Real-Time Systems in Tokamak Devices
A Case Study: The JET Tokamak
Real-Time Systems in Tokamak Devices
A Case Study: The JET Tokamak

G. De Tommasi¹, D. Alves², T. Bellizio¹, R. Felton³, A. Neto², F. Sartori⁴, R. Vitelli⁵, L. Zabeo⁶, R. Albanese¹, G. Ambrosino¹, P. Lomas³
and JET EFDA contributors*


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

¹ Associazione EURATOM-ENEA-CREATE, Università di Napoli Federico II, Via Claudio 21, 80125, Napoli, Italy
² EURATOM/IST, Instituto de Plasmas e Fusão Nuclear – Laboratório Associado, 1049-001, Lisboa, Portugal
³ EURATOM-CCFE Fusion Association, Culham Science Centre, OX14 3DB, Abingdon, OXON, UK
⁴ Fusion for Energy, 08019 Barcelona, Spain
⁵ Università di Roma, Tor Vergata, Via del Politecnico 1, 00133, Roma, Italy
⁶ ITER Organization, CS 90 046, F-13067 Saint Paul-lez-Durance, Cedex, France

(24th May 2010 - 28th May 2010)
ABSTRACT.
The achievement of the required performances during the operation of large fusion experimental reactors, is strictly related to the flexibility and reliability of the real-time infrastructures. It turns out that, in tokamak reactors, the real-time infrastructure has to be designed so as to meet a number of common requirements. An overview of the real-time infrastructure currently adopted at the JET tokamak is given in this paper, focusing the attention on the solutions that have been developed for addressing these common requirements. Furthermore, three real-time systems recently deployed at JET are described as example.

1. INTRODUCTION
In a modern tokamak device, the achievement of the required performance is strictly dependent on the flexibility and reliability of the real-time systems that operate the plant during the experiment. In large experimental plants like JET [1] (see Fig.1) or ITER [2], there are some common requirements that have to be taken into account during the design of the real-time infrastructure.

In particular, on the hardware side such an infrastructure should:

- cope with the unavoidable hardware obsolescence and the need of maintenance that are expected during the life cycle of a several decades experiment, being capable to be as hardware independent as possible;
- manage the expected increase in computational requirements by being scalable;
- allow to share the resources between the processing nodes (e.g. to share the plant measurements and the outputs of each processing node).

Furthermore, on the software side it is crucial to have an architecture that:

- supports and facilitates test and validation by separating the application algorithm from the interfaces with other plant systems;
- supports model-based development to validate software modules against a plant model, so to minimize the risk in the development of complex plant control systems;
- guarantees low latency [3].

All these features contribute to reduce the time needed for the commissioning of new real-time systems on the plant.

In this paper we present an overview of the real-time infrastructure currently adopted at JET, where recent solutions have been developed for addressing the requirements listed above.

In particular, the paper is structured as follows: a brief description of the overall JET real-time infrastructure (network and processing nodes) is given in the next section. Afterwards, the real-time framework [4] recently adopted for the development of new processing hosts is briefly introduced in Section III, while the model-based approach [5] adopted to design and validate the real-time algorithms is described in Section IV. Section V presents three examples of real-time
system recently upgraded at JET, namely the *Vertical Stabilization System*, the *Error Field Correction Coils Controller* and the *BetaLi* diagnostic. Eventually some conclusive remarks are given.

### 2. OVERVIEW OF THE JET REAL-TIME INFRASTRUCTURE

This section briefly introduce the JET real-time infrastructure. At the end of the section, the real-time network currently deployed at JET is described.

The main plasma control at JET involves:

- Plasma Magnetic Diagnostics;
- Plasma Position and Current Control (PPCC);
- Plasma Density Diagnostics;
- Plasma Density Control (PDF);
- Gas Introduction plant.

