer.

## MATLAB APPLICATIONS FOR ISAC AND ISAC-II RADIOACTIVE BEAM FACILITIES

Matlab applications were created for different subsystems at TRIUMF ISAC and ISAC-II radioactive beam facilities. Applications are developed on local Windows PCs. This is much more effective because Matlab desktop needs lots of graphics resources. Development on remote computer shows visible delays in reaction time for the Matlab graphics interface. After tests the source code is deployed to a Linux development machine where it is compiled using a Matlab compiler. Binaries are deployed to a dedicated Matlab Linux production host. One of the advantages of binary Matlab distribution is that applications don't need to get a license for running. MCA was installed both on Windows and

Linux machines. Thus EPICS PVs are accessible in the development and production environment.

A Matlab dedicated host has a mapped network drive which is accessible by user machines. The results of the data processing can be stored on this drive.

## Matlab Applications for the Beam Diagnostic Control System at ISAC/ISAC-II

One of the requirements for this system is to show processed data in real time. It gives the opportunity for beam physicists to tune the devices by looking at visualized processed data. A Beam Diagnostic System (BDS) gives access to collected data via "soft" IOC (SIOC) running on a Linux VME-based system. The description of SIOC running on BDS can be found in these proceedings [2].

Matlab applications are started on a dedicated Matlab Linux host after reboot in a Screen session for multiple-user access. It is necessary for diagnostic purposes and restarting Matlab applications by operators and developers. Matlab applications don't run any Graphics User Interface (GUI). They get data from SIOC using MCA, process them and put back into PV arrays (waveforms). PV arrays are accessed by EPICS clients such as EDM and show them on the graphs. Figure 1 demonstrates an EDM screen for flight-time monitors. A similar screen is used for silicon detectors. A watchdog (WD) in the bottom right corner of the screen indicates that the Matlab application is alive. The Matlab application is running a loop incrementing a watchdog counter. The button *Start Data Processing* in the bottom right corner controls whether the loop bypasses data processing or not. In idling mode most of the time the loop is running in sleep function and doesn't consume computer resources. The green box shows that data are being processed. Tests show that in this mode CPU can be busy up to 30%. Calculated sigma, time of flight for each section, beam velocity, energy, and error are shown in related boxes. Different input parameters are used to control both collecting and processing data. The left part of the screen is used for controlling devices and data acquisition via VME IOCS or SIOC. The right part shows raw and processed data. Initial arrays of data are shown by green color. Matlab application read these data, find the best Gaussian fit and then returns the array with processed data to SIOC. EDM shows these processed data by red color. Vertical markers on both sides of the fitted data are used to cut-off values outside the peaks. The location of the markers on the screen is controlled by the sliders. Markers are kept as arrays in SIOC and used by the Matlab application.

