Real Time Plasma Disruptions Detection in JET Implemented with the ITMS Platform Using FPGA Based IDAQ

M. Ruiz, J. Vega, G. Ratta, E. Barrera, A. Murari, J. M. López, G. Arcas, R. Meléndez, and. JET EFDA contributors

***Abstract*–The use of FPGAs in data acquisition cards for processing purposes allows an efficient real time pattern recognition algorithm implementation. Using 13 JETs database waveforms an algorithm for detecting incoming plasma disruptions has been implemented. This algorithm is written in MATLAB using floating point representation. In this work we show the methodology used to implement the real time version of the algorithm using Intelligent Data Acquisition Cards (IDAQ), DAQ devices with field programmable gate array (FPGA) for local processing. This methodology is based on the translation of the MATLAB code to LabVIEW and the final coding of specific pieces of code in LabVIEW for FPGA in fixed point format. The whole system for evaluating the real time disruption detection (RTDD) has been implemented using the Intelligent Test and Measurement System (ITMS) platform. ITMS offers distributed data acquisition, distribution and real time processing capabilities with advanced, but easy to use, software tools that simplify application development and system setup. The RTDD implementation uses a standard PXI/PXIe architecture. Two 8 channel analog output cards play JETs database signals, two 8 channel DAQ with FPGA acquire signals and computes a feature vector based in FFT analysis. Finally the vector acquired is used by the system CPU to execute a pattern recognition algorithm to estimate an incoming disruption.**

I. Introduction

The physical phenomena leading to disruptions are very complex and non-linear and therefore no satisfactory model has been devised so far either for their avoidance or their prediction. In the work developed by Ratta et al [1] a suitable method for the Real time detection of an incoming disruption in JET is described. The method uses a specific application of Support Vector Machines (SVM) based on an original combination of SVM classifiers. The identification of a disruption is carried out with a two layer classifier. The first layer follows the temporal evolution of the plasma through 3 sequential SVM classifiers and the second one implements a decision function to trigger or not an alarm. The algorithm evaluates the presence of disruptive behaviours every 30ms. The basic steps of the algorithm are:

a) Every 30ms, 30 samples of thirteen signals are acquired. So, the sample rate used to acquire the signals is 1KS/s. Table I shows JET's signals used.

Table I plasma signals used

Id number	Signal name	units
1	Plasma current	A
2	Poloidal beta	
3	Poloidal beta time derivative	s^{-1}
4	Mode lock amplitude.	T
5	Safety factor at 95% of minor radius	
6	Safety factor at 95% of minor radius time derivative	s^{-1}
7	Total input power	W
8	Plasma internal inductance	
9	Plasma internal inductance time derivative	s^{-1}
10	Plasma vertical centroid position	m
11	Plasma density	m^{-3}
12	Stored diamagnetic energy time derivative	W
13	Net power (total input power *minus* total radiated power)	W

b) With the 30ms block a Discrete Fourier Transform (DFT) is performed using a Fast Fourier Transform (FFT) algorithm in order to obtain a spectrum estimation $S_i[k]$ (i refers to signal identification number, see Table I, and k is an index of the DFT coefficients (0<k< 29). Using $S_i[k]$ the spectral deviation σ_{S_i} is estimated by mean of (1). $\overline{S}$ refers to the mean value of $S_i[k]$ for k=1 to 15.

$$\sigma_{S_i} = \sqrt{\sum_{k=1}^{16} \frac{\left|S_i[k]-\overline{S}\right|^2}{15}} \qquad (1)$$

By applying equation (1) to the 13 signals of Table I, the feature vector X_N is build (2):

$$X_N = \{\sigma_{S_0},..,\sigma_{S_{12}}\} \qquad (2)$$

c) The feature vector X_N is used to compute the distances D_j(j = 1, 2, 3), to the respective separating hyper-planes of the first layer classifiers, which correspond to the 3 closest past temporal segments of 30 ms. The equations used to compute the distances are (3), (4) and (5). In these equations α_1, α_2, α_3, $supvec_1$, $supvec_2$, $supvec_3$, are values obtained in the first layer SVM training process.

$$D_1 = \left(\sum_{i=1}^{435} \alpha_{1_i} \cdot e^{-\left\| \sup vec_{1_{mi}} - X_n \right\|^2} \right) - 3{,}28781 \qquad (3)$$

$$D_2 = \left(\sum_{i=1}^{391} \alpha_{2_i} \cdot e^{-\left\| \sup vec2_{mi} - X_{n-1} \right\|^2} \right) - 0{,}877578 \qquad (4)$$

$$D_3 = \left(\sum_{i=1}^{350} \alpha_{3_i} \cdot e^{-\left\| \sup vec_{3_{mi}} - X_{n-2} \right\|^2} \right) - 14{,}4587 \qquad (5)$$

d) These distances are the input to the decision function, equation (6). obtained in the training process of the second layer classifier. The evaluation of this function determines whether an alarm must be triggered or not.

