EFDA–JET–PR(09)26

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

The Software Architecture of the New Vertical Stabilization System for the JET Tokamak

The Software Architecture of the New Vertical Stabilization System for the JET Tokamak

T. Bellizio[1], G. De Tommasi[1], P. Mc Cullen[2], A. Neto[3], F. Piccolo[2], F. Sartori[4],
R. Vitelli[5], L. Zabeo[2] 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-CCFE Fusion Association, Culham Science Centre, OX14 3DB, Abingdon, OXON, UK*
[3]*Assoçião EURATOM/IST, Instituto de Plasmas e Fuoão Nuclear, Av. Rovisco Pais, 1049-001 Lisboa, Portugal.*
[4]*Fusion for Energy, 08019 Barcelona, Spain.*
[5]*Dipartimento di Informatic, Sistemi e Produzione, Università di Roma, Tor Vergata,*
Via del Politecnico 1, 00133, Roma, 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
IEEE Transactions on Nuclear Science

ABSTRACT.

The need to achieve better and better performance in modern tokamak operations has led to a further development of the plasma shape and position control systems. In particular, extremely elongated plasmas, with high growth rate, are envisaged to reach the required performance for ignition. This request for better performance from the experimentalists side has motivated the development of the new Vertical Stabilization (VS) system at the JET tokamak, which has been proposed within the Plasma Control Upgrade project. Indeed, the main aim of the project is to enhance the capabilities of the plasma vertical position control system, so as to operate with very highly elongated plasma in presence of large perturbations, i.e. large Edge Localized Modes (ELMs). This paper focuses on the new software architecture of the VS system, which relies on a highly configurable real-time framework. Thanks to its flexibility the new VS executes different control algorithms and it schedules the one which maximizes the performance in each plasma phase.

1. INTRODUCTION

The need of achieving always better performance in present and future tokamak devices [1], has pushed plasma control to gain more and more importance in tokamak engineering (see the recent book [2]). High performance in tokamaks are achieved by plasmas with elongated poloidal crosssection[1] (see Figure 1) and magnetic X-point [3, Tutorial 7]. Since such elongated plasmas are vertically unstable [4, Section 1.1.5], position control on a very fast time-scale is clearly an essential feature of all machines. Furthermore, in order to obtain best performance out of a device, it is always necessary to maximize the plasma volume within the available space; hence, the ability to control the shape of the plasma while ensuring good clearance between plasma and the facing components is an essential feature of any plasma shape and position control system.

In this context, the Plasma Control Upgrade (PCU) project [5] has been proposed and developed with the main aim of increasing the capabilities of the Vertical Stabilization (VS) system of the JET tokamak [6], [7]. The VS system is one of the most critical systems in a tokamak, as it is responsible for guaranteeing zero plasma vertical velocity (on avarage). 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. The actuator is the Radial Field Amplifier (RFA) circuit shown in Figure 2. To avoid saturating the current in the circuit, the VS controller also implements a current control loop.

During the experiment, some plasma magnetic events act as disturbances for the VS control system. In particular, small and frequent Edge Localized Modes (ELMs [8]) perturbations[2] can cause the overheating of the RFA power supply, while giant ELMs create a large disturbance that puts the plasma far out of the equilibrium, sometimes causing a plasma disruption.

The control system upgrade at JET will enhance the ability of the VS system to recover from large ELM perturbations, specially for the case of plasmas with high elongation, i.e. plasmas with large growth rate. The design of the new system has been carried out following a model based approach [9], [2], which turns out to be essential when high performance and robustness are required. In particular, such an approach has been adopted for:

- the design of the new power supply for the RFA circuit, called *Enhanced Radial Field Amplifier* (ERFA)[3], to assess the system performance for different choices of the amplifier's maximum voltage and current [10];
- the assessment of the best choice for the turns setup of the poloidal field coils used for vertical stabilization;
- the design of the new VS control algorithm, to optimize the controller parameters for the different operative scenarios provided by the physicists.

Thanks to the availability of reliable linear models for the plasma magnetic behavior [11], [12], a validation phase has been carried out for each design step, from conceptual design to implementation.

In this paper we focus our attention on the software architecture of the new VS control system, which is based on MARTe, the real-time framework recently developed at JET [13], [14]. Indeed, the new VS system represents the first *MARTe based* control system which has been successfully developed and deployed at JET.

Exploiting the MARTe environment, all the control algorithms, which have been previously designed and validated in the Matlab/Simulink R° environment, have been implemented as plug-ins of the whole real-time application. Hence, in order to change (to add) a control algorithm it is sufficient to modify a (to write a new) plug-in. Furthermore, by using MARTe, the real-time version of the control algorithms have been validated both offline, against a real-time prototype of the plant, and online, performing several JET pulses in open loop, before using them in closed loop.

The paper is structured as follows. The next section summarizes the main motivations for the JET VS en hancement in connection with the objectives of the PCU project. Section III gives the general overview of the control system architecture, while Section IV describes more in details the main software components. The user interface is briefly introduced in Section V, since it is strongly related with the VS software architecture. Eventually, Section VI presents the test and validation approach adopted during the commissioning, together with some experimental results.

2. MOTIVATIONS

The main motivations that have driven the design and development of the new Vertical Stabilization system at JET are recalled in this section. In order to achieve better fusion performance in tokamaks, a number of high performance scenarios with highly elongated plasmas in presence of large ELM perturbations are envisaged. In these *extreme* scenarios a *general purpose controller*[4] cannot guarantee the requirements. For this reason, in order to push the performance up to the desired level, it is usual to rely on a model based design approach ([2], [15]), which assures the needed control performance. 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, so as to optimize the system behavior. These optimized controllers will be scheduled by using either a time-driven or an event-driven policy. In the former case the whole experimental pulse is divided into a number of time segments, which are called *time windows*, wherein different control algorithm can be chosen (more details will be given in Section V). The proposed architecture is similar to the one adopted for the eXtreme Shape Controller at JET ([16], [17], [18]), and it permits to face with different scenarios during the same experiment in a simple manner. However, since the controllers are optimized, there must be a safety logic that, in case of unexpected dangerous events, switches to the general purpose controller, so as to get a safe termination of the experiment.

