Fusion Engineering and Design 89 (2014) 204–213

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

An overview of the software architecture of the plasma position, current and density realtime controller of the FTU

Luca Boncagni[a,*], Riccardo Vitelli[b], Daniele Carnevale[b], Cristian Galperti[c], Giovanni Artaserse[a], Daniele Pucci[d], FTU Team[1]

[a] EURATOM - ENEA Fusion Association, Frascati Research Centre, Division of Fusion Physics, Frascati, Rome, Italy
[b] Department of Computer Science, Systems and Production, University of Rome Tor Vergata, Rome, Italy
[c] EURATOM - ENEA - CNR Fusion Association, CNR-IFP via R. Cozzi 53, 20125 Milan, Italy
[d] Dipartimento Antonio Ruberti, Universit degli Studi di Roma La Sapienza, Rome, Italy

HIGHLIGHTS

- We implement the FTU PPCDC system using the MARTe Framework.
- We describe how it is logically divided and how it works.
- We show experimental examples to describe how it works.

ARTICLE INFO

Article history:
Received 18 May 2013
Received in revised form 11 February 2014
Accepted 20 February 2014
Available online 3 April 2014

Keywords:
MARTe
Real-time
Control systems
Nuclear fusion

ABSTRACT

Experimental fusion devices requires flexible control systems with a modern architecture, which allows the controller to be distributed and modular. The aforementioned requirements are all fulfilled by MARTe, a multi-platform framework for the development of low-latency hard-real-time control system already used with success in many European machine, it was decided to adopt it as the basis of the new FTU Plasma Position Current Density Control (PPCDC) system and the other coupled realtime systems. The main rationale to revamp the FTU control system was to use new technologies and to easily test different control solutions. MARTe has been proved effective from both the points of view, being platform independent, and having a modular architecture which completely separate the control algorithms from the rest of the infrastructure. We report on the new controller deployed at FTU. In particular, after a brief introduction on the machine, we illustrate the structure of the feedback system, together with a detailed analysis and appropriate experimental examples, of the various GAMs (modules) which make up the controller.

1. Introduction

FTU is a compact medium-sized high-magnetic-field and circular section tokamak machine equipped with two additional heating systems (Lower Hybrid, Electron Cyclotron Resonance Heating), two pellet injection systems (one shooting along the major radius and the other shooting along a vertical chord in the high field region), and a complete set of plasma diagnostics. Table 1 summarizes the main characteristics of the FTU machine.

The new PPCDC system described in [1] is now in charge of real-time control of the Plasma Position and Current and gas Density, while the toroidal field control is delegated to an independent system presented in [2].

The PPCDC system feedback sensors for the Position and Plasma Current control consists of 16 saddle loops, 16 poloidal field pickup coils that surround one sector of the vacuum vessel, one full-voltage loop all around the vacuum vessel, and its hardware integral and two Rogowski coils, one which measures the current in the toroidal magnet, and the other that measures the plasma current. The actuation on the plant is performed by four coils named F, T, V, H. Their limits and parameters are summarized in Table 2 and their positions respect the vacuum vessel are depicted in Fig. 1, where BP stands for pick up coil, while SA for saddle loops, and the arrows indicate the tangential orientation of the pick-up coil probes.

* Corresponding author. Tel.: +39 06 9400 5110.
E-mail addresses: luca.boncagni@enea.it (L. Boncagni), galperti@ifp.cnr.it (C. Galperti).
[1] See Appendix of Buratti P. et al 2012 Overview of FTU results 24th IEAEA Fusion Energy Conf. (San Diego, USA) OV/P-01.

http://dx.doi.org/10.1016/j.fusengdes.2014.02.039
0920-3796/

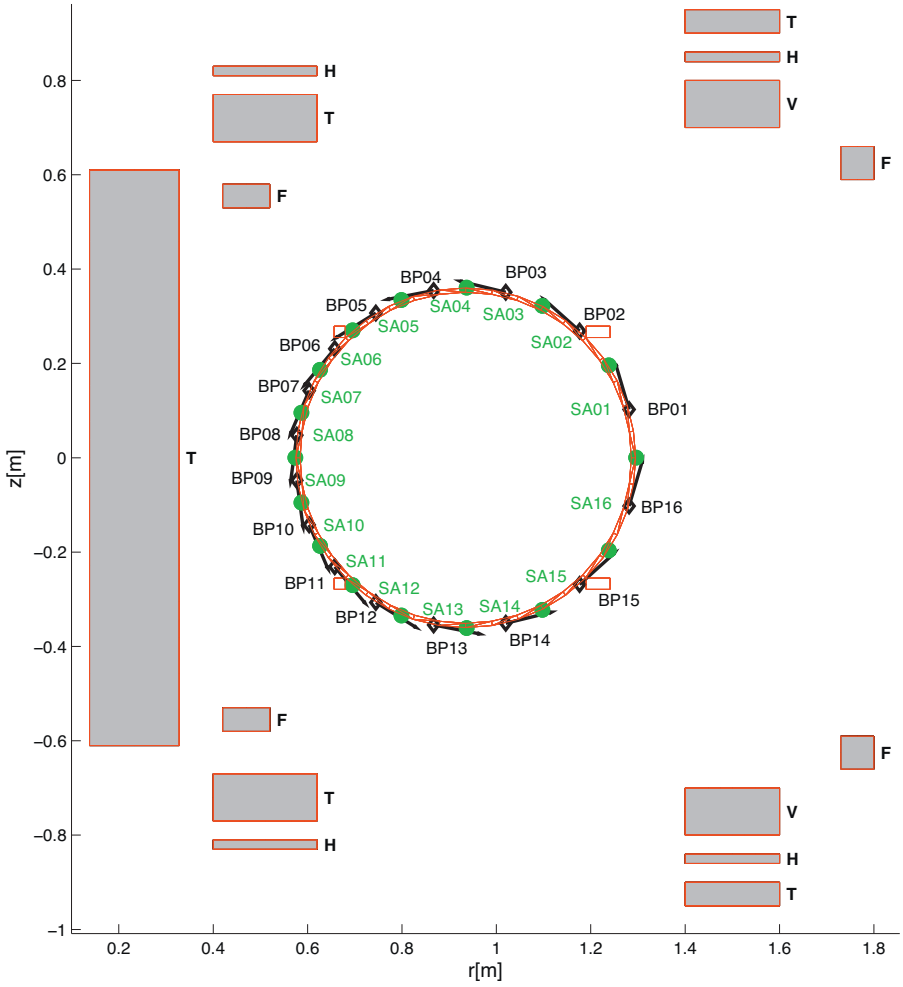

Fig. 1. FTU coils position as to the vacuum vessel.

Table 1
FTU characteristics.

Major radius	0.9365 m
Minor radius	0.35 m
Max plasma current	1.6 MA
Max toroidal field	8 T
Max elongation	1.09
Pulse repetition rate	1 pulse/20 min
Toroidal field energy	160 MJ
Poloidal field energy	200 MJ

Table 2
Coils names and limits.