Furthermore, some experiments (about 1 in 3) require special controls: e.g. in hybrid scenarios [6] is required to achieve repeatable $\beta_p$. These involve a wide range of plasma diagnostics, a programmable controller and heating and gas systems. So, over the last ten years, many diagnostics have been upgraded to provide real-time, physics data, every few ms. The dedicated controllers PPCC, PDF and a general purpose programmable controller called RTCC have been developed and exploited. Moreover, the plant systems for Neutral Beam Injection (NBI), Ion Cyclotron Heating (ICRH), Lower Hybrid Current Drive (LHCD) have been modified to switch from pre-programmed schedules to real-time tracking of power requests. Each of the systems are large and complex, with their own engineering and operations staff. The signal processing in many systems is itself quite complex. This distributing the calculations over the separate systems has benefits for basic functionality and capability, minimising impact on other system of internal changes in one. That is, each system is designed to have

i) high cohesion - any one system has everything it needs in its own domain and does not overlap other domains;

ii) low coupling - any one system has the inputs and outputs essential for the global operation.

The present real-time network was setup about 10 years ago, after an investigation of the available technology, i.e. analogue, shared ethernet, switched ATM and reflective-memory. It was decided to resort to ATM/AAL5 communications on 155MHz fibre-optic. Each system would send application-specific datagrams into the network, known as the *Real-Time Data Network (RTDN)*. For cross-platform inter-operability, the datagrams are fixed-size and and fixed structure, based on 32-bit integer or floating point fields, with a simple header (sample number, sample time) and trailer (datagram version - for verification). The network switch distributes the datagrams to whichever system needs the information. Currently there are more than 30 systems, 40 datagram-types, and a total of more than 500 signals. The systems send datagrams even between JET pulses, and the receivers can check the data-stream is flowing. The systems can even synchronise actions on arrival.
of datagrams. The ATM network provides one-to-many connections, and also isolates sources technically from the destinations (the source does not make connection with destination). With present-day operating systems, ATM NIC drivers, etc., it has been found that the typical latency is of the order of 100\(\mu\)s, which is sufficient for our fastest cycle time of 2 ms\(^1\). Many diagnostics and plant systems need only operate at 10’s ms. The connections, configured in the ATM switch, and the datagram structures are centrally managed to ensure global coordination [7].

Over the last ten years, the ATM network has been upgraded from one 16-port workgroup switch to a 4\(\times\)16 port fabric and a 1\(\times\)16 fabric in enterprise grade chassis’s. The network itself is phenomenally reliable. During commissioning, only few problems with the sources and destinations have been experienced, but these were easily solved.

Just as important as the communications, is the human-machine interfaces. Over the last ten years, a general purpose pulse-schedule-editor has been developed, which is configured for the specific systems (Diagnostic, Heat, etc.). It provides an invaluable common-look-and-feel, and schedule archiving and retrieval, in order to have an accessible record of operations of the various real-time-systems.

JET is now installing an ITER-Like wall. The Be/W tiles are more fragile than the CFC tiles, and a new 1The fastest control system that uses ATM signals is the plasma shape controller (see Section IV-A). It is worth to notice that the fastest cycle time at JET is the one of the Vertical Stabilization system (see Section V-A), which is 50\(\mu\)s. However this system does not signals from ATM to control the plasma vertical velocity. real-time protection-by-active-control system is being developed. This will involve:

- temperature measurements from thermocouples, pyrometers and ir-sensitive cameras;
- analysis of temperatures and energies;
- configurable conditions and actions;
- central coordination of PPCC, PDF and the heating systems.

The RTDN will carry the physics data, the alarms and directives (for the plant systems). New systems will added, each with a specific purpose, continuing the philosphy introduced ten years ago.

3. THE MARTE REAL-TIME FRAMEWORK
The Multi-threaded Application Real-Time executor (MARTe [4]) is built over a multi-platform library, i.e. it permits the execution of the same code on different operating systems\(^2\) (as it will be shown in Section V), and it provides the high level interfaces with hardware, external configuration programs and user interfaces, assuring at the same time hard real-time performances. Within the

\(^1\)The fastest control system that uses ATM signals is the plasma shape controller (see Section IV-A). It is worth to notice that the fastest cycle time at JET is the one of the Vertical Stabilization system (see Section V-A), which is 50\(\mu\)s. However this system does not signals from ATM to control the plasma vertical velocity.