Furthermore, new control algorithms are usually developed in a modeling and simulation environment (e.g. Matlab/Simulink R°), while are then implemented on a real-time target. 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 interface with other systems, data acquisition, etc.) before test it on the plant. Hence, it is strongly recommended to use real-time prototype of the plant, based on detailed plasma linearized models ([11], [12]), to perform this offline validation.

Given the functional requirements summarized above, it turns out that the adoption of a flexible and modular software architecture is mandatory for the VS implementation, in order to successfully cope with the requirements given within the JET PCU project. Indeed, the existing VS system [19], based on 4 Texas Instruments R® DSPs (TMS320C40), was not flexible enough to satisfy the requirements. As an example, the present control system has been used during the preliminary analysis phase of the PCU project, so as to carry out some dedicated experiment aimed to confirm the simulation results. In order to avoid a complete rewriting of a substantial part of the software, during these tests the modification of the control algorithm have been applied as *patches* to the normal control mode, since it was not possible to isolate the control algorithm from the remaining part of the software. Moreover, given the limitations of the present architecture, anytime a new functionality was required its implementation was not straightforward, mostly due to this lack of separation between the control algorithm and the auxiliary software, which is always needed in a real world control application.

Within the JET PCU project it was planned to design and develop a new software architecture for the VS, considering the flexibility and modularity issues described so far. Furthermore, it is important to note that a modular architecture permits also to minimize the unavoidable interactions between software modules. Indeed, given an higher degree of *separation* it is possible to dramatically reduce the chance of errors when a single module is changed.

In order to take into account all the functional requirements the new JET VS system has been developed exploiting the MARTe framework [14]. MARTe is built over a multi-platform library, i.e. it permits the execution of the same code in different operating systems, 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 MARTe environment, the end users are required to define and implement algorithms inside a well defined block of software, named Generic Application Module (GAM), that is executed by the real-time scheduler. The JET VS system has been implemented by using MARTe under the Real Time Application Interface (RTAI)/Linux operating system [13]. Thanks to this choice it has been possible to exploits the multi-processor ATCA[5] based hardware architecture [20].

3. GENERAL OVERVIEW

The VS software has been developed by using the MARTe framework. Within MARTe, the user application is a collection of Generic Application Modules (GAMs), which are specified by users and executed by a real-time micro-scheduler. In particular, the user specifies both the GAM inputs and the outputs, as well as a number of parameters. The functionalities performed by each GAM are then specified by implementing a standard Application Programming Interface (API), which has been designed taking into account all the peculiar needs when operating in a tokamak reactor. More details about MARTe and GAMs can be found in [14].

A functional block of the overall VS system software architecture is depicted in Figure 3, where all the GAMs that form the VS system are shown, together with the main signals exchanged among them. Inputs to the VS cubicle from other JET subsystems are acquired via ADCs, which are managed by the *ATCA-ADC GAM*. The acquired measurements includes all the magnetics and some additional inputs, such as plasma current and the current and the temperature of the power supply.

Once the input signals have been acquired, they are available to all other GAMs. In particular the measurements are sent to the *Signal Processing GAM* (SPGAM) which computes the references waveforms for the control loops, starting from what it is specified in the user interface (see Section V). Furthermore this module computes other commonly used signals such as the compensated magnetic measurements [21] and the switching frequency of the power supply.

The compensated magnetic measurements are then sent to the *Observer GAM*, which computes up to ten different estimations of the plasma vertical velocity[6]. These different estimations are available during the whole experiment. As a matter of fact, each estimation is computed by a generic dynamic linear system which are specified by the user for each experiment, as described in Section IV-A.

All the plasma velocity estimations, together with the power supply current and switching frequency, are sent as inputs to the *Controller GAM*, which is a container four (this number can be easily configured as for the ObserverGAM) different control algorithms, which computes the voltage reference to the power supply during the whole experiment.

For what concerns the planning of the experiment, every JET discharge is logically divided into a number of time windows. In each time window, a specific estimation of the plasma velocity is fed to each control algorithm. The selection of the controlled variable is made on the basis of the signals provided by the *Scheduler GAM*.

Furthermore, in each time window, the‘*Scheduler GAM*‘instructs the‘*Vertical Amplifier Manager GAM* (VAMGAM) about which voltage requests generated by the controllers should be sent to the power supply. Based on the signals received from the *Scheduler GAM*, the VAMGAM can also sum a dither to the voltage request, delay the request, or discard the voltage request, giving the control to the *Kick Controller*, which implements the kick logic (for more details see Section IV-C). In the new VS system there is also the possibility to use the divertor coils for vertical stabilization purposes.

In particular, the *Divertor Amplifiers Manager GAM* (DAMGAM) sends the voltage request to the divertor power supplies. All the request for the actuators (RFA and divertors power supplies) are sent to the DAC by the ATCA-DAC *GAM*.

Moreover, the three GAMs depicted in yellow have been specifically developed to simulate the plant in order to implement a complete close loop test-bench. The plant behaviour is 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 [11], [12]. Additional inputs for the controller are the plasma current and optionally Halpha and divertor voltages. Due to their nature, these signals are not modified by the closed loop, and are simulated with a GAM able to generate waveforms. The 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. Exploiting the potentialities of MARTe, these three GAMs are enabled by the user only when performing offline validation, while are disabled when the VS is run on the real plant.