Coil name	Current range (kA)	Max rate (kA/s)
F	±12.5	830
T	±25	63
V	±25	54
H	±1.2	32

Concerning gas regulation, whose principles are described in [3], the PPCDC system feedback measurements of chamber pressure and plasma line average density (at $R = 0.8965$ cm) are provided by two different sub-systems. During pre-fill, the measurement of the pressure is obtained by averaging the signals from two pressure transducers that are interfaced to the chamber in diametrically opposite position. During plasma discharge, the plasma line average density is provided by an ***interferometric measurement [4]. The control action is performed by up to four valves, whose calibration depends on specific experimental needs, and is usually agreed on before the beginning of each campaign.

The are many reason for a revamping of the feedback controller at FTU: the old controller ran on a single core Pentium architecture; the custom software that implement the PPCDC was realized many years ago, and ported during the years over several different operating systems without ever having been re-engineered on the top of a framework. In order to use new technologies (like shared memories and realtime network communication between controllers) and to efficiently test new control solutions (such as running realtime algorithms in parallel), a software and hardware upgrade was necessary. The mandatory requirement in the upgrade of the system was to keep backward compatibility with the most frequent experimental scenarios achieved at FTU during past years with the old feedback system. In this paper, after a brief introduction on the MARTe framework, we present a detailed analysis of the the structure of the PPCDC system, reporting as needed some experimental examples.

2. The MARTe Framework

MARTe (Multithreaded Application Real-Time executor) is a framework for real-time control systems development [5]. It is

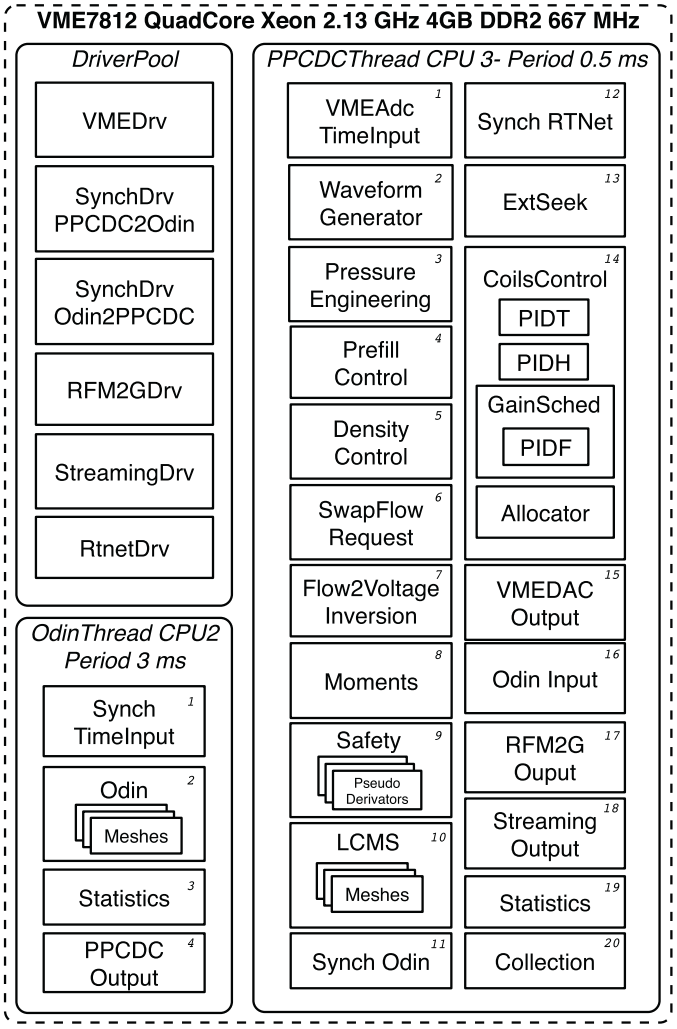

Fig. 2. The schema of the new FTU feedback control system in terms of GAMs.

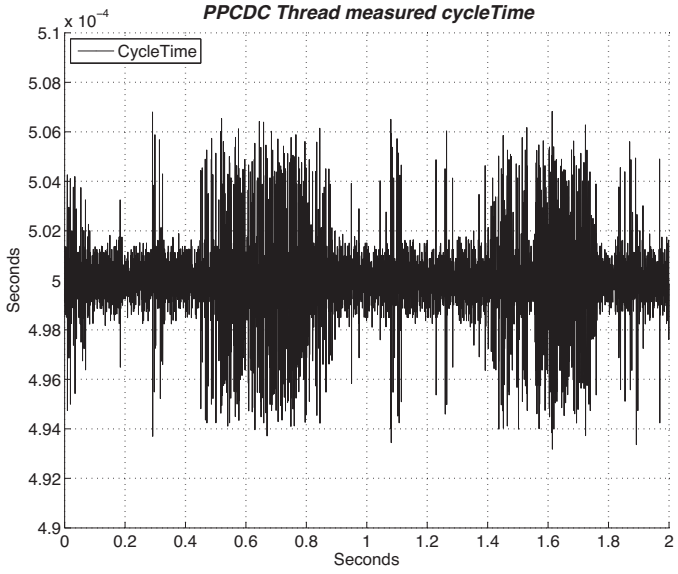

Fig. 3. PPCDC Thread cycletime.

implemented on the top of a platform-agnostic[2] C++ software library called BaseLib2, which hides to the programmer all the OS dependent details and provides tools and classes for the most common problems faced while developing control systems, such as error reporting and managing, data and configuration parameters access and matrix operations. BaseLib2 is heavily object-based and has been built from the ground up with flexibility in mind, making it possible to develop applications by juxtaposing objects and making them cooperate via message passing.

A MARTe application consists of several objects whose interconnections realize the architecture of a control system. The main algorithms of the control system are implemented by a series of small components called Generic Application Modules (GAMs), which exchange data through a Dynamic Data Buffer (DDB), and are otherwise completely independent. GAMs are sequentially ran by an instance of BaseLib2 RealTimeThread which is usually completely isolated on his own processor or core in order to obtain the best performance [6].

The communication with hardware is encapsulated into the GenericAcquisitionModule (GACQM) interface. Classes inheriting this interface are used by the RealTimeThreads for data input and output, and by the ExternalTimeTriggeringService service which triggers the new cycle. The IOGAMs (Input/Output GAMs) are a particular set of GAMs which call, in their Execute method, the GetData/WriteData methods of the GACQM to which they are directly linked.

The deployment of a MARTe system consists basically in writing the required high-level hardware drivers (i.e., the GACQMs), the control algorithm itself (i.e., the GAMs), and a configuration file (i.e., a text file with a proprietary syntax and semantic), which tells MARTe what GAMs will be used, and how they will be interconnected via the DDB. More than one RealTimeThread can be managed by a single instance of MARTe, each one running its own set of GAMs on a different processor, possibly synchronized by a pair of SynchronizationGAMs.

