EFDA–JET–CP(09)04/04

M. Ruiz, J. Vega, E. Barrera, J. González, A. Murari, R. Meléndez, G. Rattá,
S. González and JET EFDA contributors

Test-Bed of a Real Time Detection System for L/H & H/L Transitions Implemented with the ITMS Platform

Test-Bed of a Real Time Detection System for L/H & H/L Transitions Implemented with the ITMS Platform

M. Ruiz[1], J. Vega[2], E. Barrera[1], J. González[1], A. Murari[3], R. Meléndez[1], G. Rattá[2],
S. González[2] and JET EFDA contributors*

JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK

[1]Grupo de Investigación en Instrumentación y Acústica Aplicada. Universidad Politécnica de Madrid,
Crta. Valencia Km-7 Madrid 28031 SPAIN
[2]Departamento de Sistemas Electrónicos y de Control, Universidad Politécnica de Madrid,
Crta. Valencia Km-7 Madrid 28031 SPAIN
[3]Consorzio RFX – Associazione EURATOM ENEA per la Fusione, Padova, Italy
** See annex of F. Romanelli et al, "Overview of JET Results",*
(Proc. 22nd IAEA Fusion Energy Conference, Geneva, Switzerland (2008)).

Preprint of Paper to be submitted for publication in Proceedings of the
7th Technical Meeting on Control, Data Acquisition and Remote Participation for Fusion Research,
Aix-en-Provence, France.
(15th June 2009 - 19th June 2009)

ABSTRACT.

A basic requirement of the data acquisition systems used in long pulse fusion experiments is to detect events of interest in the acquired signals in real time. Developing such applications is usually a complex task, so it is necessary to develop a set of hardware and software tools that simplify their implementation. An example of these tools is the Intelligent Test and Measurement System (ITMS), which 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. This paper presents the application of the ITMS platform to solve the problem of detecting L/H & H/L transitions in real time based in the use of efficient pattern recognition algorithms.

1. INTRODUCTION

This paper presents the development of a real time L/H and H/L detection test-bed in JET based on the ITMS platform. The ITMS platform offers a set of tools to simplify the system setup, from the data acquisition process, to the data distribution among multiple CPUs and the definition of the data processing algorithms running in CPUs [1,2,3].

In this work, the platform is used to implement a realtime detection system for L/H and H/L transitions in order to test its performance and to demonstrate its usability. The implementation uses the pattern recognition algorithm described in ref [4].The system has been tested with real signals reproduced offline using data obtained from JET databases for a set of previous discharges. The results obtained with the system are compared with the transition times determined by the experts [5].

2. TEST-BED DESCRIPTION

The test-bed implementation is based on the use of the ITMS platform, which, in this case, consists of the following elements (see figure 1):

- **Hardware:** National Instruments (NI) Hybrid PCI and PCIe chassis in 3U size PXIe-1062Q, embedded system CPU (SCPU) NI-PXIe8130, two high-speed analog 8 output modules NIPXI6713 (S_GEN) (up to 1 MS/s of per channel update), 16-channel data acquisition card NIPXIe- 6251(DAQ) (1.25 MS/s of maximum sample rate), peripheral processing CPU (PCPU) ICP-P4 from Innova Computers and an external PC.
- **Software:** SCPU: LabVIEW RealTime and NIDAQmx; PCPU: Windows, standard LabVIEW and standard MATLAB; External HOST: JAVA, JINI [6] and SCXML engine [7], LabVIEW development tools and user interface. SCPU does not have graphical user interface because is running LabVIEW/RT.