4. MAIN SOFTWARE COMPONENTS

The main software modules previously introduced are described in details in this section. In particular more details about the Observer, the Controller, the VAM and the DAM GAMs are given.

4.1. OBSERVER GAM

The architecture of the new JET VS system has been conceived to operate in advanced plasma scenario, where different estimations of the plasma vertical velocity must be available in order to optimize system performance. For these reason, the Observer GAM has been designed as a *container* of ten different observers which computes different estimations of the plasma vertical velocity.

An observer receives as input a set of measurements and a transformation matrix. The resulting outputs can be used as inputs for other observers, in a daisy chain design, enabling the eventual reuse and optimization of some calculations. At the end of the production chain, a special observer produces a last signal, which is the result of a configurable linear combination of the output of all the observers.

The observer computational interface can be extended and specialized in order to meet and model specific requirements, loosing in flexibility but leveraging configuration and functionality. One example is the state space model observer [22], where instead of specifying one anonymous matrix, the end-user is expected to provide the matrices with a direct correspondence to the observer dynamic model.

4.2. CONTROLLER GAM

As for the Observer GAM, the Controller GAM has been conceived as a container of four different control algorithms which are available during whole pulse. Thanks to this choice, it is possible to meet the requirements in terms of disturbances rejection and thermal losses in the RFA circuit, by selecting the *optimal* controller in each phase of the pulse. Furthermore this architectural choice permits to safely validate new control algorithms on the plant by running them in open-loop during the experiments.

There are a number of inputs that are common to all the control algorithms. In particular each control algorithm receives as inputs all the plasma vertical velocity estimations computed by the Observer GAM, together with the current in the RFA circuit and the current reference waveforms. Moreover, each algorithm can have its own input signals. The selection of the plasma vertical velocity to be used for the control is made on the basis of the scheduling signal provided by the *Scheduler GAM*.

The control algorithms can implement any linear or nonlinear control algorithm, provided that the computational effort is achievable. However each control algorithm must satisfy two basic requirements:

- control of the plasma vertical velocity, in order to achieve vertical stabilization;
- control the current in the RFA circuit, so as to avoid current saturation and to reduce the thermal losses in the coil.

Figure 4 shows the basic structure of a control algorithm contained in the ControlGAM.

4.3. VERTICAL AMPLIFIER MANAGER GAM

The VAMGAM selects the desired controller outputs, on the basis of the scheduling signals. Before sending it to the ERFA, the selected voltage request could be further processed by the following VAMGAM components: the *Dither* module, the *Delay* module, the *Kicks* module and the *Relay Characteristic*.

4.3.1 Dither:

The *Dither* component adds a sawtooth waveform to the selected voltage request. This feature is used to reduce the effect of the voltage quantization. Indeed ERFA is composed of four units each rated 3 kV, 5 kA, which can be configured to deliver 12 kV, 5 kA [10].

4.3.2.Delay:

The *Delay* module is used to delay the voltage request by a given number of time samples. The resulting delay introduced in the system is used to estimate the stability margins [23] during dedicated tests.

4.3.3 Kicks:

The *Kicks* module is the most important component of the VAMGAM, since it is the most innovative.

It implements all the various types of kicks, which are voltage pulse of a given length and amplitude, and which can be specified by using the VS graphical user interface (see Section V-A). A *kick logic* is specified by using a *kick waveform* and a *kick type*. The former describes the voltage waveform to be applied by the kick component, while the latter decides when to apply the waveform itself.

A kick waveform is defined as a sequence of time windows, each one specifying the following parameters:

- the *duration* (in seconds) of the time window;
- the *amplitude* (in volts) of the window;
- the *kick modality* which can be set either equal to
  - *ON*, so as to apply in feed-forward the amplitude of the current time window, substituting the value calculated by the controller;
  - *OFF*, so as to ignore the amplitude specified and to turn off the kick logic in the current time window;
  - *ADD*, so as to add the amplitude specified to the value calculated by the controller); *2'* the *time*, which can set equal to
  - *DEFAULT*, so as to use as duration of the current time window the value specified by the *length* parameter;
  - *WAVEFORM*, so as to use as duration of the current time window the values specified by a given waveform.

By using the kick waveform and the kick type parameter a very high level of customization is achieved, allowing the user to specify:

- *timed kicks* which are kicks applied at a precise time during the experiment, and which are used to simulate Vertical Displacement Events (VDEs) and to perform halo currents studies [24];
- *periodic kicks*, used for ELM pacing [25]; Figure 5 shows an example of 12kV negative kicks;
- Hα kicks which are triggered at the occurrence of an ELM, and which are used to switch off the controller during an ELM phase;
- *saturation kicks*, which are used as protection system when the amplifier current reaches the safety threshold, i.e. when the current is close to the saturation. If this is the case, then there is the risk of a disruption if a big disturbance occurs. For this reason the current is moved far from saturation by using voltage kicks.

4.3.4. Relay Characteristic:

the *Relay Characteristic* module implements the same variable hysteresis logic of the power supply ensuring that the correct voltage is applied by the amplifier even in presence of noise or not perfect calibrated DACs. Furthermore this block, if required, generates the digital word used to command the amplifier through the digital link.

4.4. DIVERTOR AMPLIFIERS MANAGER GAM

The DAMGAM is a module created in order to let the VS system act on the divertor coils, which are normally controlled by the Shape Controller [7]. In particular the DAMGAM made possible the application of voltage kicks to the divertor coils.

A block diagram of the DAMGAM is shown in Figure 6, where P is a 4-by-4 invertible matrix which defines a linear transformation that maps the four divertor voltage requests received from the Shape Controller into a custom *P-space*. In this space a gain and a saturation can be applied to each signal, and the transformed signals pass also through a kick controller which works in almost the same way as the VAMGAM module described in Section IV-C. Eventually the signals are transformed back in voltage requests to the divertor amplifiers.