3. The PPCDC system

Fig. 2 shows the schema of the new FTU feedback controller in terms of GAMs (contained in the RealTimeThread boxes) and GACQMs (contained in the DriverPool box). With respect to the structure illustrated in [1], the ECRH wave injection control subsystem has been modified and moved to a set of industrial controllers [7]; the RTNet link has been used to get inputs for ExtSeek algorithm [8] from the LH Satellite System; and the RFM2GDrv implements the bidirectional communication with other real-time nodes; furthermore, a SafetyGAM has been added and the CoilControlGAM has been improved with new features which will be discussed in Section 3.5.

3.1. Plant synchronization and data acquisition

The first GAM that runs during a generic cycle of the PPCDC thread is the VMEADCTimeInputGAM and it uses the VMEDrv. This GACQM encapsulates the code to manage the I/O and synchronization operations of different VME devices[3] and can be configured to use one or more of them.

[2] Currently available for Windows, Linux, Linux/RTAI, Solaris, VxWorks, Mac OS X and Raspberry ARM1176.

[3] In particular it supports the Tundra chip universe II or tsi148 and different devices such as GE VME3123A ADC, CAEN V652b ADC, ACROMAG9210 DAC, Themis Timer,CAEN SCALER V830.

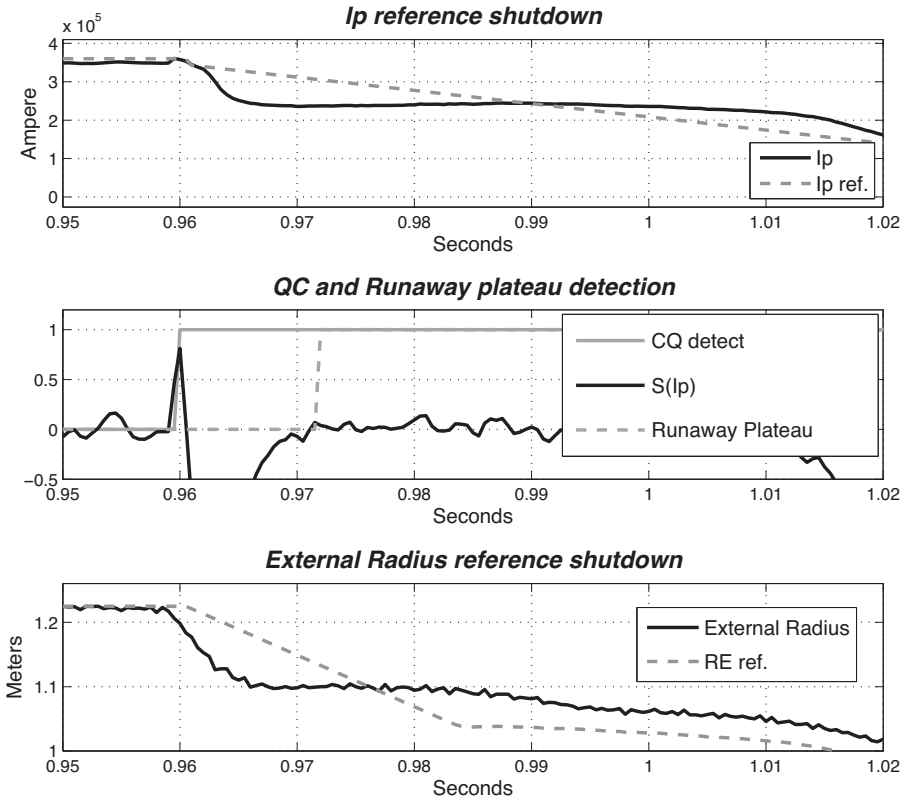

Fig. 4. CQ detection, runaway plateau detection and Ip and external radius references shutdown.

In FTU the timing and synchronization is performed by means of a timing network, a 1 MHz clock together with two gates,[4] which are distributed among the buildings through a fiber-optic loop and electro-optical/opto-electrical converters.

In the PPCDC Thread, the ExternalTimeTiggerInterface associated to the VMEADCTimeInputGAM is configured to poll the VMEDrv. The VMEDrv, in this case, waits until the DataReady bit of the first of four VME CAEN 652b ADCs (64 total available synchronous channels) signals the end of the acquisition. The four ADCs are triggered by a downsampled frequency (2 *kHz*) of the FTU master clock (1 MHz). When the data is ready and stored locally, the driver increments the usecTime state variable, which stores the time elapsed after the control action has been started (in microseconds), and subsequently provides the feedback signals to the DDB. The PPCDC thread uses other IOGAMs at different times during the realtime cycle, namely the SynchOutputGAM, RTNetInputGAM, the RFMGOutputGAM, the SynchInputGAM and the StreamingOutputGAM. These IOGAMs are linked to many GACQM types (the SynchDrv, the RTNetDrv,the RFM2GDrv and the StreamingDrv) which deal with, respectively, with the data synchronization with the Odin thread, with the LH Satellite Station, with the shared memory module for the ECRH system and with the non hard-realtime data streaming towards the control room GUI.

As concerning the realtime performance, on this hardware architecture by using Linux/RTAI, the measured cycle time (from the MARTe PerformanceMonitor and the StatistcGAM) reported in Fig. 3 presents a jitter less then ±1.2% of the period of the system.

3.2. Safety

The vast majority of the security checks on critical input signals (such as the runaway electrons detection monitoring the plasma current and the hard-X level and the reference shutdown policies) have been embedded in the SafetyGAM. Generally, a references shutdown policy patches some of the experiment's preprogrammed waves, for examples I_{Pr} plasma current and R_{Er} external radius references, in case of runaway detection or just I_{Pr} in case of disruption. In order to estimate the current quench (CQ) and RE (runaway electrons) beam plateau onset or plasma disruption, the SafetyGAM needs the time derivative of some signals such as I_P to check against an opportune threshold. The PseudoDerivator object implements a discrete time filter $S(\cdot)$ that approximates the time derivative of the signal $I(t_k)$ as

$$S(I(t_k)) = \frac{\sum_{i=0}^{c-1} I(t_{k-i}) - \sum_{i=1}^{c} I(t_{k-d+i})}{c(d-c)T_s}, \quad (1)$$

where $T_s = 0.5$ ms is the sampling time, $I(t_k)$ is the sample of the signal $I(t)$ at $t = t_k$ and $d \geq 2c > 0$ are tuning parameters. The proposed method allows to effectively reduce the high frequency noise which is usually amplified by standard (Euler) backward difference.

Fig. 4 shows the I_{Pr} and R_{Er} shutdown during an runaway induced disruption.[5] Specifically in the second plot the CQ flag rises when the $|S(I_p)| > 7.5e^6$, while the runaway plateau flag rises when |IptCQ/Iptk| < 0.85 for a configured number of cycles (10 cycles). Another example of security checks performed during a generic discharge is related to the plasma density and chamber pressure

[4] Each of them signals a precise time of the pulse sequence, respectively STARTRUN −20 s and FSC −10 s before plasma zero.

