EFDA–JET–CP(09)04/06

F. Sartori, A. Barbalace, A.J.N. Batista, T. Bellizio, P. Card, G. De Tommasi,
P. Mc Cullen, A. Neto, F. Piccolo, R. Vitelli, L. Zabeo
and JET EFDA contributors

The PCU JET Plasma Vertical Stabilisation Sontrol System

The PCU JET Plasma Vertical Stabilisation Sontrol System

F. Sartori[1], A. Barbalace[4], A.J.N. Batista[2], T. Bellizio[5], P. Card[1], G. De Tommasi[5], P. Mc Cullen[1], A. Neto[2], F. Piccolo[1], R. Vitelli[3], L. Zabeo[1]
and JET EFDA contributors*

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

[1]*EURATOM-UKAEA Fusion Association, Culham Science Centre, OX14 3DB, Abingdon, OXON, UK*
[2]*Associação EURATOM/IST, Instituto de Plasmas e Fusão Nuclear, Av. Rovisco Pais, 1049-001 Lisboa, Portugal*
[3]*Dipartimento di Informatica, Sistemi e Produzione, Università di Roma, Tor Vergata, Via del Politecnico 1-00133, Roma, Italy*
[4]*Euratom-ENEA Association, Consorzio RFX, 35127 Padova, Italy*
[5]*Associazione EURATOM/ENEA/CREATE, Universita di Napoli Federico II, Napoli, Italy*
** See annex of F. Romanelli et al, "Overview of JET Results", (Proc. 22 nd 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.
Modern tokamak machines are designed to run elongated X-point configurations for which the plasma is vertically unstable. To enable plasma operation beyond the vessel time the introduction of a Vertical Stabilisation (VS) System is required. Furthermore, in presence of top-bottom asymmetric passive structures, fast perturbation to the internal pressure (ELMs, H-L transition) can perturb the unstable vertical mode. This is the case for JET where H mode is restricted by the risk that an ELM could trigger a Vertical Displacement Event (VDE) leading to a disruption.

The PCU enhancement project was setup with the aim of effectively increasing the JET operating space by both improving the Vertical Stabilisation System Controller and increasing the Radial Field Amplifier power. The hardware and software technological advancements that have been used allowed to implement a 200 channel, 20kHz control system with less than 2ms jitter and over 2*Gops* of processing capability. The software design, driven by an object oriented analysis, enabled a good compromise between real-time performances and reliability-maintainability objectives.

The PCU project has adopted from its beginning a model-based approach to solve the plasma vertical stabilisation control problem. The method implies that, at all levels, the control system will be tested on a model of the machine before being validated on the real plant. The controller software is therefore to be tested against the plasma models before being used in the machine. The software was designed to allow the tests against the model, both by providing a multi-platform compatibility (although the code runs on RTAI, it can be run offline on Windows or Linux without changing the source code) and by organising a clear separation between real-time I/O, configuration and control interfaces and scientific/technical code. The first operational experience of the system will be presented, especially focussing on the first 5 months when the system has been operating in parallel to the old VS and the controller has been progressively commissioned.

1.§INTRODUCTION

The Plasma Control Upgrade (PCU) project aimed at increasing the capabilities of the Vertical Stabilization (VS) system [1]. It is envisaged that this upgrade will enhance the ability of the VS system to recover from large Edge Localised Mode (ELM) perturbations, in particular for the case of plasmas with high elongation, i.e. plasmas with large growth rate parameters.

The Vertical Stabilisation is one of the most critical systems at the JET Tokamak. It is responsible for guaranteeing a zero vertical velocity of the plasma while at the same time trying to keep the average radial field to zero [6]. The stabilisation is obtained by using a velocity and a current control loop: the former is a proportional control which stabilises the vertical velocity of the plasma, while the latter has a proportional-integral action and tracks a current reference, usually zero, in order to prevent drifts in the vertical position and to avoid exceeding the limits of the power supply. As it is not possible to directly infer the vertical velocity of the plasma, an observer capable of translating the magnetic Preprint submitted to Elsevier June 11, 2009 signals from the vessel to an estimation of the vertical velocity is required.

Along with these two components, the Vertical Stabilisation system also provides a collection of features that enable the scientific exploitation of the system during experiments. In particular it is able to apply constant voltage references for a defined period of time, named kicks [7], to either the vertical amplifier or the divertors, at a certain frequency or as response to a particular event, e.g. ELMs. Finally and in order to test, commission and diagnose problems, the Vertical Stabilisation system also provides a way of simulating a full JET pulse in closed loop using a model of the plasma[4]. Due to the complexity of the requirements, instead of building monolithic and unmanageable software, it has been decided to divide the controller in smaller and independent blocks, thus facilitating its debug and development, resulting in a higher level of flexibility, as each module can be easy substituted or modified without requiring a complete re-commissioning of the code.