Thanks to its highly configurable structure the DAMGAM can be effectively used to explore all the possible interactions and advantages of using also the divertors for the task of the vertical stabilization. An example of use of the DAM is shown in Figure 7. In this experiment a full voltage kick on the divertor coils $D1$, $D2$ and $D4$ was applied by mean of a P matrix with the first row defined as $[1 \quad 1 \quad 0 \quad -1]$, corresponding to the linear combination of the divertor voltages that affects the plasma vertical movement. The other rows of P correspond to the linear combinations that less affect the vertical movement and have been computed through a singular value decomposition (SVD) decomposition. Figure 7(a) and 7(c) show the voltage input and output of the DAMGAM respectively, during the application of a 3 kV periodic kick (see Figure 7(b)).

5. USER INTERFACE

This section introduces the VS system user interfaces. Two main graphical interfaces are available, namely the *Level 1 Interface* (L1-Interface) and the *Web Interface*. The former allows the user to setup all the VS system parameters before the experiment, while the latter permits to monitor the state of the system during the experiment.

5.1. LEVEL 1 INTERFACE

The structure of the L1-Interface is made of several graphical layers each one corresponding to a different level of abstraction. Such layers are organized in two main levels:

- *Real-time executor level*, which allows the user to load the configuration of MARTe. In particular, the user can specify the GAMs to be executed together with their parameters, specifying them by means of text files. It is important to note that this level is common to all MARTe-based applications.
- *Application level*, which is customized for the VS system. This level is designed so as to allow the user to set each single parameter of the controller before the experiment.

The *Real-time executor level* is made of three different graphical pages:

- the *MARTe Layer* page, which is used to setup the interfaces between MARTe and the other external systems;

- The *MARTe Thread Layer* page, where the user can load all the GAMs that make up the real-time system to be deployed. In particular for the VS system all the GAMs listed in Section III are loaded from this page, together with their configuration files.
- The *Patch* page, which is used when a change of the default system parameters (as they are specified in the configuration files) is necessary.

In general, the *Application level* depends on the particular system developed with the MARTe framework. The *Application level* deployed for the VS system is made of two graphical pages:

- The *General* page, which is used to set the parameters of the controller. In particular this page is organized in four subsections, each one corresponding to one of the following module: SPGAM, Observer GAM, Controller GAM and VAMGAM.
- The *Scheduler* page. This page is dedicated to the Scheduler GAM and allows the user plan the experiment by setting the VS behavior in each of the 25 available time windows, so as to For example, in each time window the user can choose the estimation of the plasma vertical velocity to be controlled together with the desired control algorithm (see Section III). This page permits also to set the desired VAMGAM and DAMGAM behaviors, and in particular to switch on the kicks perfomed by these two modules (see Section IV-C and IV-D).

5.2. WEB INTERFACE

The *Web Interface* is based on the MARTe framework and it is automatically generated by the real-time application, i.e. by the VS system. This graphical interface allows the user to *navigate* into the GAMs structure so as to check the value of the parameters loaded in the VS system.

Note that, while the L1-interface is a JET specific SunOS based application, the user can access to the Web Interface by using any system equipped with an internet browser.

6. EXPERIMENTAL RESULTS AND OFFLINE SIMULATIONS

The commissioning of the new VS system has been done during C26 experimental campaign at JET. During the commissioning period the old control algorithm has been implemented on the new system, and ran in parallel with the old one.

The first tests consisted in comparing the voltages requested made by the new VS controller with the ones provided by the old system, so as to check the accuracy of the new references, and to verify 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 close 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.

Finally, the first plasmas were controlled using the new VS system in different operational scenarios. No software failures were ever observed during the execution of an experiment.

After this phase the new system has been employed during C27 JET experimental campaign for the commissioning of the ERFA amplifier and also to choose the optimum number turns for the coils in the RFA circuit. For this purpose, the performance of the vertical controller has been assessed not only with plasmas of varying growth rate but also with different plasma-wall clearance, q-profiles [26, Ch. 11], etc. This has been achieved by means of the analysis of the response to controlled perturbations (vertical and divertor kicks) in as wide a range of configurations as possible.

The kicks frequency has been assessed by means of closed-loop simulations in order to limit the excursion of the amplifier current and to avoid disruptions. In particular, the software architecture depicted in Section III has been exploited activating the three GAMs depicted in yellow in Figure 3.

In order to choose the correct duration of kicks so as to avoid disruptions, these durations have been assessed by means of closed-loop simulations. In particular, the software architecture depicted in Section III has beenexploited activating the three GAMs depicted in yellow in Figure 3.

As an example, in Figure 8 a comparison between experiments and simulation is shown for the case of 12kV positive kicks.

CONCLUSIONS

An overview of the new Vertical Stabilization system developed at the JET tokamak has been given in this paper. Thanks to the flexibility of its software architecture, the new system permits to customize the controller behaviour so as to meet the requirements during each phase of the experiment. Exploitation of such an architecture will guarantee the achievement of the desired performance during the advanced scenarios with highly elongated and unstable plasmas.

ACKNOWLEDGEMENTS

This work, supported by the European Communities under the contract of Association between EURATOM and CCFE, ENEA/CREATE, and IST, 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 would like to thank Dr. Peter Lomas, Dr. Fernanda Rimini for her useful comments, the members of CREATE Prof. Raffaele Albanese, Prof. Giuseppe Ambrosino, Prof. Marco Ariola, Prof. Alfredo Pironti, and the leader of the *JET PCU Enhancement* Dr. Flavio Crisanti.

FOOTNOTES

[1] If $R_{\max}$; $R_{\min}$; $Z_{\max}$; and $Z_{\min}$ are, respectively, the maximum and minimum values of the R and Z coordinates along the plasma boundary (see Figure 1), then the elongation is defined as

$$\kappa = \frac{Z_{max} - Z_{min}}{R_{max} - R_{min}}$$

It follows that for plasmas with circular cross section it is $\kappa = 1$, while for elongated plasmas it is $\kappa > 1$.