[5] Low plasma density and Neon injection, in this case from 0.89 and 0.95 s.

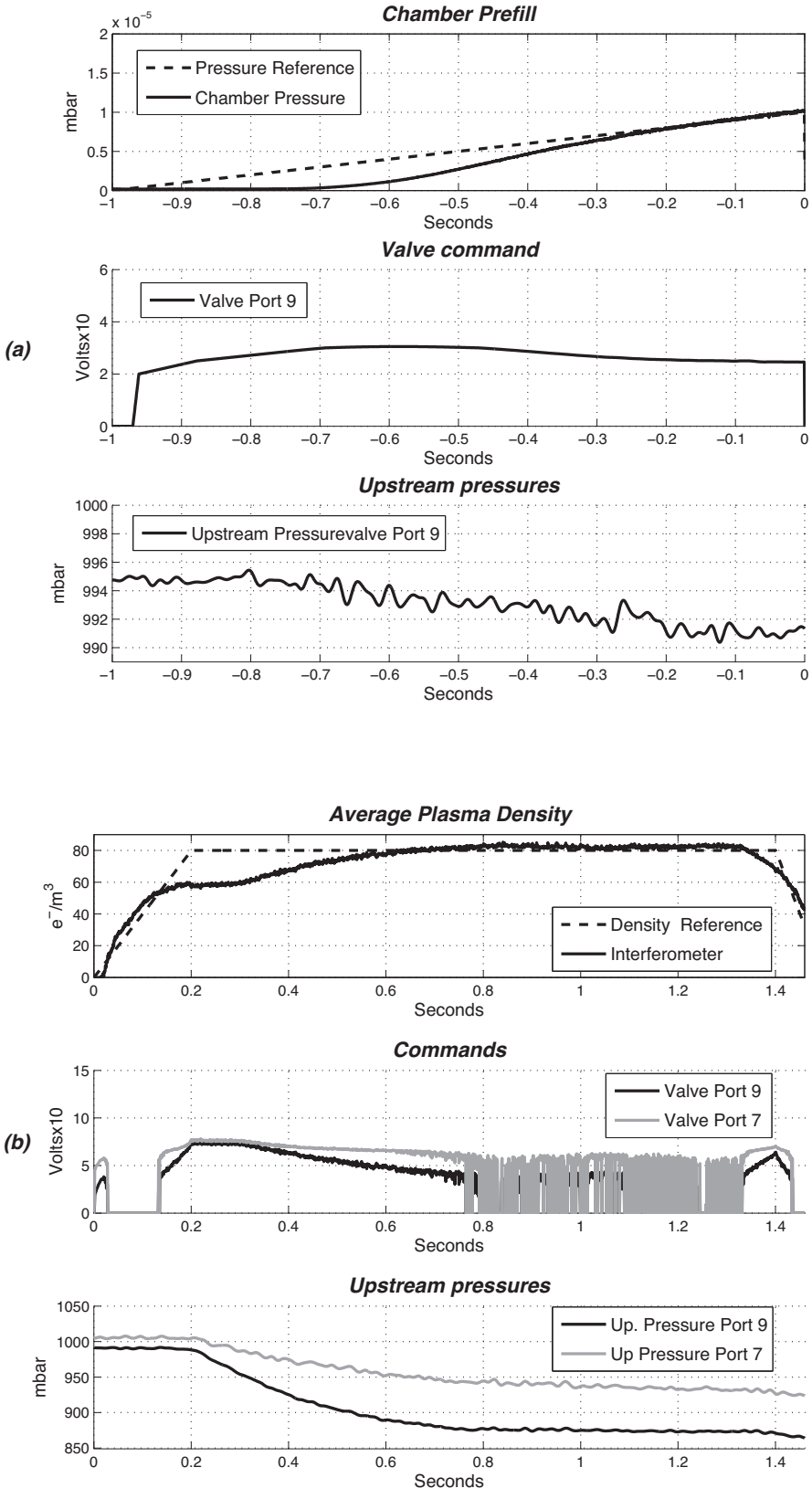

Fig. 5. (a) Prefill Control $-5 \geq t \leq 0$ and (b) plasma line average density control $0 > t \leq 2$.

measurements: if one of the above signals is currently used in feedback and it is outside its nominal ranges, within a configurable time interval, BadGas flag is switched on and the plasma current reference shutdown starts.

3.3. Gas injection

The gas regulation of a FTU discharge is programmed choosing both prefill reference in mbar (in the range $-5 \leq t \leq 0$ s) both

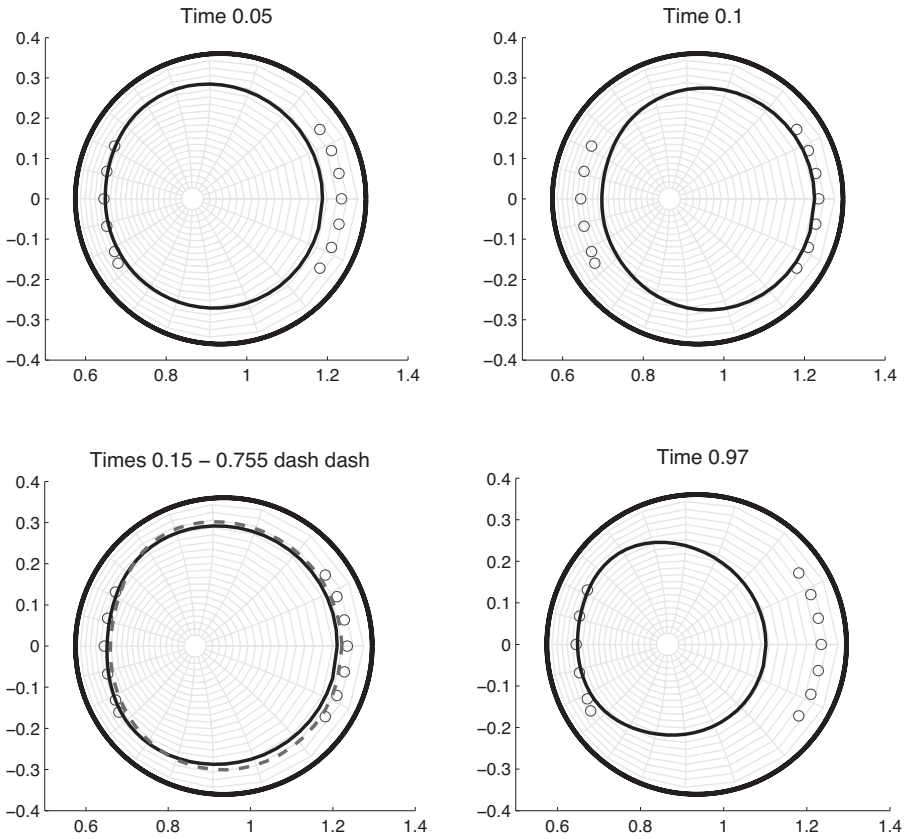

Fig. 6. Reconstructions at startup, at flat-top, elongating (allocator active mode 2 from 300 ms), during a runaway disruption.