\(^2\)The current version of MARTe can be deployed on Windows®, Linux, Linux/RTAI, VxWorks® and Sun Solaris®.
MARTe environment, the end users are required to define and implement algorithms inside a well defined software block named *Generic Application Module* (GAM), which is executed by the real-time scheduler.

Being portable and modular, the MARTe framework permits:

- to standardize the development of real-time applications;
- to increase the code reusability;
- to separate the user application from the software required to interface with the plant infrastructure;
- to reduce the time needed for commissioning.

The separation between application and infrastructure software allows the scientists (which are the process expert) to abstract from the plant interfaces when developing their applications. In particular, scientists can focus their attention on the design of algorithms and required GAMs. This feature increases code reusability. Furthermore, being portable MARTe is *hardware independent*; this feature allows to manage the unavoidable hardware obsolescence that is expected during the life cycle of a 20 years experiment such as ITER.

### 4. MODEL-BASED DESIGN AND VALIDATION

In this section we introduce the model-based design and validation approach. As an example of project where this technique has been successfully adopted, the case of the eXtreme Shape Controller (XSC [8]) is discussed at the end of the section.

In addition to the adoption of a standard framework for the development of real-time systems, model-based design and validation is another effective approach to reduce the time needed for commissioning a new system on the plant. Such an approach relies on the availability of accurate plant models that can be used to develop and test new real-time algorithms. This modelling activity may appear costly in terms of people and time. However, it is essential when developing new systems on a plant where the priority is the scientific exploitation. As a matter of fact, reliable plasma models are available for the JET tokamak [9], [10], and they have extensively been used to design the two main plasma magnetic control systems, i.e. the plasma shape controller and the vertical stabilization system [11].

Modelling helps the designers to:

- design the algorithms;
- make performance analyses;
- validate real-time implementation of the systems;
- perform offline analyses to forecast experimental behaviour.

In particular, when developing a new control system, detailed plant models are needed to design control algorithms and to assess control system performance. This task is usually performed into an
**offline environment such as Matlab/Simulink®.** Furthermore, real-time version of even simplified plant models can be effectively used to perform **offline** debug of the software implementation, and for the **offline** validation of a control system with or without hardware-in-the-loop\(^3\) (an example of this technique applied to the ITER Central Safety System is described in [12]).

Thanks to its modularity and to the separation between the application and the infrastructure software, two approaches for model-based validation are possible within the MARTe framework:

1) the real-time control application can be validated in the **offline** simulation environment;
2) the plant model can be added in the real-time system (as it will be shown in Section V-A).

### 4.1. THE DESIGN OF THE EXTREME SHAPE CONTROLLER

The XSC [13] has been developed to control the whole plasma boundary controlling up to 32 geometrical descriptors, thus it can be used to obtain high performance plasmas, controlling the shape in the presence of disturbances. It has been successfully validated on the JET tokamak in 2003, and it is now used during the experimental campaigns at JET, particularly to control ITER-like plasmas.

The design of the XSC has been carried out adopting the design approach described so far. Indeed, its design is based on a linearized model of the plasma and Poloidal Filed (PF) coils system behavior, thus the controller parameters are different for different operative scenarios.

When preparing the XSC for a new scenario, first a model of the plasma and of the surrounding coils must be retrieved, then the controller design and validation can be started, in a typical iterative procedure.

To automate the controller design and validation phases, a set of Matlab/Simulink graphic applications, called XSC Tools [14], has been developed. Using these tools, new XSC scenarios can be easily prepared and validated via closed-loop simulations. Furthermore, once the controller has been validated, all its parameter are stored into a text file, called (**configuration file**), that can be directly loaded to setup the real-time controller running on the plant.

Thanks to the model-based design approach, XSC functionalities have been easily extended in order to include strike-points sweeping [15] and plasma boundary flux-control [16].