2. MODEL BASED APPROACH AND SOFTWARE REQUIREMENTS

Given these requirements, a new design of the system has been carried out following a model based approach [3], [2]. Such an implementation turns out to be essential when high performance and robustness are required. In particular, it has been adopted for the design of the new VS power supply and to optimize the controller parameters in different plasma operative scenarios. This approach also enabled to assess the system performance for different choices of amplifier's characteristics, in particular the maximum voltage and current limits.

Thanks to the availability of reliable linear models for the plasma magnetic behaviour [4], it was possible to validate and tune each design step, from the conceptual to the final implementation. Furthermore, exploiting the new real-time framework recently developed at JET [5], all the control algorithms designed offline in the Matlab/Simulink environment, have been implemented as plug-ins modules of the real-time application. The high modularity of the framework permitted the real-time versions of the control algorithms to be validated both offline, against a real-time model of the plant, and online, performing several JET pulses in open loop, before actually using them in closed loop.

3. HARDWARE ARCHITECTURE

The VS controller hardware and data acquisition system is based on the PICMG 3.0 Advanced Telecommunication Computing Architecture (ATCA) [9]. The system comprises 6 data acquisition cards, an x86 based controller and a Rear Transition Module (RTM). Each acquisition board provides 32 galvanically isolated, 18-bit resolution, analog to digital converters. The RTM comprehends 8 galvanically isolated analog output channels, a RS-485 based external clock and trigger input and Vertical Stabilisation, real time, control, Linux RTAIan optical digital link output. Boards are orchestrated by the x86 controller, using x1 PCI express (PCIe) links trough the ATCA backplane. Signals are acquired at 2 MSPS and, using a full-mesh protocol named Aurora [Aurora], available at the same time in the 6 boards. This feature enables the possibility to develop distributed control algorithms, directly implemented inside the board FPGAs. In parallel, acquired data is digitally filtered and decimated to 20kHz. Using the PCIe links this data is continuously sent to the central controller, permitting to implement a control system running with a closed loop cycle time of 50us.

The controller board accommodates a uATX motherboard and 3 PCIe switches that provide the bridge to the PCIe mesh in the ATCA backplane. The motherboard contains an Intel Quad-core CPU, 2GB of DIMM DDR2 DRAM and a Gigabit Ethernet interface. Benefiting from the multi-core technology an interrupt-less communication scheme, between the operating system and the boards was implemented. A driver continuously polls a memory region that is updated by one of the boards, using DMA, as soon as new samples are available in all boards. This synchronization mechanism permits to attain jitters in the order of the hundreds of nanoseconds [8]. The system runs, always providing data at this frequency, 24 hours per day.

Although ERFA can be controlled using the analogue outputs from the RTM, there are major advantages associated with the usage of a digital interface: (i) more information available about the actual state of the amplifier; (ii) reduced noise susceptibility; (iii) exact control of the set point value, instead of specifying an over voltage step. The RTM is able to digitally drive and monitor the amplifier using the Aurora protocol.

4. SOFTWARE ARCHITECTURE

In order to meet the requirements of the new VS control system the entire real-time software framework was redesigned. Although driven by VS, the new framework has been designed to address a wider set of specifications. High level of flexibility and improved debugging and testing without compromising the real-time performance were among the root design choices.

4.1. THE REAL-TIME INFRASTRUCTURE

The real-time framework is built upon a C++ multiplatform real-time library called BaseLib2. The library is built around the concept of interdependent layers where a clear boundary and entry point to operating system specificities is provided in a single layer. This enables a clean porting of the library to different platforms whenever it is required.

Strongly based on object oriented programming paradigms, among the main capabilities a large collection of advanced features can be found: garbage collection, enabling pointer free code and reference only programming; automatic creation of objects based on external configurations; runtime navigation and introspection inside objects; http interfaces and built-in servers and an high level meta configuration language for the objects. A messaging mechanism expands the capabilities of the framework to allow objects to interact with each other or with external applications, using higher level protocols. These functionalities, together with an improved standard set of operations like I/O classes, strings and networking communication, provide a flexible and exportable infrastructure. A large majority of the functionalities were optimised in order to meet the requirements of real-time systems.