the plasma line average density reference in e^-/m^3 (in the range $0 < t \leq 2$ s). The references are designed by mean of a tool and configured in the WaveformGenerator before the start of each experiment. The PKREngineeringGAM applies a gas dependent exponential formula to calculate the right value for the chamber pressure signals. The PrefillControl and PlasmaDensityControl GAMs produce the flux requests to be satisfied by the valve(s). The SwapReqGAM takes as inputs this two requests and generates a third signal that contains the first one when $t \leq 0$ and the second when $t > 0$. The new signal goes to FluxVoltageInversionGam that produce the valve (or valves) command(s). The curious reader can catch the basis of the new control strategy for gas injection at FTU from [9,10] and [3]. Actually, we extend such controller to more valves during the line average plasma density control. This has been done inside the FluxVoltageInversionGam by splitting the controller request.[6]

Moreover VMEDrv acquires the pressures of the volumes that are ahead of the valves (the volumes are known but their sizes differ in the range 36–48 cm^3). Each volume can be isolated from the main gas line by mean of a permeable membrane. The permeable membrane allows the gas to fill the volume with a slow time constant respect the injection one in the experiment. In this situation the valve flux is not only a function of the command, but it is also function of the upstream pressure. The advantage of this choice relies on the fact that we are able to measure the amount of gas particles injected by each valve in the chamber.[7] After a set of experiments at different upstream pressures for each valve, we was able to make a linear fit between the default valve characteristic Volt/Flow (estimated at 1300 mbar) and the others in the range 1300–500 mbar, In next experiments this linear expression will be added in the realtime inversion to chose the command for the required flow at the actual upstream pressure.

Fig. 5 shows the prefill and plasma density controls behavior. The prefill (Fig. 5a) is performed just by one valve and the upstream pressure decreases of about 10 mbar to perform the requested operation.[8] For the same shot, Fig. 5b shows the plasma line average density control using two valves, in this case the 70% of request is demanded to the first valve and the remaining to the other (because of their calibration maximum).

3.4. Last closed magnetic surface

$$\psi(\theta, \tilde{\omega}) = \frac{1}{\sqrt{ch(\theta) - \cos(\widetilde{\omega})}} \sum_{m=0}^{\infty} (M_m^i f_m(ch(\theta)) + M_m^e g_m(ch(\theta))) \cos(m\widetilde{\omega}) + C \qquad (2)$$

As concerning the equilibrium, the LCMSGAM and the OdinGAM must know the numbers and the coordinates (r, z) of the limiters, which can be moved or added (Lower Lithium Limiter) according to the request of the experimental programme. Moreover the desired position[9] is used by the LCMSGAM to calculate the flux errors along R and Z directions respect the reconstruction. As described

[6] The valves do not have the same start command and the same behavior so the flux/voltage functions are not the same for that a good compromise for the split-ratio should be chosen according to the valve calibration.

[7] This measures in conjunction to valve commands and chamber pressure can be used to make a better model of all the chain.

[8] The chamber volume around 3.77 m^3 and during the operation its temperature around 93 K. The port 9 volume is 48 cm^3 and the gas line temperature is around 293 K.

[9] The position is specified in terms of internal and external radii on the machine equatorial plan $z = 0$, and the upper and lower radii on the center vessel major radius $r = 0.9365$ m.

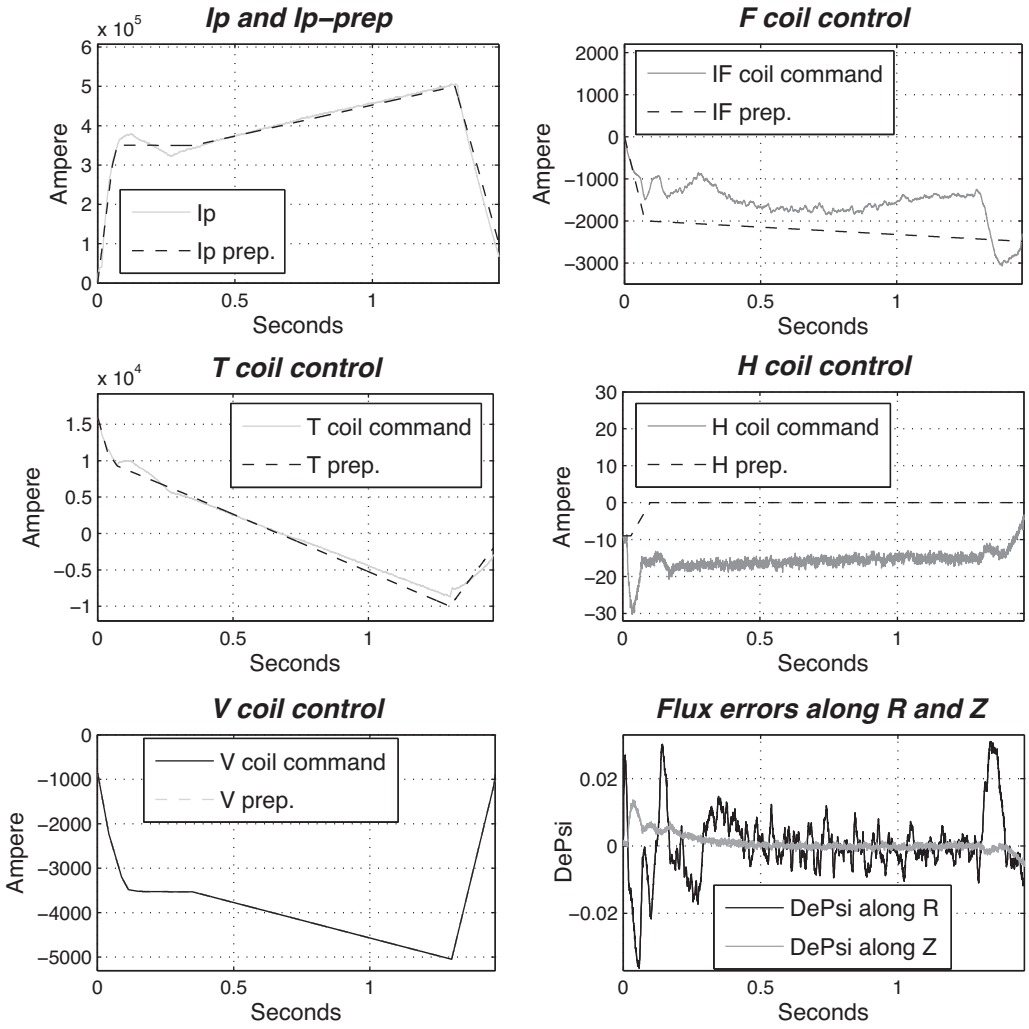

Fig. 7. PFC currents control using V only feed-forward.