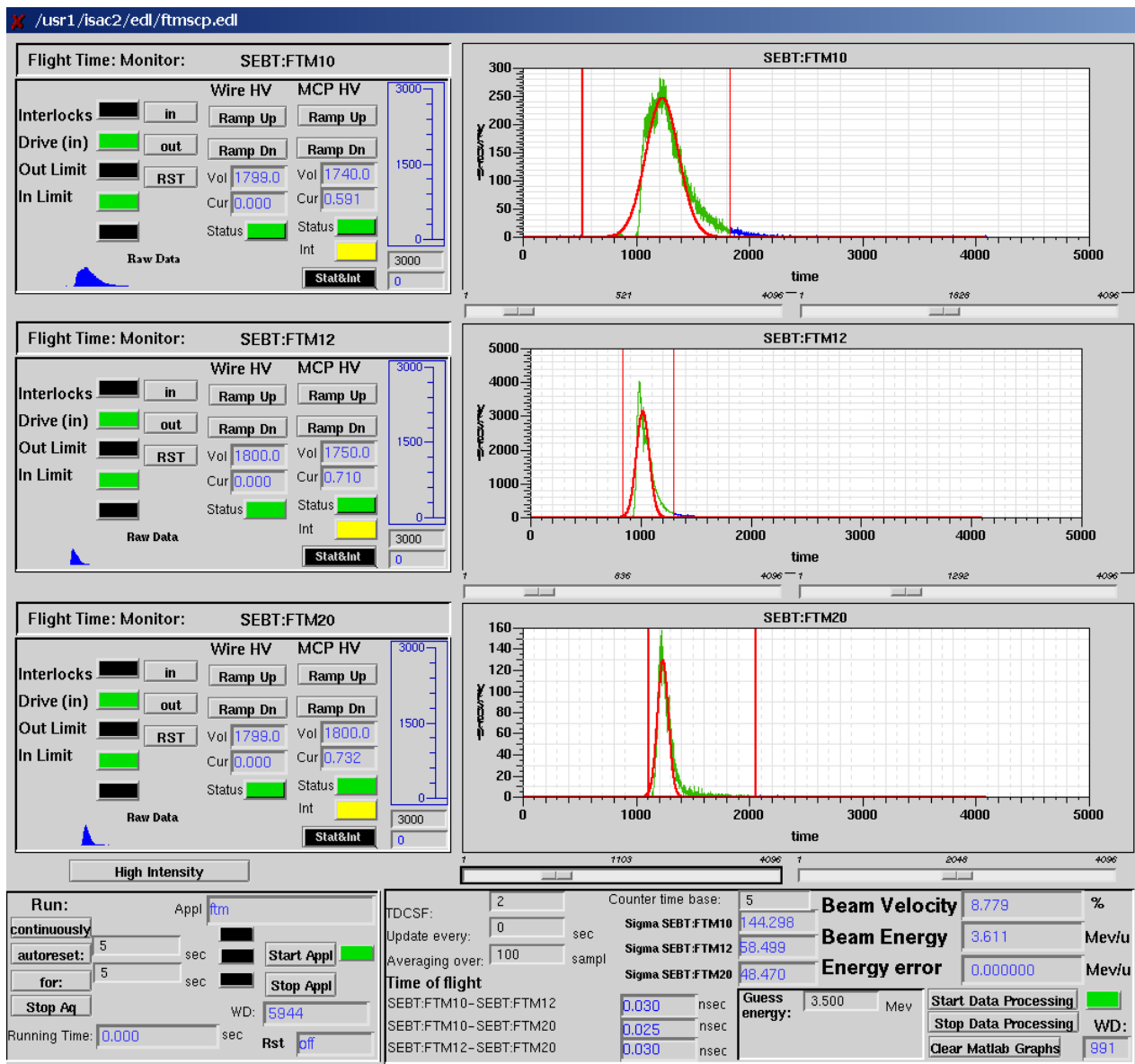


Figure 1: EDM screen used for control and visualization of live data for flight-time monitors at ISAC-II.

### Matlab Application for Emittance Scanning with Pulsed Beam at ISAC

Besides running Matlab applications in a continuous loop it is also necessary to process and visualize data "on demand", by pressing a button on EDM screen. It is also desirable to use visualization capabilities of Matlab which are much better than EDM in the case of 3D graphics. The Matlab application should run on a dedicated remote computer for the same reasons as for the previous system. We used this approach for the TITAN project at ISAC [3]. The EDM screen which is run by users to get emittance for a pulsed beam is shown in Fig 2. Users start the process of emittance scanning by pressing the button *Start*. After finishing the procedure the compiled Matlab application is started automatically on the Matlab Linux host. It is also possible to start the Matlab application

manually by pressing button *Start Matlab3D Plot*. In both cases the ssh command is used to start the application on a remote computer. Emittance scanning data are read from a mapped network disk, processed and saved on the same disk as a PNG file. The right part of the EDM screen is an embedded window where the PNG file is displayed. EDM automatically updates the plot after it is rewritten by the Matlab application.

As in the previous case the Matlab application doesn't run any GUI. However, in contrast to the previous case processed data are saved not in PV arrays but into graphics file. This approach is viable when it is not necessary to get processed data in real time. For TITAN emittance scanning it is perfectly OK because the scanning procedure itself takes several minutes and data processing/visualization takes only a couple of seconds.