4.2. MARTE: THE REAL-TIME FRAMEWORK

MARTe (Multiplatform Application Real-Time executor) [MARTe] [5] is a real-time framework for developing and executing generic real-time applications. It is based on a modular collection of components that run together in order to provide the required functionality.

The atomic element of MARTe is called Generic Application Module (GAM). Functionality if provided by developing and arranging sequence of GAMs. Each module contains 3 communication channels: one for configuration and two for input and ouput. Modules are completely independent from each other and can only communicate through these pipes. This separation between GAMs is the fundamental characteristic of the framework and it what enables the free replacement of components.

The GAMs input and output channels are connected to a shared memory location named Dynamic Data Buffer (DDB). In this database each signal is identified by unique name and each GAM declares what set of signals it is going to produce and what signals it expects as input. MARTe schedules the module execution and acts as supervisor, by running in sequence all the GAMs within a pre-defined cycle time but without any knowledge about the specific type of elaboration. Even hardware interaction is abstracted by using a specialised set of GAMs called IOGAMs. MARTe is also responsible for all the communication with the plant and experiment configuration tools. This extreme modularity approach has a significant repercussion on the debugging and testing phases. The debug and test is focused not on the entire application but separately on each module. Moreover, thanks to the functionalities offered by the BaseLib2, the possibility to navigate inside each single GAM improves the debug and reduces the testing time. The availability of the final hardware choice does not provide any constrains in what concern the operational functionalities of an application. Any test, simulation or debug can be performed on a different platform. The application will functionally behave exactly in the same way when executed in the real platform. In the absence of the data acquisition hardware, the input can be simulated with specific GAMs able to read data from files or databases. Also the output, can be stored locally, sentvia network or even live visualised.

The organisation of the modules in MARTe is programmed using a standard configuration language defined in the BaseLib2. The behaviour of each object, communication between objects and plant, operational state machine etc., is described using this syntax. This additional degree of flexibility offered by the framework enables to adapt the behaviour of an application just by changing its configuration.

The framework does not provide a standard tool for the editing of configuration files. In JET this facility is supplied by its generic user-interface named Level1. Level1 sends to MARTe a final configuration file following the VS specifications where 2 different configuration layers were implemented. The first is oriented to the commissioning phase where all the parameters of the application are accessible to the expert user. The second, is instead designed to the final operator user and only the generic behaviour of the VS is available.

The high performance required for a generic real-time application in MARTe coexists in two parallel environments. Each real-time executor in MARTe can be configured to run on a different thread responding to the higher performance where the less demanding supervisor can operate in a more relaxed environment. In the case of the VS the Linux-RTAI platform and the x86 multi-core processor technology permits to run MARTe communications and general services on a CPU and the real-time elaborations on a different and completely isolated from spurious interrupts, processor core.

During the design process an additional problem had to be addressed: due to the large amount of data acquired, large memory allocations were required in order to store information from a full experiment at maximum frequency. As the 32 bits Linux kernel only permits allocations of up to 1 Gbytes of data, an UDP message based streaming mechanism [11] was developed. A continuous data streaming between the real-time thread and an external server (again running MARTe) is able to sustain up to 56MB/sec without compromising the real-time performance.

5. SOFTWARE IMPLEMENTATION AND SIMULATION SUPPORT

The complexity of the control system and the large number of options available, require operational tests at certain key points of the development. Although some tests require working in the real environment, with actuators and plasma, in order to reduce the commissioning phase it is possible to validate up to a certain degree the response of the controller, by replacing the plasma and actuators with simulators.

A large collection of software modules (GAMs) are used in the VS system: data is transferred from the data acquisition boards and sent to an observer module, responsible for providing a vertical velocity estimation using either static or dynamic observers. A signal processing block computes derived synthetic signals, such as N-compensated magnetic, or ELM detection triggers. These are followed by the control module which implements the two control loops referred in the previous section. Finally the amplifier managers drive the amplifiers. The Vertical Amplifier Manager (VAM) module is responsible for the radial amplifier and contains the vertical kick application logic among other experimental features. The Divertor Amplifier Manager (DAM) implements custom space transformation matrices and a kick logic scheme for the divertors. This scheme is shown in Fig. 1.