[2] ELMs are observed as a regular, but not absolutely periodic, loss of edge plasma confinement.

[3] ERFA is the name of the new voltage amplifier for the RFA circuit [5].

[4] We refer to *general purpose controller* as a controller which is robust enough to satisfactory work under any envisaged operational scenario, without pushing the performance to the best. In particular a general purpose controller should at least do not disrupt the plasma for almost all the possible operational scenarios.

[5] Advanced Telecommunication Computing Architecture.

[6] It should be noted that ten different estimations have been supposed sufficient for the JET application. However this number can be easily configured.

REFERENCES

[1]. J. Wesson, *Tokamaks*. Oxford University Press, 2004.

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

[3]. A. Beghi and A. Cenedese, “Advances in Real-Time Plasma Boundary Reconstruction,” *IEEE Control Sys. Mag.*, vol.**25**, no.5, pp. 44–64, Oct. 2005.

[4]. J.Y. Favez, “Enhancing the control of tokamaks via a continuous nonlinear control law,” Ph.D. dissertation, CRPP-EPFL, 2004.

[5]. F. Sartori *et al.*, “The JET PCU project: an international plasma control porject,” *Fus. Eng. Design*, vol.**83**, no.2–3, pp.202–206, Apr. 2008.

[6]. J. Wesson, *The science of JET*. Abingdon, Oxon: JET Joint Undertaking, 2000.

[7]. F. Sartori, G. De Tommasi, and F. Piccolo, “The Joint European Torus,” *IEEE Control Sys. Mag.*, vol.**26**, no.2, pp.64–78, Apr. 2006.

[8]. M. Bécoulet *et al.*, “Edge localized mode physics and operational aspects in tokamaks,” *Plasma Phys. Control. Fus.*, vol.**45**, no.12A, pp. A93–A113, Dec. 2003.

[9]. G. Ambrosino and R. Albanese, “A survey on modeling and control of current, position and shape of axisymmetric plasmas,” *IEEE Control Sys. Mag.*, vol.**25**, no.5, pp. 76–91, Oct. 2005.

[10]. V. Toigo *et al.*, “Conceptual design of the enhanced radial field amplifier for plasma vertical stabilisation in JET,” *Fus. Eng. Design*, vol.**82**, no.5–14, pp. 1599–1606, Oct. 2007.

[11]. 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, pp.723–738, May 1998.

[12]. R. Albanese, G. Calabr‘o M. Mattei, and F. Villone, “Plasma response models for current, shape and position control at JET,” *Fus. Eng. Des.*, vol.**66–68**, pp. 715–718, 2003.

[13]. A. Neto *et al.*, “Linux real-time framework for fusion devices,” *Fus. Eng. Des.*, vol.**84**, no.7–11, pp. 1408–1411, Jun. 2009.

[14]., “MARTe: a Multi-platfrom Real-time Framework,” in *Proc.* 16th *IEEE NPSS Real Time Conference (NPSS-RT2009)*, Beijing, People’s Republic of China, May 2009.

[15]. M. Ariola and A. Pironti, “The design of the eXtreme Shape Controller for the JET tokamak,” *IEEE Control Sys. Mag.*, vol.**25**, no.5, pp. 65–75, Oct. 2005.

[16]. R. Albanese *et al*., “Design, implementation and test of the XSC extreme shape controller in JET,” *Fus. Eng. Design*, vol.**74**, no.1–4, pp. 627–632, Nov. 2005.

[17]. G. De Tommasi *et al*., “XSC Tools: a software suite for tokamak plasma shape control design and validation,” *IEEE Trans. Plasma Sci*., vol.**35**, no.3, pp. 709–723, Jun. 2007.

[18]. M. Ariola *et al*., “Integrated plasma shape and boundary flux control on JET tokamak,””*Fusion Sci. Technol*., vol.**53**, no.3, pp. 789–805, Apr. 2008.

[19]. M. Lennholm *et al*., “Plasma vertical stabilisation at JET using adaptive gain control,” in *Proc. 17*th *SOFE Conf*., vol.**1**, San Diego, CA, 1997, pp. 539–542.

[20]. A.J.N. Batista, J. Sousa, and C.A.F. Varandas,–“ATCA digital controller hardware for vertical stabilization of plasmas in tokamaks,” *Rev. Sci. Instruments*, vol.**77**, no.10, p. 10F527, Oct. 2006.

[21]. F. Sartori *et al*., “n=2 compensation and variable gains for the JET vertical stabilisation,” *Fus. Eng. Design*, vol.**66–68**, pp. 727–734, Sep. 2003.

[22]. D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications*. John Wiley and Sons, 1979.

[23]. G. Franklin, J.D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed. Prentice Hall, 2006.

[24]. V. Riccardo *et al*., “Analysis of JET halo currents,” *Plasma Phys. Control. Fus*., vol.**46**, no.6, pp. 925–934, June 2004.

[25]. E. de la Luna *et al*., “Magnetic ELM triggering using the vertical stabilization controller in JET,” in *Proc*. 36th *EPS Conf. on Plasma Phys*., Sofia, Bulgaria, June 2009.

[26]. J. Freidberg, *Plasma Physics and Fusion Energy*. Cambridge University Press, 2007.