The system is developed with two parts, as shown in Figure 2. The first part is in charge of signal generation. This task of the test-bed does not belong to the system performance evaluation. Its goal is simply to generate continuously the analog signals so that the system can work with them.The second part deals with data acquisition and processing implemented in the ITMS platform. The features of these parts are described in the following paragraphs.

a) The signals are reconstructed by retrieving data from JET's databases. Once a discharge is selected, the data acquired previously for the subsequent signals are retrieved from JET's databases files: Bndiam, Ptot, Wmhd, RxPL, ZxPL, Bt80, Bt, FDWDT, Q95. These signals are used for L/H and H/L transition detection. Each signal, which is originally sampled at 1 kS/s, is normalised by using the minimum and maximum. These values are obtained as the minimum and maximum values of the set of discharges used during the training phase of the pattern recognition algorithm. Then, the normalised signal is upsampled and linearly interpolated up to 100 kS/s to prepare it for generation with the PXI6713 cards because the cards do not have analog reconstruction filter. Two cards are used to generate the aforementioned 9 signals by adding the Mode signal, which is also stored in the same database files. This signal indicates the state of the system, H or L, for every sampling interval.

b) Data acquisition and real time data processing take place in the ITMS platform. The generated signals are acquired and analysed continuously with the ITMS hardware and software. The data acquisition process runs in SCPU as a LabVIEW real time task implemented with NIDAQmx. This task collects data with a sample rate of 40kS/s in blocks and sends them to PCPU using another real time tasks. PCPU receives data samples and stores them in a FIFO. Therefore, the user application can read this FIFO to process the data. PCPU is running a standard windows operating system with LabVIEW and Matlab.

In order to configure the entire system, an ITMS platform client, running on a PC, is used with a simple user interface, like that shown in Figure 3. The user configures the data acquisition parameters in the “activeDAQslots” control, the PCPU installed “activePCPUslots“, the channel configuration and the name assigned “channel setup” and the distribution process (data acquired by SCPU are moved to the PCPU) indicated in”“chantoPCPU”. This control also has the LabVIEW program (VI) associated with a specific channel.

3. IMPLEMENTATION OF LH & HL TRANSITION DETECTION

The algorithms used to detect the transitions use the following set of signals: Bndiam, Ptot, Wmhd, RxPL, ZxPL, Bt80 for L/H transitions; and Bndiam, Bt, FDWDT, Q95, RxPL, ZxPL for H/L transitions. As three of these signals (Bndiam, RxPL, ZxPL) are used in both algorithms, each of them has been connected to two input channels of the PXIe6251 DAQ card to simplify system setup. The same was done with the Mode signal in order to simplify the evaluation of the results. The system was developed and tested using two different implementations. Both are based in the use of a SVM classifier trained and tested with datasets from 50 discharges. This classifier estimates the class of the input feature vector and the distance to the separating hyper-plane. The complete theoretical description of LH and HL transition time detection algorithm is described in ref [4].

METHOD 1.

All 14 input signals are continuously acquired at 40kS/s and sent to the PCPU for processing. The Dynamic Data Processing System (DDPS) module [6] running in the PCPU is configured to run a LabVIEW VI program that performs the following operations (Figure 4 and 5):

- Read from the FIFO input buffers the acquired samples in blocks of 4096 samples, corresponding to a signal duration of approximately 100ms.
- Divide every block into segments, whose lengths depends on a parameter (decimation factor,

N), and averaging the block to obtain a mean value. This mechanism allows selection of the output sample rate to be chosen by the software.

- De-normalise the value using the maximum and minimum values of the LH model for channels 0 through 6, and the parameters of the HL model for channels 8 through 12. These two models are generated in MATLAB in the training phase and are stored in MATLAB .mat files. These models are loaded on ITMS modules at start-up. Channels 8 and 12 are not de-normalised because they correspond to the Mode signal.
- Compute the vector distance to hyper-plane using a LabVIEW subVI that calls a MATLAB command window; a SPIDER [8] function, named“test”, is executed to compute the hyperplane distance using the feature vector and the SVM model obtained in the training process. The same process runs twice in parallel; using the LH vectors in one instance and using the HL in the other.
- Finally, display the distance to the hyper-plane in a graph together with the Mode signal to simplify their comparison, and indicate the transitions with a LED indicator. If the distance is positive, the mode is L, and if it is negative, the mode is H.

METHOD 2.

The system model implemented in this case is the same as the one described above, but it is developed using the ITMS tools for representing an experiment through a state machine specified in SCXML. This experiment has two phases, or profiles. In the first phase, the system is in the L mode, so we need to run specific processing to detect LH transition. This phase ends when the distance becomes negative. In the second phase, the system is in the H mode, so we run the HL transition processing. This phase ends when the distance becomes positive.