$$\mathrm{R} = 2.0245 \cdot \mathrm{D}_3 + 7.3944 \cdot \mathrm{D}_2 + 39.259 \cdot \mathrm{D}_1 + 48.15 \qquad (6)$$

The complete algorithm must be executed in a cyclical way and the computation time must not exceed 30ms. Both the development of the algorithm and its validation havebeen done using MATLAB.

II. Intelligent Data Acquisition

The actual trend in nowadays data acquisition systems for fusion experiments is to extract the maximum amount of information in the diagnostic analyzed. In this way, it is very interesting to have instrumentation solutions that provide conventional tasks for data acquisition, processing and data logging combined with new techniques based on pattern recognition, event detection and data reduction. The main objective of this work is to include in a conventional fusion DAQ system, DAQ cards with FPGAs devoted to specific processing tasks related with pattern recognition and event detection. These cards are working as intelligent DAQs (IDAQ) devices. Therefore some mathematical or signal processing operations have to be implemented in the IDAQ instead host CPU (in a PXI crate, system CPU or SCPU). The algorithm implementation in a FPGA must consider the limitations imposed by the used of fixed-point arithmetic. Floating-point arithmetic in a FPGA is a very complex task and consumes a lot of hardware resources. In this particular case, the detection of an incoming disruption uses many floating operations in the computation of hyperplane distances obtained with SVM method.

The optimal solution will be the use of only one IDAQ card executing the algorithm in an autonomous way with a host or SCPU acquiring the result of the algorithm execution (the result of variable R in eq(6)). This optimal solution depends on ADC channel numbers and FPGA total amount of resources. Alternatives solutions are the use of more than one IDAQs and the use of a HOST CPU to execute some pieces of code that cannot be fitted in the FPGA..

III. Methodology

With the main objective of implementing as much as possible processing functions and pattern recognition algorithms in the FPGA we have developed several steps described in the following paragraphs. First, we have analyzed the MATLAB code implemented for the detection of the incoming disruption. This code is written using standard MATLAB functions using floating-point numbers representation (double). The code reads JET's database files and executes the algorithm. This code has been rewritten in LabVIEW in order to obtain conclusions about its performance and how to tune it to run in real time. In this sense the MATLAB code performs 30 points FFTs, and the LabVIEW version uses 32 point FFTs (so, the time block is now 32 ms instead 30ms). With this modification, we increase the execution speed of the algorithm and we simplify the implementation in the FPGA version, because intellectual properties (IPs) hardware modules available are only power of two). This modification has not impact on the behavior of the algorithm because the result in the detection of the incoming disruption is the same. In the system CPU, the execution time of FFT and spectral estimation for the 13 signals is 6ms and the hyper-plane distance computation (D_1, D_2, D_3) is 2ms. The global execution time for the algorithm is approximately 8ms, so it is possible to execute it in real time. Secondly with regard to translating the LabVIEW code to a FPGA target, we have rewritten the code using fixed-point number representation. In this sense, it is possible to adjust every particular variable to a specific fixed-point size. This is very useful in order to dimension adequately the resources in the FPGA. Thirdly, using LabVIEW/FPGA, we have coded the processing functions in order to estimate the amount of resources consumed in the synthesis process and we have verified the correct operation of the functions. The conclusions of this step are the following:

- It is possible to implement the FFT but only for one channel. The amount of resources consumed by the FFT IP block is very high for implementing more than one in the FPGA device available.
- A sequential implementation of the FFT for 8 channels could be implemented.
- It is possible to implement the spectral deviation estimator in the FPGA.
- Additionally we try to implement the equations for computing the distances but the amount of resources consumed is very high for the FPGA available. We use RAM memory to implement look-up tables with coefficients but the result is the impossibility to fit the design in the FPGA.
- Finally, there is not a DAQ card with FPGA with more than eight channels in the form factor (PXI) selected for the implementation. Therefore, we need to use two cards and implement some parts of the algorithm in the system CPU. Each FPGA based DAQ card implements the FFT algorithms for eight channels and the spectrum deviation estimation. The host is in charge of executing the pattern recognition algorithm computing the distances and taking the decision.