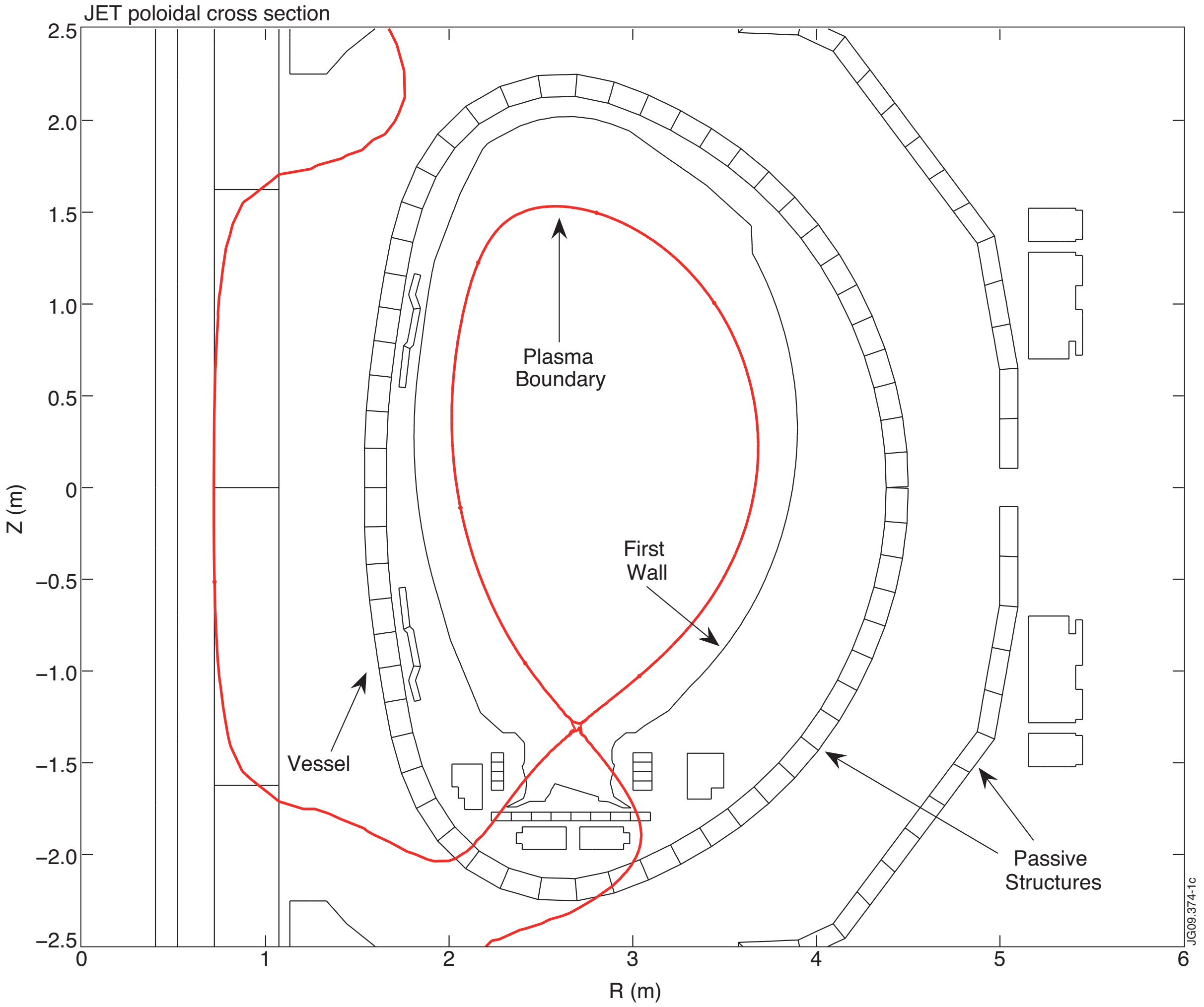

Figure 1: Poloidal cross-section of the JET tokamak. In particular the cross section of the JET Pulse No: 63005 at t = 31s is shown. This plasma has an elongation κ≅ = 1.7 and a growth rate greater than 1000.

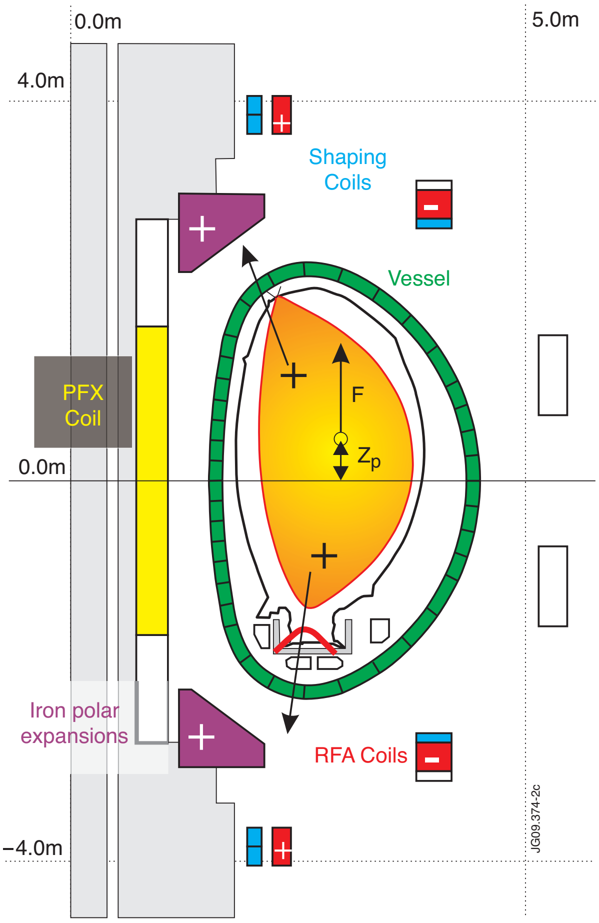

Figure 2: The mechanism of plasma elongation and resulting vertical instability. The iron polar expansions (shoes), colored in purple, are the main reason for the natural elongation of JET plasmas. Different plasma elongation can be obtained by adjusting the currents in the shaping circuit (blue) and PFX circuit (yellow). The vessel (green) provides the plasma passive stabilization, and the RFA circuit (red) is the actuator for active stabilization.