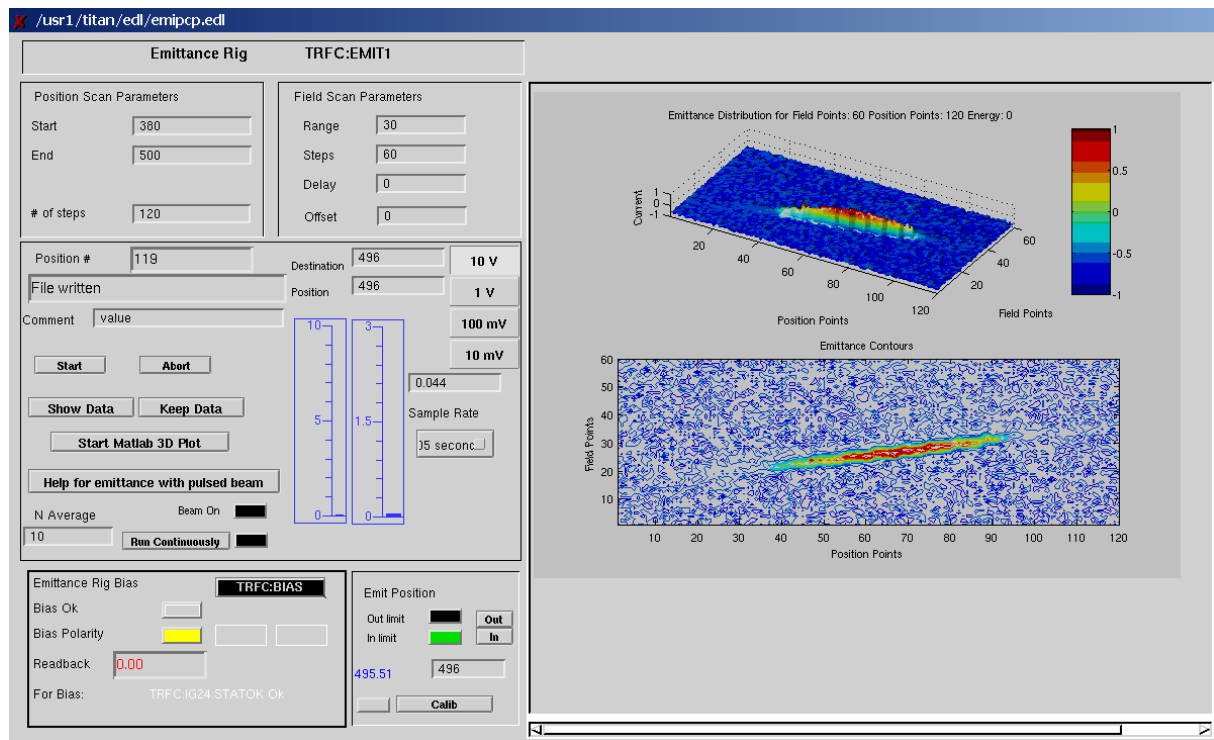


Figure 2: EDM screen used for control emittance scanning and visualization of data.

## DISCUSSION

Matlab is widely used for data processing and visualization. With MCA it is also possible to process and visualize live EPICS data using EPICS clients and SIOC. Matlab 3D visualization capabilities can be effectively used by creating graphics files which can be displayed by EPICS tools. Licensing limitations can be overcome by compiling the Matlab application and running binary code.

A traditional use of Matlab as a desktop integrated development environment is very effective for the development and testing the Matlab application on a PC. Development can be done on one platform and transparently moved and compiled on another. Such a development model provides flexibility in interaction between different development groups.

Another traditional use of Matlab is running GUI, which can control devices via MCA. Such a setup is used by beam physicists at ISAC-II for the development of specific physics applications. These applications are running on a dedicated Linux computer at the ISAC Control room. The Matlab GUI are launched from EDM screens. An important question is the integration of Matlab GUI into the existing EDM “world”. Ideally, fonts, colors, and, as a whole, “the look and feel” should be the same for EDM and Matlab GUI, especially when applications migrate to the operators for day-to-day usage. This work is underway and is part of our Matlab-EPICS integration project.

## ACKNOWLEDGEMENTS

The author thanks J. Richards for her support in the installation and testing of the described systems and K. Ezawa for investigations and problem solving related to the Matlab applications.

## REFERENCES

- [1] MCA, <http://ics-web1.sns.ornl.gov/~kasemir/mca/>
- [2] E. Tikhomolov, “Interfacing of Peripheral Systems to EPICS Using Shared Memory”, this proceedings.
- [3] D. Dale *et al.*, “The Control System for TITAN Experiment at ISAC”, this proceedings.