When the ITMS system starts, the DDPS module begins running the LH detector because the plasma starts in the L mode. Once a LH transition is detected, a message is automatically sent to the ITMS-IDAQ&P module, which runs an application implementing the state machine specified in SCXML. Simultaneously, the LH detector is replaced with a HL detector, because the plasma is now in H mode. It takes 15 ms to unload the LH detector dynamically, and another 60 ms to load the Hl detector. During this period, no data is lost because the FIFO input buffers store the acquired data. In order to load the LH and HL detectors dynamically, two processing profiles must be defined in the ITMSIDAQ& P module, as shown in Figure 6.

4. RESULTS & DISCUSSION.

Results obtained with the first methodology are shown in Figure 7 for data from JET’s Pulse No: 55596. The figure shows the output signals of both transition detection algorithms (transitionLH and transitionHL), which indicate the state of the system (distanceLH ITMS, distanceHL ITMS and mode ITMS). The results obtained in an offline analysis using Matlab (distanceLH and distanceHL) and the mode signal retrieved from JET database [5] are also displayed in the figure. The figure shows the difference in the transition time between the database mode signal andthat computed by the ITMS.

It has been proved that the system developed using the first methodology works well in real time in the ITMS platform. However, it presents several disadvantages: first, both transition detection algorithms are run in parallel, where only one is needed at any given moment depending on the system state; second, due to the use of sample blocks of 4096 samples, we need a delay of approximately 100ms to start the analysis. If the block size is reduced, calls to Matlab become more frequent, which significantly increases the processing time. A solution for this would be to implement the hyperplane distance computations routines directly in LabVIEW.

It has been proved that the system developed using the first methodology performs correctly in real time in a continuous scenario, using a decimation factor of N = 64 samples, which means obtaining a state variable output sample approximately every 1.6ms.

The computation time using the second methodology is reduced, because the system knows the current state at all times, and it therefore only runs the appropriate transition detection algorithm. This allows the use of a decimation factor of N = 32 samples, which means producing a state variable output sample approximately every 0.8ms, which is faster than the sampling rate of the original signals (1ms). In addition to this, the second methodology presents the advantage of embedding all the system logic into the SCXML state machine, simplifying the development process. On the other hand, it introduces an additional delay of 60ms and 15ms for every HL and LH transition, respectively.

This is a potential problem if we have oscillations in the mode. This second approximation allows the analysis algorithm to be modified easily because users can mix more sophisticated routines based on other mathematical approximations that increase the success of the classifier. The purpose of having evaluated this methodology in this application has been to prove the advantages it presents in the development of intelligent data acquisition systems.

ACKNOWLEDGEMENTS

This work was partially funded by the Spanish Ministry of Science and Innovation under the Projects Nº DPI2006-06624 and Nº. ENE2008-02894/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]. J. Vega, A. Murari, B. Carvalho, G. de Arcas, R. Felton, M. Riva, M. Ruiz, J. Svensson and JETEFDA Contributors. New developments at JET in diagnostics, real-time control, data acquisition and information retrieval with potential application to ITER. Fusion Engineering and Design, In Press, Corrected Proof, Available online 7 May 2009.

[2]. M. Ruiz, JM. López, G. de Arcas, E. Barrera, R. Melendez, J. Vega. Data reduction in the ITMS system through a data acquisition model with selfadaptive sampling rate. Fusion Engineering and Design, Volume 83, Issues 2-3, April 2008, Pages 358-362

[3]. De Arcas G, Lopez JM, Ruiz M, et al. Selfadaptive sampling rate data acquisition in JET's correlation reflectometer. REVIEW OF SCIENTIFIC INSTRUMENTS Volume: 79 Issue: 10 Article Number: 10F336 Published: OCT 2008

[4]. J. Vega, A. Murari, G. Vagliasindi, G. A. Rattá. "Automated estimation of L/H transition times at JET by combining Bayesian statistics and Support Vector Machines". Submitted to Nuclear Fusion.

[5]. A. J. Meakins, D. C. McDonald. "The Application of Classification Methods in a Data Driven Investigation of the JET L-H Transition". In submission to Plasma Physics and Controlled Fusion.