### 5. JET REAL-TIME SYSTEMS: THREE EXAMPLES

In this section three real-time systems recently upgraded at JET are presented. In particular we will describe:

- the **Vertical Stabilization System** that permits to stabilize elongated plasmas;
- the **Error Field Correction Coils Controller** allows for the reduction of dominant error field components in the plasma boundary;

\(^3\)**Hardware-in-the-loop (HIL) simulation is a technique that is used in the development and test of complex real-time embedded systems. HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform. This is achieved by adding a mathematical representation of plant.**
• the BetaLi diagnostic system which estimates in real-time both the plasma poloidal beta $\beta_p$ and internal inductance $l_i$.

5.1. VERTICAL STABILIZATION SYSTEM

The Vertical Stabilization (VS) system is one of the most critical systems in a tokamak, as it is responsible for guaranteeing zero plasma vertical velocity (on average). Indeed, the VS controller is designed to vertically stabilize the plasma so that the shape controller can successfully control the plasma position and shape. The feedback signal is the plasma vertical speed. At JET, the actuator is the Radial Field Amplifier, which feeds the control coils [17].

Scenarios with highly elongated plasmas in presence of large Edge Localised Mode (ELM [18]) perturbations are envisaged to achieve better fusion performance in tokamaks. In these extreme scenarios a general purpose controller, which is robust enough to work satisfactorily under any envisaged operational scenario, cannot meet the requirements, since it is incapable of pushing the performance to the maximum. As stated in Section IV, in order to achieve the desired performance, it is a common practice to rely on a model–based design approach.

In particular, for each plasma scenario, it is envisaged that the JET VS system could potentially use different estimations of the plasma vertical velocity, as well as different adaptive algorithms for the controller gains, in order to optimize the system behavior.

The architecture proposed for the VS system is similar to the one adopted for the XSC at JET ([8], [13]). Indeed, it permits to cope with different scenarios during the same experiment in a simple manner. However, since the controllers are heavily optimized, a safety logic capable of switching to the general purpose controller in case of unexpected events must also be present, in order to get a safe termination of the experiment.

Since control algorithms are usually developed in a modeling and simulation environment (e.g. Matlab/Simulink®), another requirement for the new VS software architecture concerns the possibility to check and validate the whole real-time code (including both the control algorithm and the auxiliary code, i.e. communication interfaces with other systems, data acquisition, etc.) before testing it on the plant. To perform this offline validation, real-time computational models of the plant based on detailed plasma linearized models ([9], [10]) are needed.

In order to take into account all the functional requirements, the new VS system has been developed exploiting the flexibility of the MARTe framework described in Section III. The JET VS system has been implemented under the Real Time Application Interface (RTAI)/Linux operating system [19]. Thanks to this choice it has been possible to exploit the multi-processor ATCA4 based hardware architecture [20].

A functional block diagram of the overall VS software architecture is depicted in Fig.2. The peculiar modules that gives the required level of flexibility to the VS system are: the Observer GAM, the Controller GAM, and the Vertical Amplifier Manager GAM (VAM GAM). A brief description of these modules is given hereafter.

4 Advanced Telecommunication Computing Architecture
The **Observer GAM** computes up to ten different estimations of the plasma vertical velocity\(^5\). All these different estimations are available during the whole experiment. As a matter of fact, each estimation is computed by a generic dynamic linear system.

All the plasma velocity estimations, together with the power supply current and switching frequency, are sent as inputs to the **Controller GAM**, which contains four different control algorithms\(^6\), and computes the voltage reference to the power supply.

For what concerns the planning of the experiment, every JET discharge is logically divided into a number of *time windows*. In each time window, all control algorithms receive all the plasma velocity estimations.

The selection of the controlled variable is made on the basis of the signals provided by the **Scheduler GAM**. Although all the plasma velocity estimations are always computed, in each time window four out of forty possible paths are available. Such an architectural choice can be effectively exploited to minimize bumps during control algorithm transfer. Indeed, to avoid control bumps the desired controlled variable can be selected as input to an inactive controller one time window before activating it.