in [11], the scalar flux function ψ that satisfies the Grad–Shafranov equation admits, under certain conditions, a decomposition, called multipolar expansion, reported in (2), where f_m and g_m are geometrical functions and M_m^i and M_m^e are the internal and external multipolar moments of order m. In FTU, whose aspect ratio is between 2 and 4, the external and internal multipolar moments of order 0, 1, 2, 3 can be computed (MomentsGAM in Fig. 2) as a linear combination of the components S_ψ and S_β arising from the Fourier analysis, around the poloidal angle $\widetilde{\omega}$, of the signals defined as:

$$S_\psi(\tilde{\omega}) = \sqrt{ch(\theta_{pr}) - \cos(\widetilde{\omega})}\psi_{pr}(\widetilde{\omega})$$

$$S_\beta(\tilde{\omega}) = \frac{sh(\theta_{pr})}{2\sqrt{ch(\theta_{pr}) - \cos(\widetilde{\omega})}}\psi_{pr}(\widetilde{\omega}) - \frac{2\pi R_0^2 sh(\theta_{pr})}{(ch(\theta_{pr}) - \cos(\widetilde{\omega}))^{\frac{3}{2}}}\beta_{pr}(\widetilde{\omega})$$

where $\psi_p r(\widetilde{\omega})$ and $\beta_{pr}(\widetilde{\omega})$ are the measures coming from saddle and bpol probes showed in the vacuum vessel in Fig. 1.

The plasma boundary can be computed if the approximation that the multipolar expansion is constant at all points of such a surface is made. The limiter-plasma contact point for which the flux value results to be maximum, determines the correspondent constant value Ψ_L. Searching this constant flux value in a discrete mesh, leads to the construction of the last closed magnetic surface. Fig. 6 shows a set of reconstructions at different times; they refer to the runaway disruption discharge already presented in Fig. 4 of Section 3.2. Respectively the plots show the plasma boundary at startup, at the flat-top start, during elongation induced by the F coil pre-charge (see Section 3.5.1) and finally during the external radius shutdown at current plateau.

3.5. Coils control

As concerning coils control, a FTU discharge is programmed by choosing: the preprogrammed references for the currents of each of the coils F, T, V, H and the preprogrammed reference for the plasma current. During the ramp-up, (from 0∼ 100 ms), the V and T coils are used to regulate the I_P current growth rate,[10] in this phase the T coil preprogrammed is corrected by the PIDT which is feed by the plasma current error while the V, F, H coils are driven only by the preprogrammed signal; In the flat-top phase ($t > 0.1\ s$) two PIDs correct the F and H coils preprogrammed references to minimize the reconstructed magnetic flux error coming from the LCMSGAM while, as default, V coil is driven by only the preprogrammed signal.

Fig. 7 shows the coils control of the discharge presented in Section 3.3. On the left side, the plasma current is driven by

[10] The mutual inductance on the plasma of this coils are respectively estimated to be 110 and 80 mH, while F has a mutual inductance respect the plasma current of the order of 20 μH.

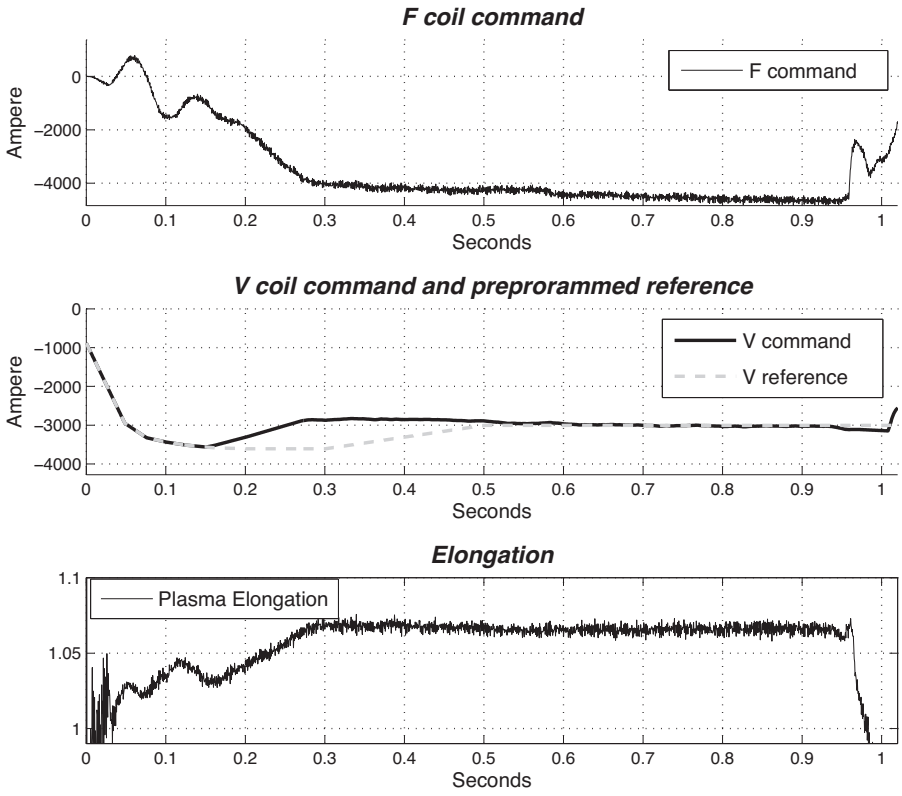

Fig. 8. PFC currents control using allocator.

the T (preprogrammed plus PIDT correction) and V (just preprogrammed) coils actions, while on the right side, the F and H coils actions minimize the $\delta\Psi$ errors along R and Z directions. The Responsible of the Operation (RDO) can chose if enable the the dynamic input allocation strategy for the flat-top phase. If it is enabled, the V coil current reference is adjusted[11] in conjunction with the F one to reach the configured equilibrium[12] without perturbing the plasma horizontal position. As shown in Fig. 2, the CoilsControlGAM encapsulates this logic by using different configurable objects. On the plant, the references calculated by the PPCDCThread are converted into analogue signals and sent to four thyristor converters, named ALF, ALT, ALV and ALH (VMEDacOutputGAM).

3.5.1. Allocator

Dynamic input allocation is a nonlinear control technique [12] that allows to reconfigure redundant actuators (e.g. the F and V coils at FTU) without perturbing the system output (i.e., the plasma horizontal position). This control scheme has been recently adopted at FTU [13] for elongation control, and at JET to achieve PF coils current saturation avoidance [14]. The Current Input Allocator (CIA) changes dynamically the PF coils current I_V and I_F to drive I_F to a desired value I_{Fr} without changing the total vertical field, i.e., $272/63\Delta I_V + \Delta I_F = 0$, and without any perturbation as to the horizontal position. The speed of the CIA to drive I_F to the reference value I_{Fr} can be adequately tuned up to the V coil rate saturation thresholds. The Allocator can work in three different configuration: I_F current control (mode 0), elongation control (mode 1), Runaway I_F control (mode 2) that permits to pre-charge the F coil thus ensuring the maximum excursion when a fast vertical field reduction is required (i.e., Runaway control [9]).