[6]. J. González, M. Ruiz, E. Barrera, JM. López, G. De Arcas, J. Vega. Configuration and supervision of advanced distributed data acquisition and processing systems for long pulse experiments using JINI technology. Fusion Engineering and Design, In Press, Corrected Proof, Available online 28 January 2009

[7]. J. González, M. Ruiz, E. Barrera, J.M. López, G. de Arcas, J. Vega. Services oriented architecture for adaptive and intelligent data acquisition and processing systems in long pulse fusion experiments. Proceedigns of this conference.

[8]. "The Spider - A machine learning in Matlab".http://www.kyb.tuebingen.mpg.de/bs/people/spider/main.html. Max-Planck Institute for biologicalCybernetics, Tuebingen, Germany.

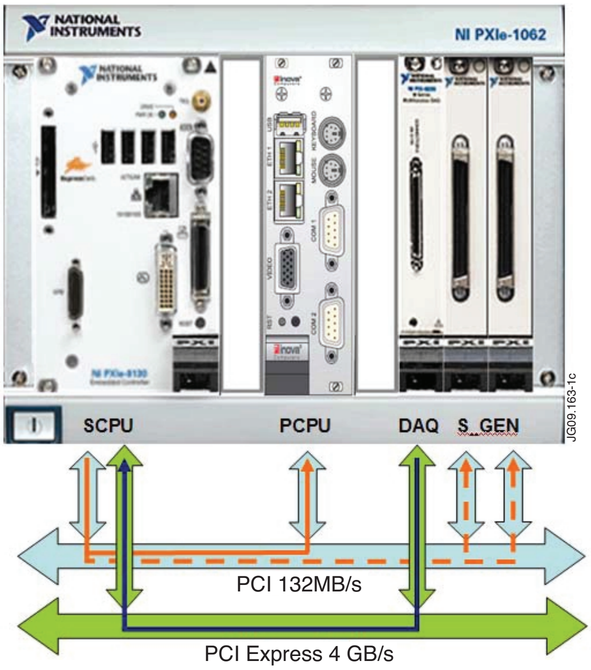


Figure 1: Test-bed hardware implementation.

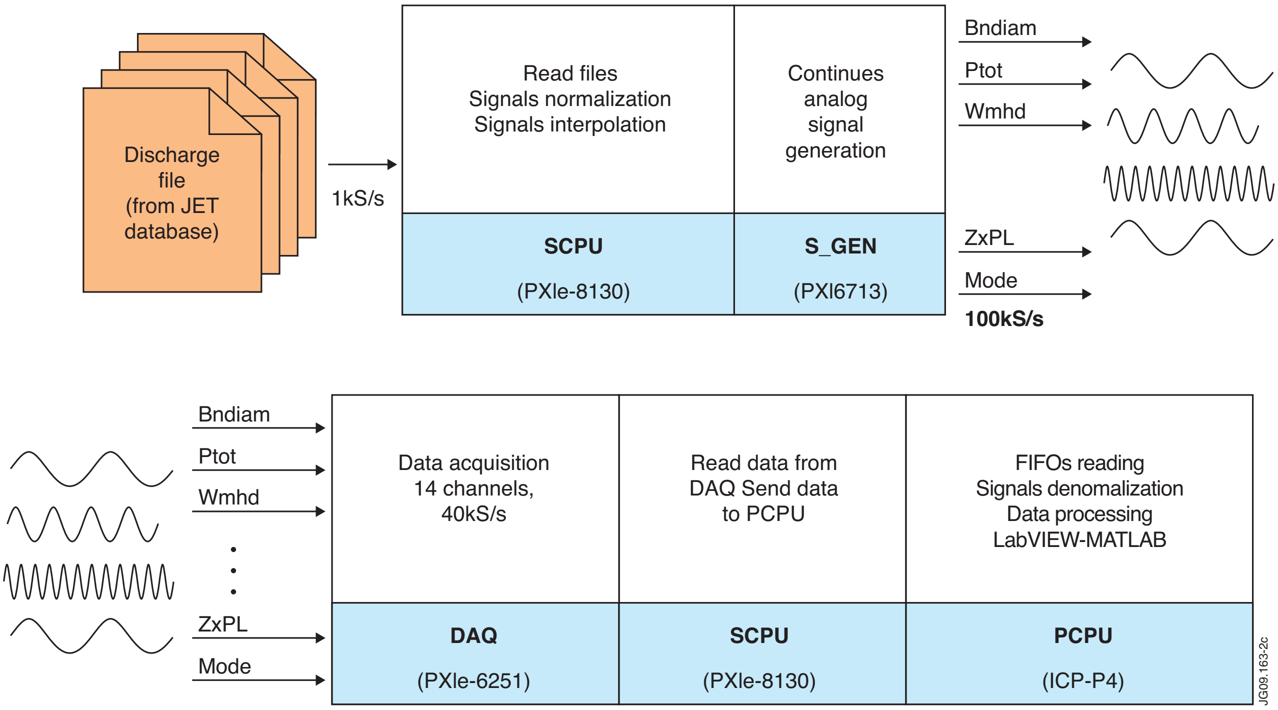
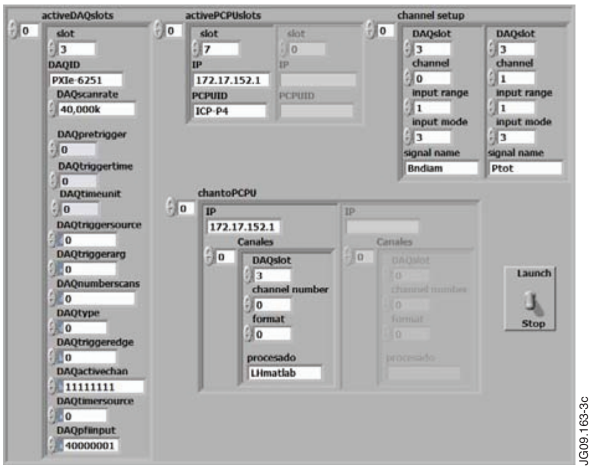