IV. Implementation Details

In order to implement the complete system we have chosen the ITMS platform [2][3]. This platform uses the PXI technology form factor. For this specific implementation, we use commercial products from National Instruments. The chassis is a hybrid PCI and PCIe chassis in 3U size model PXIe-1062Q, the embedded system CPU (SCPU) is the model PXIe8130. We have added two high-speed analog 8 output modules PXI6713 (up to 1 MS/s of per channel update), two 8-channel intelligent data acquisition cards PXI-7851R (750kS/s of maximum sample rate and 16 bits of resolution), and an external PC for development purposes. The analog output cards have been added in order to generate the analog signals, using files from JET database, and implement a test of the algorithm.

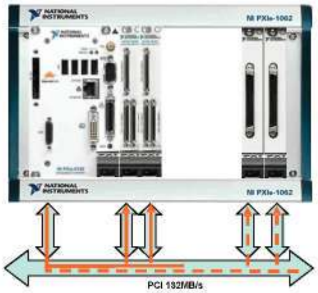


Fig 1: PXI system used for the disruption detection.

The software used in the implementation are the ITMS modules, see ref [2], based on the use of LabVIEW Real Time for the SCPU, NIDAQmx and RDA as drivers for analog output generation and data acquisition from FPGA DAQ cards, and LabVIEW development tools in a PC. The details of the implementation are described in the following sections .

A. Signal Generation.

The signal generation process is implemented using the two DAQ cards PXI6713 (8 channels on each one) for generating the 13 necessary signals (see Fig 2). An application developed in LabVIEW sends the samples to the DAQ cards (S_GEN). The signals are reproduced using binary files constructed using JET database files. The waveform must be normalized to transform the engineering values to voltages and interpolate (up to 100kHz) to enhance the quality of the signal generated. The application can play different signal sets of different discharges.

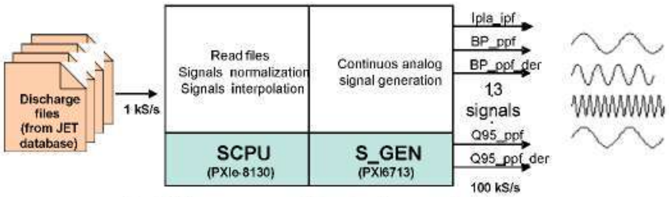


Fig 2: Resources used in signal generation process

B. Data Acquisition and Spectral Deviation Estimation.

The 13 signals are acquired using two PXI7851R cards (RDAQ in Fig.3). These cards include a VIRTEX 5 LX60 FPGA. We are going to use the eight channels available for the two cards. In the second card only five channels results will be taking into account. The code for each DAQ card is the same. The operations executed by the FPGA are summarized in Fig 4.

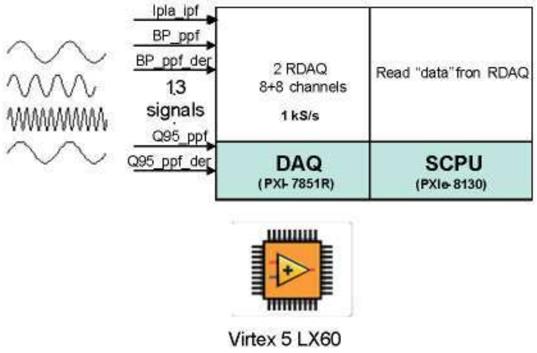


Fig 3: Block diagram of signal acquisition and processing.

The signals are sampled at 1kS/s and the binary values are stored in eight FIFO memories. When 32 samples of one channel are available in the FIFO, the FFT processor executes the FFT algorithm and stores the result in another FIFO (FFTFIFO). This process is executed reading the information for the eight FIFO channels. The information stored in the FFTFIFO is read by another hardware processor devoted to compute the spectral deviation. The result of this deviation for the 8 channels is sent using direct memory access (DMA) to the HOST.

In this solution, one IDAQ card has 8 channels connected and the other only use 5 of 8 channels.

C. Execution of the Pattern Recognition Algorithm in the System CPU.

With the solution implemented, vectors from the data acquisition cards are read. Therefore, the vector read must be passed to a function that computes the distances to the hyperplane. Figure 5 shows the LabVIEW code running in the SCPU implementing the calculation of the distances and the decision function