Fig. 8 shows the requests for the thyristors ALV and ALF of a shot in which the allocator is active. As the reader can notice, the V coil request is corrected respect the RDO's I_{Vr} reference. In this case the allocator is configured in mode 2 in order to maximize the I_F excursion to possibly mitigate the runaway induced disruption. The variation of the I_V current requested by the allocator starts at $t > \geq 150$ ms.

3.5.2. F Gain Scheduler

When I_{Fr} is within the region $|1.5|$ kA the ALF-PHSC current driver introduce a greater phase-lag (i.e., the approximate model that simulate its responses has a larger phase-lag[13]). This fact and the high value of the integral term of the PIDF lead oscillations. To avoid this, the Gain Scheduler block has been implemented. This hybrid controller uses a first order filter and Smith predictor based techniques, to perform a dynamic gains switch for the PIDF, thus increasing the derivative and the proportional gains and decreasing the integral one, as a configurable percentage of their original value. The proposed new gains avoid controller-induced oscillations, improving the closed loop performances. Fig. 9 compares the results of shots 36956 and 36953 with and without Gain Scheduler block. In this type of discharges, low toroidal field and low plasma current (Ip current 230 kA, toroidal field 2.5 Tesla, plasma line average density $50e^{18}$ e^-/m^3), the F coil current amplifier behaves nonlinearly resulting into a larger phase lag between the current demanded to the ALF amplifier I_{Fr} and the measured one I_{Fmeas}. As Fig. 10 shows in the first plot, there exits a considerable phase lag

[11] Currently the V coil current changes have to be significantly limited in rate to avoid undesired variations on I_P current due to the structural coupling between I_V and I_P, which is not yet compensated by PIDT.

[12] For example to obtain a desired elongation or to drive I_F current to a specific value.

[13] The estimation is obtained using an approximate second order model of the whole chain between I_{Fr} and I_{Fmeas}.

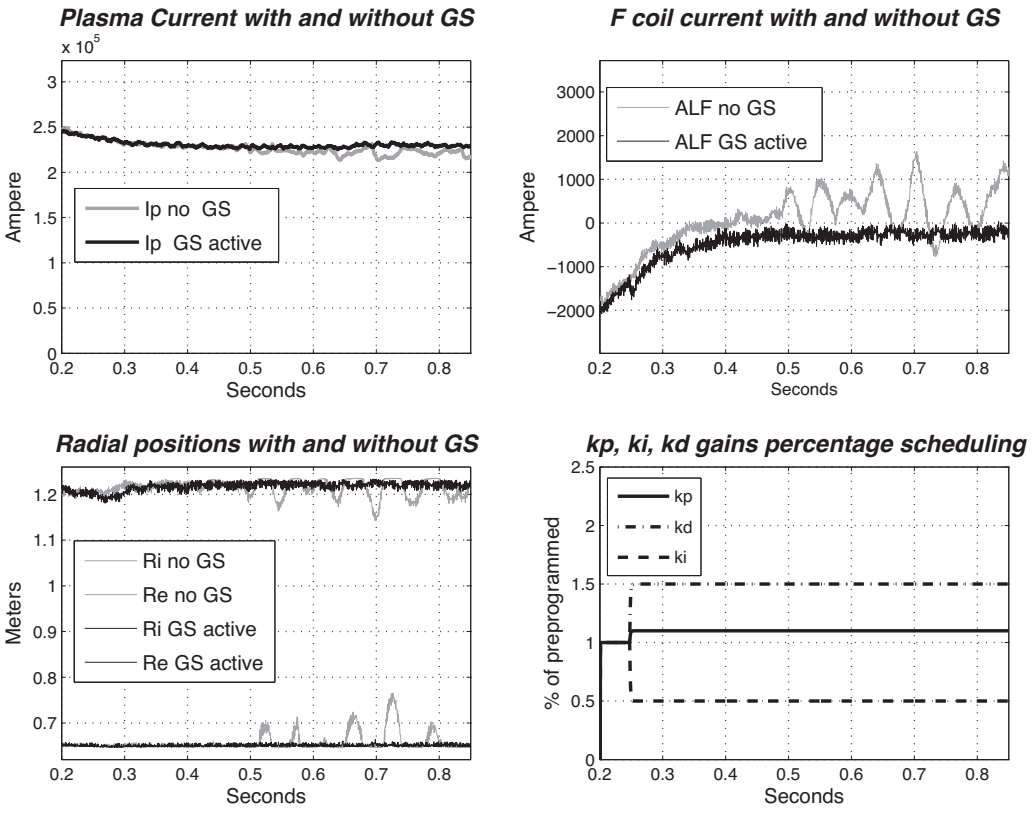

Fig. 9. F coils and Plasma Current Control with and without Gain Scheduler.

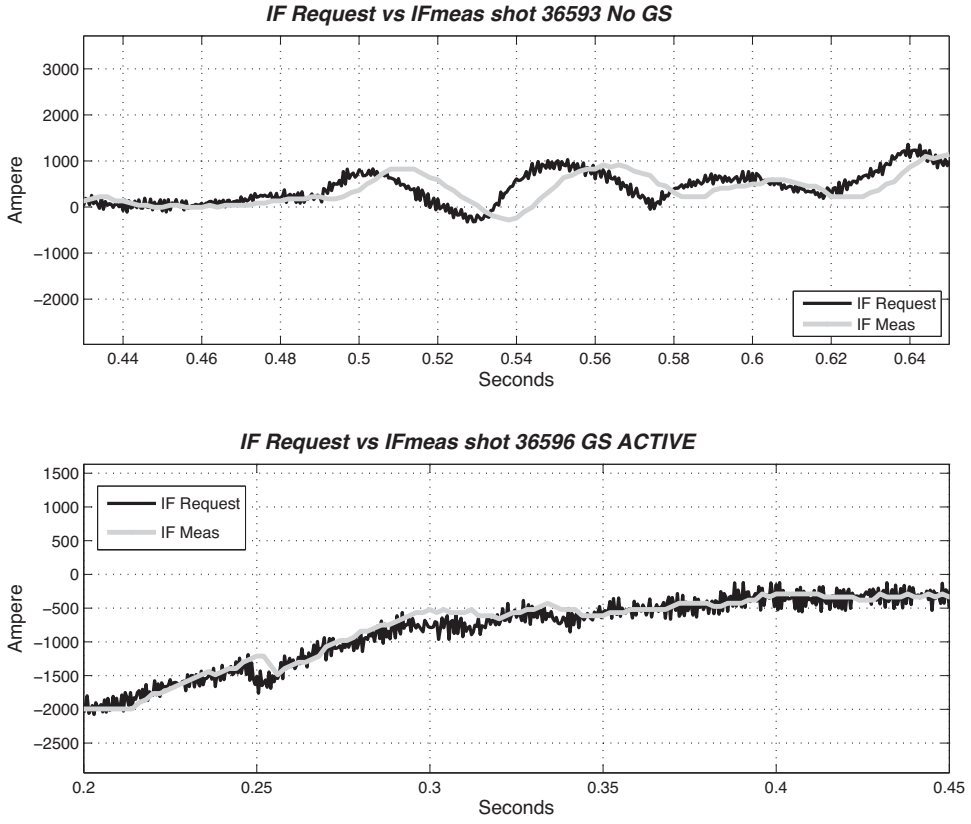

Fig. 10. Delay on $I_{Fmeas} - I_{Fr}$.

between I_{Fmeas} and I_{Fr} around 0.49 ms when $|I_{Fmeas}| = 1.5$ kA. This large phase lag induces closed loop oscillations. In the last plot of Fig. 9, the improvements yielded by the use of a new PIDF Gains Scheduler on the plasma horizontal stabilization are shown.