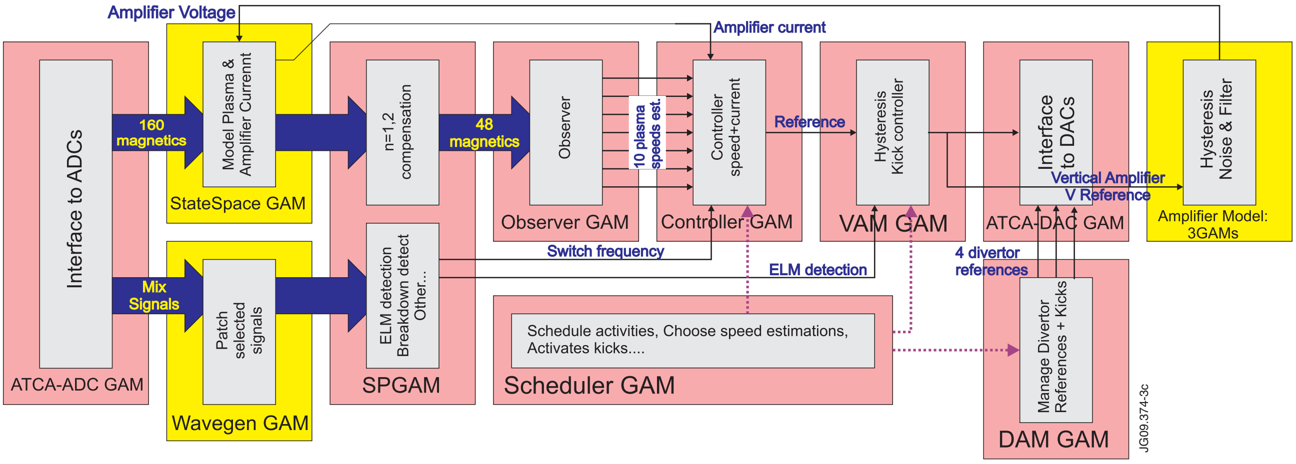

Figure 3: Block diagram of the overall JET Vertical Stabilization system software architecture.

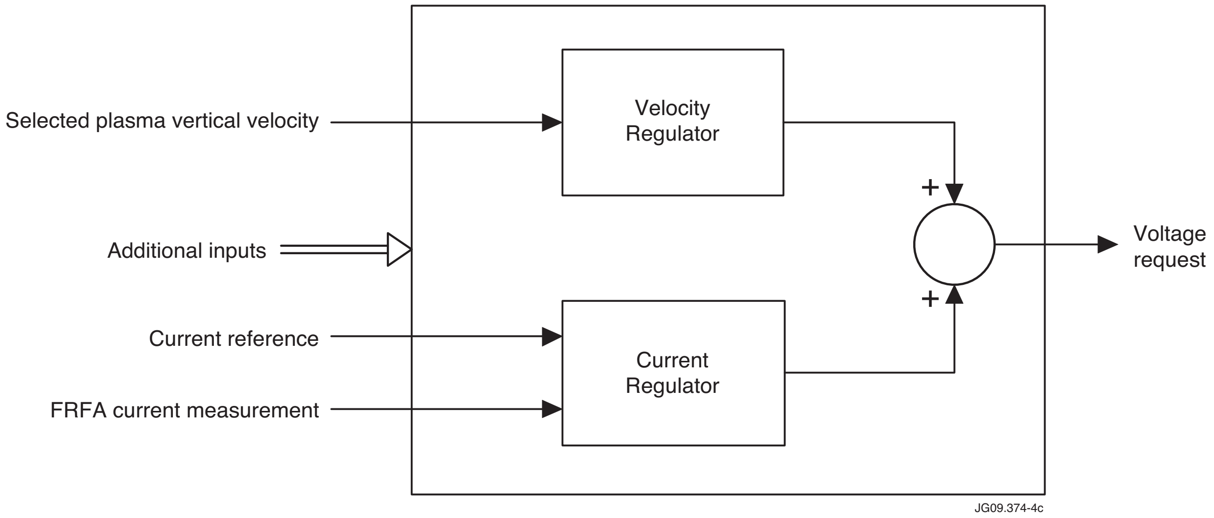

Figure 4: Basic structure of a control algorithm. Each control algorithm must provide a Velocity Regulator to achieve vertical position stabilization, and a Current Regolator to reduce the thermal losses in the actuator circuit.