Exploiting the potentialities of MARTe, a collection of modules were specifically developed to simulate the environment and a complete close loop test-bench implemented. The plasma behaviour was simulated with a state-space model GAM that receives as input the requested voltages to the actuators and produces the estimation of the plasma vertical velocity. The state space has been provided by using CREATE code[4]. Additional input for the controller are the plasma current and optionally Halpha and divertor voltages. Due to its nature, these signals are not modified by the closed loop, and are simulated with a GAM able to generate waveforms. A third module simulates the hysteresis and noise in order to recreate the hysteretic characteristic of the amplifier and add some white noise to simulate a real acquired signal. These modules are depicted in yellow in Fig. 1.

A future improvement will be to include in the chain of execution also the Shape Controller system to enhance the simulation of the overall plasma and control system behaviours[10].

6. EXPERIMENTAL RESULTS

The new vertical stabilisation system is currently being commissioned with the new radial field amplifier. During the past C26 JET experimental campaign the system ran in parallel to the old VS controller, with only one of the systems driving the amplifier at a time. The first tests consisted in comparing the output requested by the new VS controller with the output actually provided by the old VS, allowing to track until a given stage, always limited by the open loop restrains, if the requested references were accurate and if all the experimental features were being activated when requested.

After gaining some confidence on the validity of the acquired data and on the software modules, the new VS started to closed the loop in plasma during the ramp down phase, where the plasma current is smaller and the risk of actually endangering the machine very small. Also on the tail of the experiments the first closed loop kicks were applied. Finally, the first plasmas were controlled using the new VS system in different operational scenarios. A very important improvement provided by the new hardware regards the larger ADC dynamic range, which permit to a follow a magnetic signal during an ELM, while the old VS system saturated and delayed.

No software failures were ever observed during the execution of an experiment. This is due to the design choice of validating all the parameters and performing the majority of the operations before the experiment. The number of software validations that is performed in run time is kept to the absolute minimum and must be thoroughly justified.

Following a very important design decision, the system is always running, acquiring data at 20kHz and executing the software modules. Synchronisation and realtime is always guaranteed at 50 *s* even outside the experiment, as shown in Fig.2. The system has been running, 24 hours per day, for weeks without ever losing real-time. This concept is particularly relevant for long pulse duration machines, like ITER, where control and data acquisition must be assured without compromising performance during long time periods.

CONCLUSIONS

The PCU project triggered a large collection of highly innovative developments. The new ATCA based hardware provided access to state of the art technology and processing power, enabling the simultaneous acquisition of almost 200 channels at 2MHz driven by the new multi-core CPU technology.

On the software side, motivated by the need to have high reliable software that could guarantee both a safe operation of the machine and an experimental exploitation, a highly modular framework was developed. This permitted to divide the software in several atomic modules which could even interchange functionality with real hardware, enabling true simulation functionality. In order to guarantee real-time a solution based on opensource software was found by using Linux and RTAI, but going a step further and enabling advanced object oriented features in the RTAI hard real-time space.

Merging these developments a low jitter, highly modular system was obtained. It already controlled a full JET plasma, with experimental features enabled, and guarantees high performance continuous real-time operation, 24 hours per day.

ACKNOWLEDGMENTS

This work was jointly funded by the United Kingdom Engineering and Physical Sciences Research Council and by the European Communities under the contract of Association between EURATOM and UKAEA. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work was carried out within the framework of the European Fusion Development Agreement.

REFERENCES

[1]. F. Sartori et al., “The JET PCU project: an international plasma control project”, Fus. Eng. Des., ol. **83**, no. 2-3, 2008, pp. 202- 206.

[2]. M. Ariola and A. Pironti, Magnetic Control of Tokamak Plasmas. Springer, 2008.

[3]. G. Ambrosino and R. Albanese,“A survey on modeling and control of current, position and shape of axisymmetric plasmas”, IEEE Contr. Syst. Mag., vol. **26**, no. 5, 2005, pp. 76–9.

[4]. R. Albanese and F. Villone, “The linearized CREATE-L plasma response model for the control of current, position and shape in tokamaks”, Nucl. Fus., vol. **38**, no. 5, 1998, pp. 723–738

[5]. A. Neto et al., “MARTe: a multi-platform real-time framework”, Proc. of the 16th IEEE NPSS Real-Time Conference, Beijing, China, May 2009.