Figure 2: Test-bed description (a) Read files and signal generation (b) Data acquisition and real time data processing.



activeDAQslots 0
slot 3
DAQID PXIe-6251
DAQscanrate 40,000k
DAQpretrigger 0
DAQtriggertime 0
DAQtimeunit 0
DAQtriggersource 0
DAQtriggerarg 0
DAQnumberscans 0
DAQtype 0
DAQtriggeredge 0
DAQactivechan 11111111
DAQtimersource 0
DAQpfiinput 40000001

activePCPUslots 0
slot 7
IP 172.17.152.1
PCPUID ICP-P4
slot 0
IP
PCPUID

channel setup 0
DAQslot 3
channel 0
input range 1
input mode 3
signal name Bndiam
DAQslot 3
channel 1
input range 1
input mode 3
signal name Ptot

chantoPCPU 0
IP 172.17.152.1
Canales 0
DAQslot 3
channel number 0
format 0
procesado LHmatlab
IP
Canales 0
DAQslot 0
channel number 0
format 0
procesado

Figure 3: User interface client for setup ITMS.

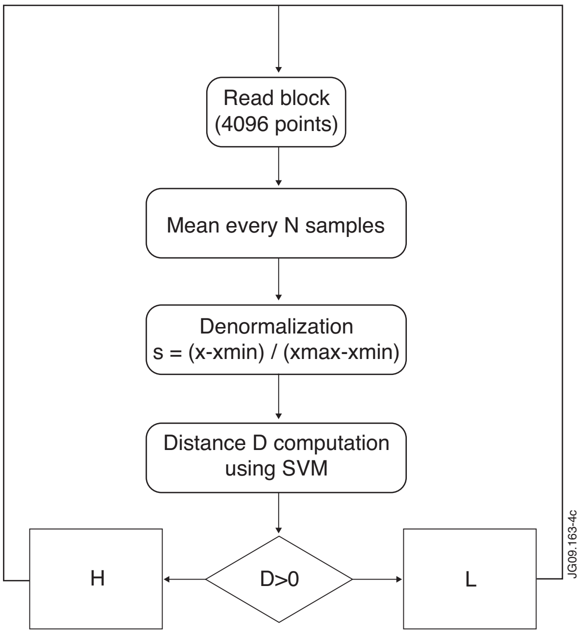


Figure 4: First method flow chart.