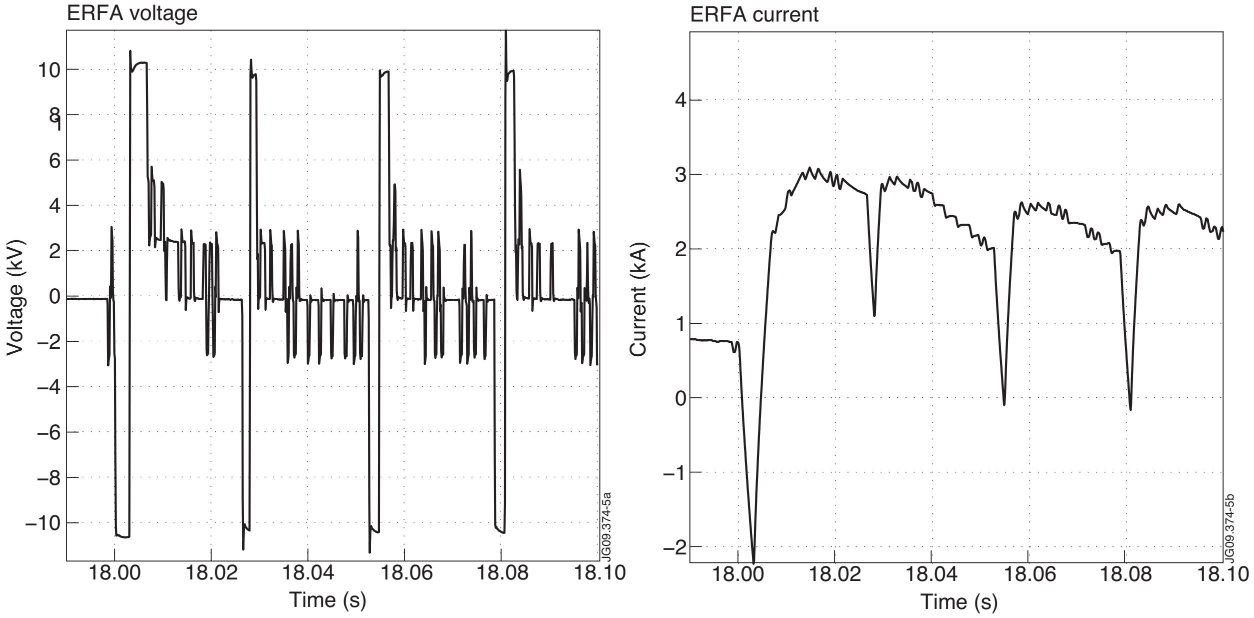

Figure 5: (left)Amplifier voltage. Note that after each negative kick there is a positive counter kick due to controller reaction. (right) 12kV negative kicks applied during Pulse No: 78951 starting from t = 18s.

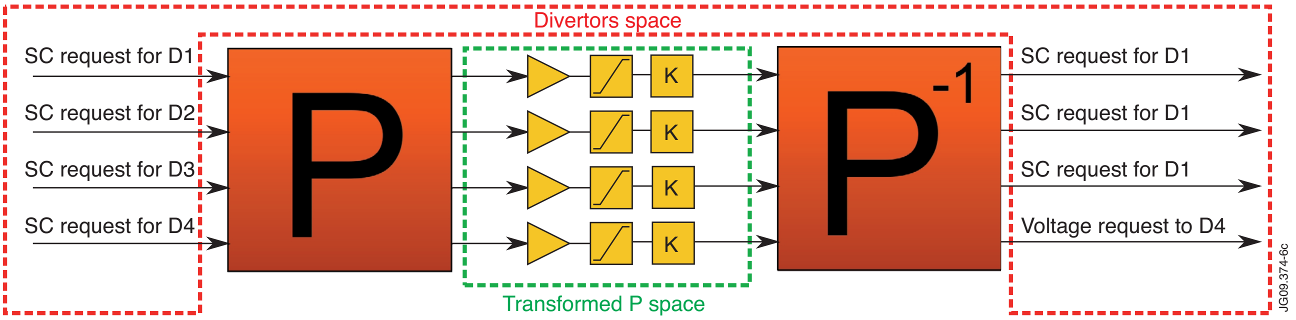

Figure 6: Block diagram of the DAMGAM module.

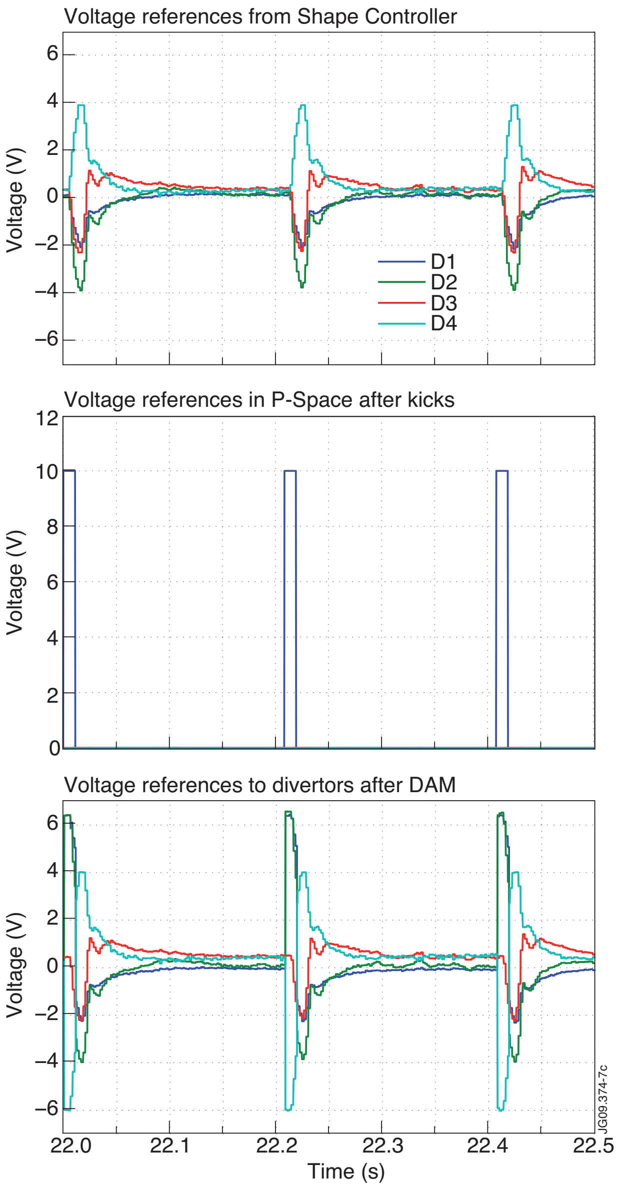

Figure 7: Example of periodic kicks in the DAM (Pulse No: 78528). Note that ADC and DAC voltages are shown. These voltages range between [−10V,+10V]. (a) Divertor voltages as request by the Shape Controller. (b) Full voltage kick in the P-space. (c) Actual voltage applied to the divertor coils. As required, D3 does not receive any kick.