Furthermore, in each time window, the **Scheduler GAM** instructs the **VAM GAM** about which voltage request generated by the controllers should be sent to the power supply. Based on the signals received from the Scheduler GAM, the VAM GAM can also perform *kicks*, which are voltage pulses of a given length and amplitude, used, for example, to perform ELM pacing [21].

The three GAMs depicted in yellow have been specifically developed to simulate the plant. This feature allows the user to implement a complete closed–loop test–bench. In particular, they permit to simulate the plant behavior, the hysteretic characteristic of power supply, and to add some noise to the simulated signals.

These three GAMs are enabled by the user only when performing offline validation and are disabled when the VS runs on the real plant.

In particular this closed-loop simulator has been used to study the operational limit of the VS in terms of maximum controllable ELM, which are localized plasma instabilities. They manifest themselves as strong magnetic perturbations associated with a burst of radiation and a loss of particles and energy from the plasma periphery. The perturbation affects the magnetic fields creating a strong variation in the plasma speed measurement. In particular, due to an ELM occurrence, the VS system experiences a rapid increase of plasma speed followed by a rapid inversion and a slower decay. This causes an undesired firing of the power amplifier resulting in a vertical excursion of the plasma which can lead, in some cases, to a disruption, i.e. to a loss of vertical control. Therefore the VS system and the shape controller must be able to counteract these disturbances. By modelling the effects of ELMs on plasma position and shape it is possible to simulate the behaviour of the VS system when an ELM event occurs, in order to explore the operational limit of the system.

As for the computational delay introduced by the software, all the modules are executed in about 40\(\mu\)s with a jitter that is less than 1\(\mu\)s, and is mainly due to driver and hardware of the I/O boards.

\(^{5}\)It should be noted that ten different estimations have been supposed to be sufficient for the JET application. However this number can be easily configured.

\(^{6}\)As for the Observer GAM, this number can be easily upgraded.
The exact time at which the reference is sent to the amplifier varies accordingly to the enabled features in the different parts of the experiment, but it is always bounded by the 50µs figure. The synchronization mechanism guarantees that the software is always ready and waiting for a new set of samples, with an error in the order of the hundreds of nanoseconds.

5.2. ERROR FIELD CORRECTION COILS CONTROLLER

The Error Field Correction Coils (EFCCs) are a set of 4 non-axisymmetric coils used to reduce the dominant error field components in the plasma boundary, and in ELM pacing and mitigation experiments. A set of two Poloidal Radial Field Amplifiers (PRFAs) are responsible for driving the currents in the EFCCs. The system can be configured in two distinct modes of operation: either one PRFA drives all four EFCCs or each PRFA drives two EFCCs. These configurations are directly related to the induced magnetic field topology. Performance limitations of the PRFAs restrict the effective bandwidth of the system. For this reason a new controller with the ability to improve the PRFA response has been designed and implemented using the MARTe framework.

The new control algorithm is based on a theoretical PRFA model and on an anticipation-based approach of the current reference waveform. This anticipation is adapted during the experiment in order to maximise the performance and to mitigate eventual disturbances due to model inaccuracies. This algorithm, together with a proportional-integral feedback controller guarantees, at least in simulation, a significant boost of the system performance.

The controller has been implemented using two different modules: the FeedForward GAM, which incorporates the amplifier model and implements the adaptation logic generating the feedforward signal for the amplifier and the PID GAM, which implements the standard PID controller.

Unfortunately, during the past JET campaign, the controller application was operated without the feedforward module of the algorithm, retaining only the usual PID feedback contribution. It is worth noticing however, that the modular-oriented design of the system along with MARTe’s framework implementation philosophy greatly reduces the burden of managing functionality addition or removal.