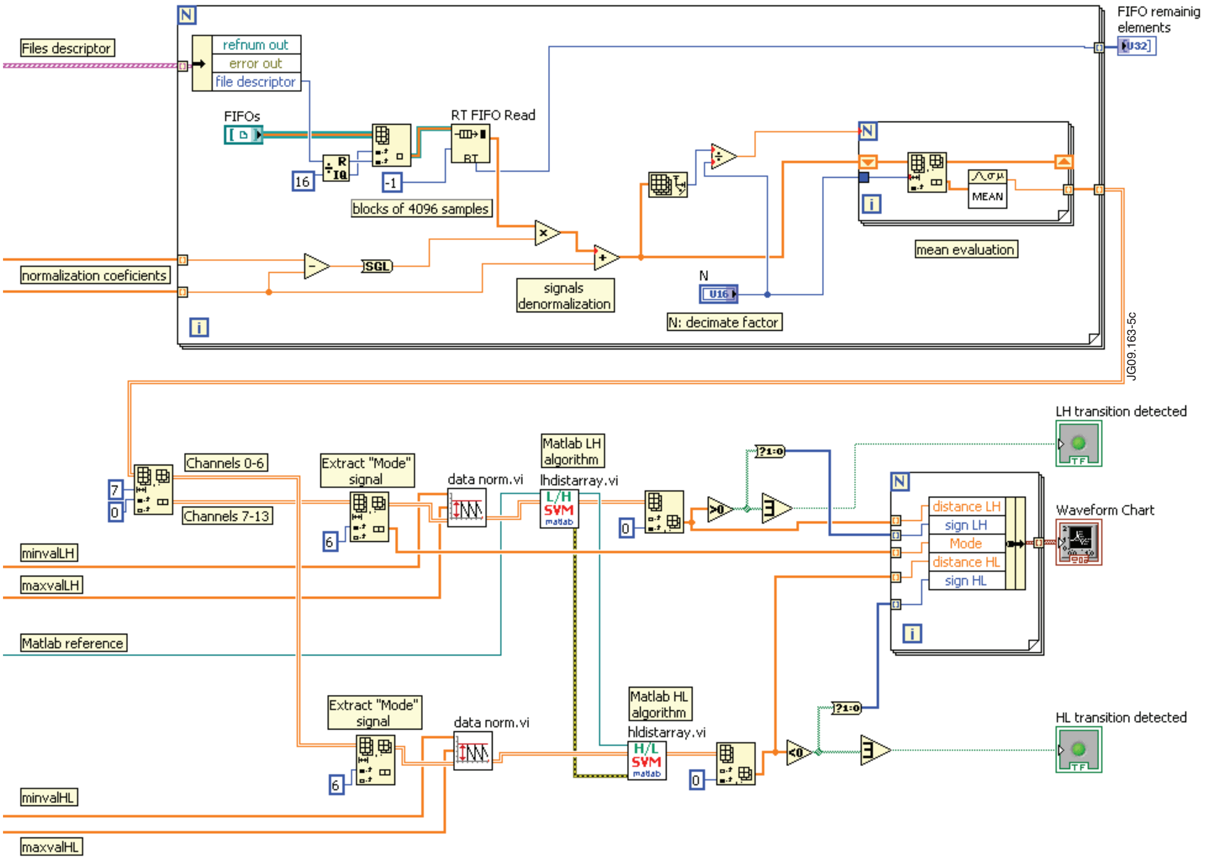


Figure 5. LabVIEW block diagram of method 1 VI.

```
<?xml version="1.0" ?>
<test xmlns"">
<profile>
<processing>transitionLH</
processing><channel>B
ndiam,Ptot,Wmhd,RxPL,ZxPL,Bt80</channel>
</profile>
<profile>
<processing>transitionHL</
processing><channel>B
ndiam,Bt,FDWDT,Q95,RxPL,ZxPL</channel>
</profile>
…..
</test>
```