[6]. F. Sartori, et al The Joint European Torus, “Plasma position and shape control in the world’s largest tokamak“, IEEE Control System Magazine, April 2006 vol.**26** Number 2

[7]. F. Sartori, P. Lomas, F. Piccolo, M.K. Zedda,““Synchronous ELM Pacing at JET using the Vertical Stabilisation Controller”, 35th EPS Conference on Controlled Fusion and Plasma Physics, Crete, Greece, June 2008

[8]. A. Neto, F. Sartori, F. Piccolo, A. Barbalace, R. Vitelli, and H. Fernandes,”“Linux real-time framework for fusion devices“, Fusion Engineering and Design, vol. In Press, Corrected Proof, 2009

[9]. A.J.N. Batista, J. Sousa, and C.A. F. Varandas,““ATCA digital controller hardware for vertical stabilization of plasmas in tokamaks“, Review of Scientif Instruments, vol. **77**, no. 10, Oct 2006.

[10]. G. Ambrosino, M. Ariola, A. Pironti, F. Sartori, “A new shape controller for extremely shaped plasmas in JET“, Fusion Engineering and Design, Volumes **66-68**, 22nd Symposium on Fusion Technology, September 2003, Pages 797-802

[11]. L. Zabeo, F. Sartori, A. Neto, F. Piccolo, D. Alves, R. Vitelli, A. Barbalace, G. De Tommasi, “Continuous data recording on fast real-time systems“, this conference

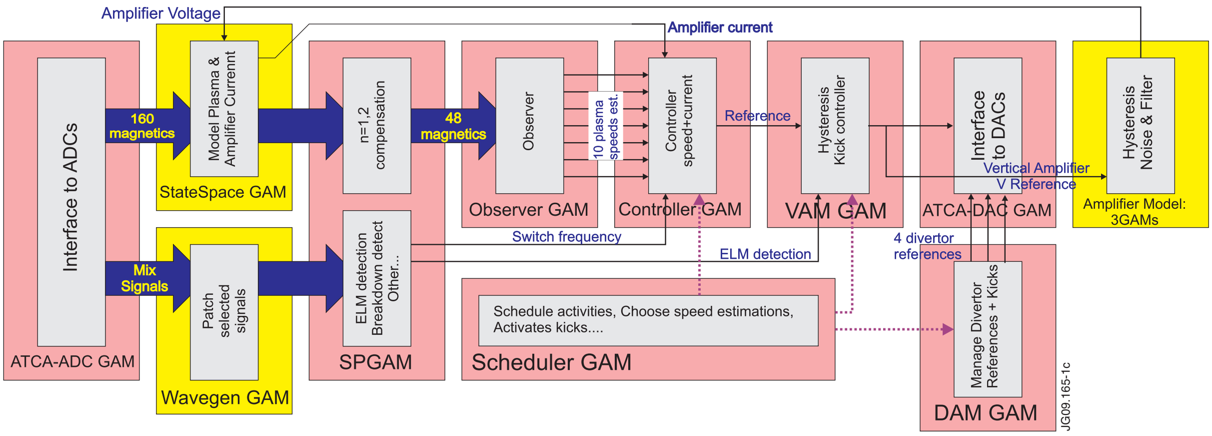

Figure 1: Controller scheme. The physics and control requirements have a direct correspondence in the software organisation. Each functional requirement was implemented in a different software module, enabling the replacement of some of the modules with simulators in a completely transparent way. In this figure a model of the amplifier, of the input data is being used to replace the actual hardware.

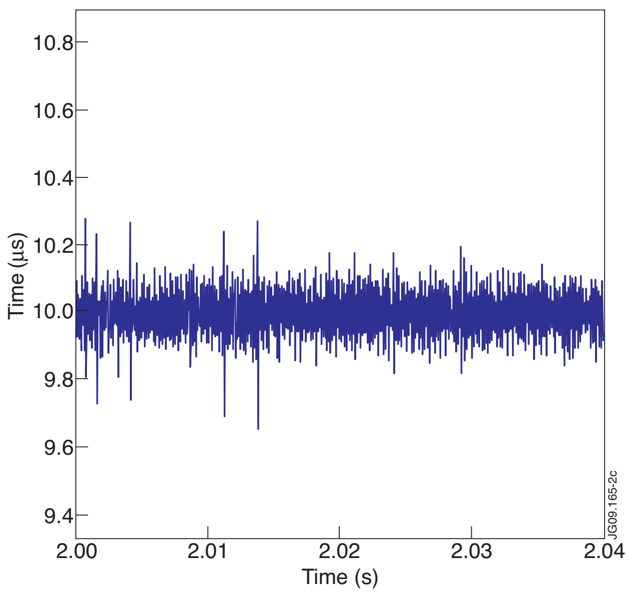

Figure 2: Jitter values in the start time for a new control loop cycle. Each cycle is started every 50µs and synchronisation is achieved within a jitter in the order of the hundreds of nanoseconds.