This controller is hosted in a VME crate holding a VME Programmable Logic Service (VPLS) for distributing JET’s central timing, a 64 MB and 400 MHz Motorola® MVME5100 PowerPC board for performing the control algorithms, a Pentland R MPV956 analog i/o board and a Pentland® MPV922 digital i/o board.

As shown in figure 3, this system has been used successfully to perform the required current control in the EFCCs under not too demanding circumstances, namely, current waveform frequencies below 10Hz.

Due to lack of experimental time, the system has been tested using only the PID GAM during the 2009 JET experimental campaign (see Figure 3). The complete control system will be tested on the machine after the restart of the tokamak, by simply inserting the FeedForward GAM in the execution chain of the system.
5.2. BETALI DIAGNOSTIC

The plasma conditions can be characterized by a set of parameters that can be extracted by means of combination of diagnostic data. Magnetic measurements, plasma temperature, plasma density, together with information on the additional power, permit to identify the plasma status [22].

If available in real-time, this information can be exploited during the experiment to achieve improved plasma regimes. As an example, recently the control of the poloidal beta $\beta_p$ by using the Neutral Beam Injection (NBI) has been successfully tested at JET [23].

At the JET tokamak, the BetaLi real-time code [24] is able to provide a large set of plasma parameters, and it is routinely used for many different real-time feedback controls. Given the increased interest in the plasma feedback control, the software architecture of this code has been recently redesigned, in order to improve its performance and precision.

A modular approach has been adopted to design and implement the present version of the BetaLi system. In particular, the code has been split in five different modules (see Fig. 4). The proposed structure separates the common calculation (which are needed to compute any plasma parameter) from the evaluation of the specific plasma parameters. In particular, the following modules have been implemented:

- **Magnetic topology GAM**, which identifies the flux topology needed for the plasma boundary reconstruction and the magnetic flux/field determination [25]. This module is the most demanding in terms of computational effort. Indeed, the identification of the magnetic topology is performed computing the local minimum of the flux distribution along a given set of segments (called GAPs). Hence the computational effort increases with the number of desired boundary points. Typically the code uses 100 segments for the boundary reconstructions.

- **Plasma geometry GAM**, starting from the flux topology, this module provides the plasma geometry with the associated magnetic information. For example, this module calculates the plasma perimeter, area and volume, together with other plasma shape parameters (i.e. elongation, triangularity, etc.).

- **Shafranov integrals/moments GAM** which computes the Shafranov integrals and moments starting from the plasma boundary information [26].

- **Magnetic compensation GAM** which is an independent module introduced for calculating compensated magnetic data [27].

- **Plasma parameters GAM** is the module that calculates all the plasma parameters starting from the output of the previous modules plus additional inputs. The outputs coming from the other modules are used to compute the desired parameters like plasma $\beta_p$, internal inductance $l_i$ or plasma energies.

The implemented plasma parameterisation has been tuned on a specific set of plasma scenarios, and using a specific set of equations. However, more sophisticated and accurate methods could provide improved precision in the plasma parameters identification.
For example, the boundary reconstruction module could be replaced with an improved version, which is able to provide boundary data from the very early phase of the discharge, giving the possibility to turn on the feedback control during the start-up phase (which is usually carried out in open-loop). Moreover, if new diagnostics will become available they can be easily included to extend the set of plasma parameters computed by the BetaLi system.

The system is presently running with a time resolution of $2\text{ms}$ on a standard PC running WindowsNT®. This time resolution is sufficient for the type of the feedback control presently operating at JET, but it could be easily increased if required.

**CONCLUSIONS**

In this paper the real-time infrastructure currently adopted at JET tokamak has been presented. Furthermore, the model-based design approach has been introduced. Such an approach has been successfully adopted to develop the main JET real-time control systems. As an example, three real-time systems recently deployed have been described.

**REFERENCES**


Figure 1: View of the JET vacuum chamber.

Figure 2: Block diagram of the overall JET Vertical Stabilization system software architecture.

Figure 3: JET Pulse No: 78954. PRFA current behaviour and system cycle time.

Figure 4: Block diagram of the BetaLi software architecture.