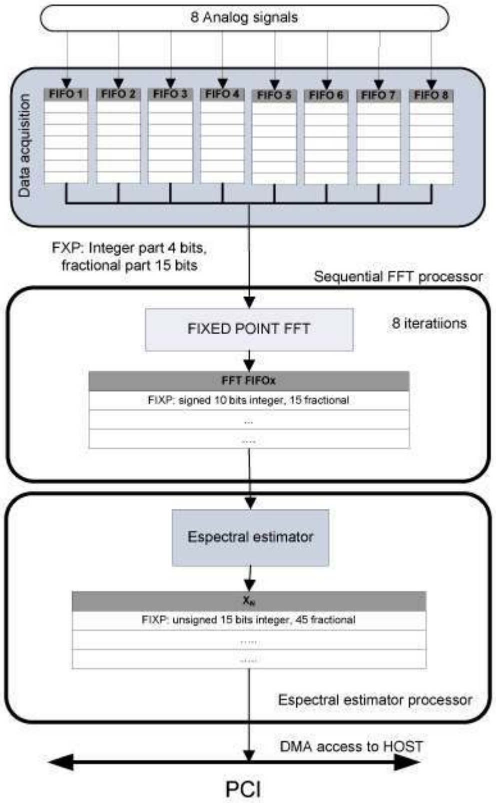


Fig 4: Hardware implemented in the FPGA using LabVIEW.

V. RESULTS AND CONCLUSIONS

In order to verify the correct performance of the implementation, the test bed has been tested using several discharges files from JET's database. We have demonstrated that the results obtained are the same that are obtained with the MATLAB simulations [1].

The total amounts of resources used into the FPGA are the following (data summarized from XILINX compiler):

- Total Slices: 61.6% (11820 out of 19200)
- Flip Flops: 45.2% (8669 out of 19200)
- Total LUTs: 55.2% (10595 out of 19200)
- DSP48Es: 68.8% (22 out of 32)
- Block RAMs: 34.4% (11 out of 32)

The CPU load in the SCPU is less than 3%. So we can conclude that this is an efficient real time implementation of the RTDD that can be included in a conventional fusion DAQ system.

ACKNOWLEDGMENT

This work is funded by the Spanish Ministry of Science and Technology under the Projects Nº DPI2006-06624, ENE2009-10280 and Nº ENE2008-08294/FTN.

This work, supported by the European Communities under the contract of Association between EURATOM/CIEMAT, was carried out within the framework of the European Fusion Development Agreement. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

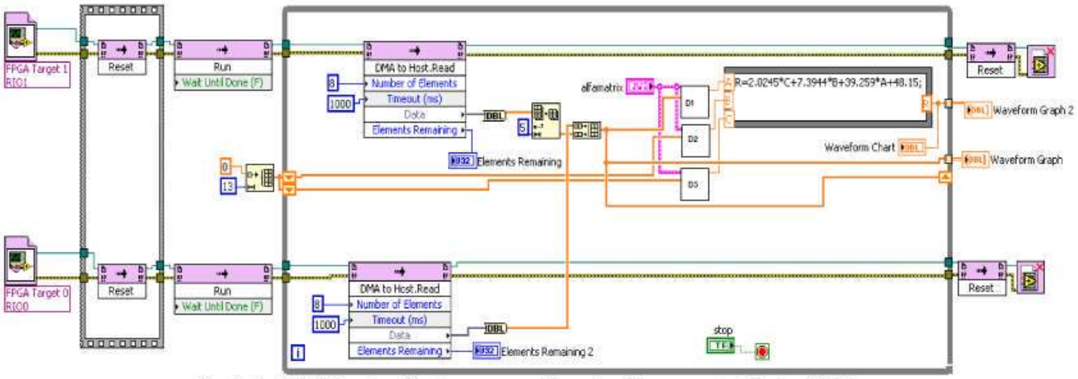
The authors are grateful for the aid provided by National Instruments for lending the equipment necessary to develop the test-bed.

REFERENCES

[1] G.A. Rattá , J. Vega , A. Murari , G. Vagliasindi , M.F. Johnson , P.C. de Vries and JET EFDA Contributors. "An advanced disruption predictor for JET tested in a simulated real-time environment" Nuclear Fusion, vol 50, no 2, pp , Feb. 2010.

[2] M. Ruiz, JM. López, G. de Arcas, E. Barrera, R. Melendez and J. Vega. Data reduction in the ITMS system through a data acquisition model with self-adaptive sampling rate. Fusion Engineering and Design, vol 83, no 2-3, pp 358-362, Apr 2008.

[3] J. González, M. Ruiz, E. Barrera, J.M. López, G. de Arcas and J. Vega "Service-oriented architecture of adaptive, intelligent data acquisition and processing systems for long-pulse fusion experiments" Fusion Engineering and Design, 2010.



```
R=2.0245*C+7.3944*B+39.259*A+48.15;
```

Fig 5: LabVIEW code of pattern recognition algorithm executed in the SCPU