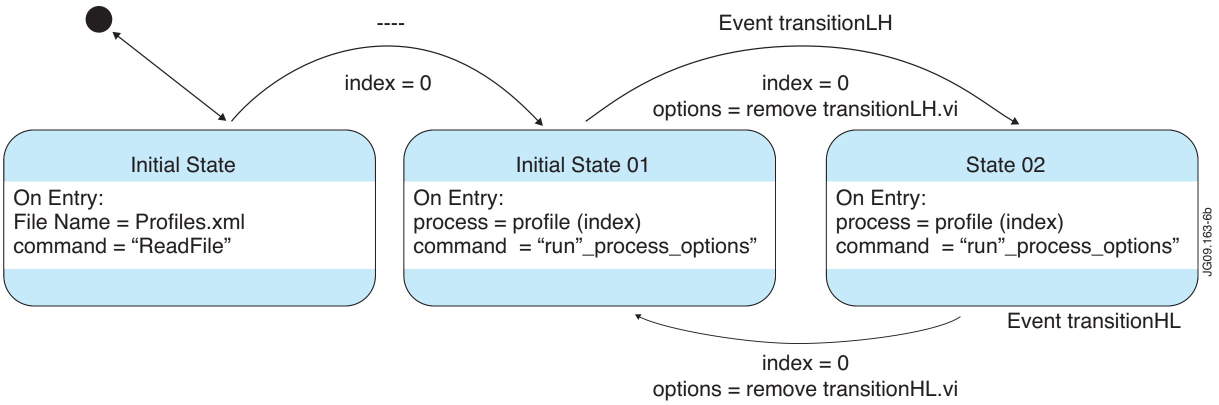


Figure 6. Second method description. (a) Profile.xml content (b) State machine

```
<?xml version=”1.0" ?>
<scxml
xmlns=”http://www.w3.org/2005/07/scxml
" version=”1.0"
initialstate=”Initial_State”>
….
<state id=”Initial_State”>
<onentry>
<assign name=’FileName’
expr=’”Profiles.xml”’ />
<assign name=’command’
expr=’”ReadFile”’ />
</onentry>
<transition target=”State01">
<assign name=”index” expr=’0' />
</transition>
</state>
<state id=”State01">
<onentry>
<assign name=”process”
expr=’profile[index]’/>
<assign name=”command”
expr=’”run_process_options”’ />
</onentry>
<transition event=”transitionLH”
target=”State02">
<assign name=”index”
expr=’1' />
<assign name=”options”
expr=’”remove=transitionLH”’ />
</transition>
</state>
<state id=”State02">
<onentry>
<assign name=”process”
expr=’profile[index]’ />
<assign name=”command”
expr=’”run_procesado_options”’ />
</onentry>
<transition event=”transitionHL”
target=”State01">
<assign name=”index” expr=’0' />
<assign name=”options”
expr=’”remove=transitionHL”’ />
</transition>
</state>
```

(c) State machine description in SCXML.

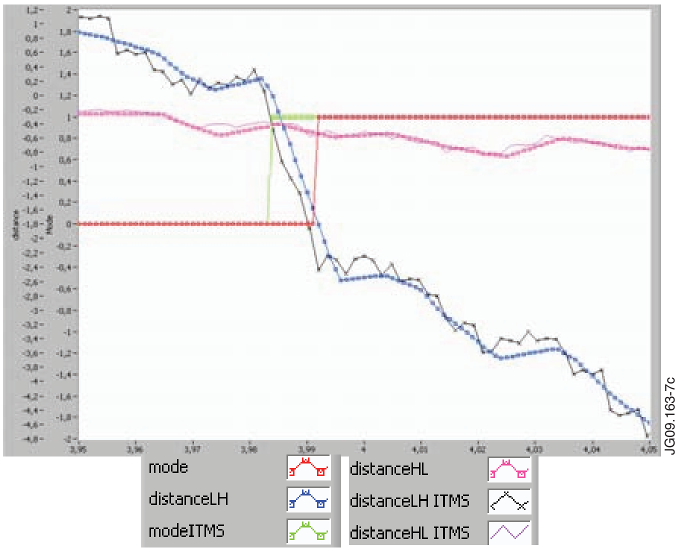


Figure 7. LabVIEW displayed results of method 1.