When the PIDF Gains Scheduler becomes activate, within a configurable time interval and when the current $|I_{Fmeas}| < 1.5$ kA, the integral gain k_i is smoothly reduced (in percentage) whereas the proportional and derivative ones are increased; The gains change has been selected exploiting experimental models (second order) of ALF.

3.6. The Odin Thread

The solution of Grad–Shafranov equation inside the plasma implies the knowledge plasma current density, that can be expressed as flux surface function:

$$J_\phi(\theta_0, \tilde{\omega}_0) = 2\pi R_0 \frac{dp(\psi)}{d\psi} + \frac{\mu_0}{4\pi R_0} \frac{dI^2(\psi)}{d\psi}$$

where $p(\psi)$ is the kinetic pressure and $I(\psi)$ the diamagnetic plasma current. From a mathematical point of view, the solution is not unique, but anyway from a physical point of view, all experimental equilibria so far analyzed in tokamaks experiments seems to be very well fitted (see Section 5 and Appendix E of [11]) by a flux expansion limited to:

$$p(\psi) = p_\alpha \psi^\alpha + p_\beta \psi^\beta$$

$$I(\psi) = I_\alpha \psi^\alpha + I_\beta \psi^\beta$$

such expression allows the description of a large set of current density profiles. So, given the values of internal multipolar moments of orders $m = 0, 1, 2, 3$ as measured at $\theta = \theta_{pr}$ and having set up the same kind of mesh of the previous sections, the four constants $p_\alpha, p_\beta, I_\alpha$ and I_β are computed together with the ψ distribution until they fit (least squares) the measured internal moments. The predictive reconstruction code is iterative and it stops when the flux difference between two successive solutions is less than a configurable error or if the number of iterations exceeds the given threshold. The equilibrium reconstruction is performed by an auxiliary Realtime Thread; because of time constraints, its period can be configured from 3 to 10 ms.[14] The Odin thread, as explained in the previous section, is synchronized by the PPCDC Thread and receive from it a copy of the already computed internal and external moments; it runs the OdinGAM and then returns its results to the PPCDC one, for collection and to be forwarded through the reflective memory. We should remark that, at the moment, any kind of synchronization is performed in this second path, the PPCDC Thread does not wait to get the Odin's results. The reconstructed plasma equilibrium (the poloidal magnetic flux map on the poloidal cross section) is passed to ECRH control subsystem which will use it to evaluate a number of quantities involved in ECRH injection position control, among them: the magnetic equiflux surfaces and the ECRH heating position on equiflux surfaces, whose safety factor value is rational.

4. Conclusions

In this work we have reported on the new implementation of the PPCDC system of FTU, using the MARTe framework. During last experimental campaign, we tested all basics features regarding the discharge configurations supported by the old system. We tested the *ZERO* shot (i.e., only toroidal field presence for the saddle and bpol offset - no plasma presence but just magnetic data acquisition and storage), then we executed the PLANT TEST discharge (just low current preprogrammed waves with no PIDs and no plasma). After that we started plasma discharges, we tested all the admissible combinations of poloidal coils current signs, we moved the poloidal limiter inner, we used the Lower Lithium Limiter and more over we successfully extended the experiment time up to 2.5 s, an operational configuration never tested with the old system As to the old implementation, we introduced few changes in the basic plasma control policy, but we enabled the new Runaway I_F policy in the Allocator, we added the complete LCMS reconstruction over 128 points and the ODIN reconstruction over a 40×20 points mesh. Many discharge information like plasma current, plasma line average density, Hard-X ray level, security flags and Odin flux map are now streamed to a visualization GUI program, shown in [1], located in the control room enabling live views of what's going on during experiments. Furthermore, the use of the MARTe framework enables all the technological improvements that derives from it.

Acknowledgements

This work was supported by the European Community within the framework of the ENEA/EURATOM Association on Fusion Research. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

The authors would like to thank the PPCC team for their generous help in carrying out this research.

References

[1] L. Boncagni, Y. Sadeghi, D. Carnevale, G. Varano, R. Vitelli, C. Galperti, MARTe at FTU: the new feedback control, Fusion Engineering and Design 87 (12) (2012) 1917–1920.
[2] V. Vitale, C. Centioli, F. Di Maio, M. Napolitano, M. Panella, M. Rojo, et al., FTU toroidal magnet power supply slow control using ITER CODAC core system, Fusion Engineering and Design (2012).
[3] L. Boncagni, D. Pucci, F. Piesco, E. Zarfati, G. Mazzitelli, S. Monaco, A control approach for plasma density in tokamak machines, Fusion Engineering and Design (2013).
[4] A. Canton, P. Innocente, O. Tudisco, Two-color medium-infrared scanning interferometer for the frascati tokamak upgrade fusion test device, Applied Optics 45 (36) (2006) 9105–9114.
[5] A.C. Neto, F. Sartori, F. Piccolo, R. Vitelli, G. De Tommasi, L. Zabeo, et al., Marte: A multiplatform real-time framework, IEEE Transactions on Nuclear Science 57 (2) (2010) 479–486.
[6] D. Alves, A. Neto, D. Valcárcel, R. Felton, J. López, A. Barbalace, et al., A new generation of real-time systems in the jet tokamak, in: Real Time Conference (RT), 2012 18th IEEE-NPSS, IEEE, 2012, pp. 1–9.
[7] C. Galperti, L. Boncagni, E. Alessi, C. Sozzi, S. Nowak, G. Granucci, et al., Testing and commissioning the multinode ECRH realtime control system on the FTU tokamak, Fusion Engineering and Design (2014) (this issue).
[8] D. Carnevale, A. Astolfi, C. Centioli, S. Podda, V. Vitale, L. Zaccarian, New extremum seeking technique and its application to maximize RF heating on FTU, Fusion Engineering and Design 84 (2–6) (2009) 554–558.
[9] L. Boncagni, D. Carnevale, C. Cianfarani, B. Esposito, G. Granucci, G. Maddaluno, et al., A first approach to runaway electron control in FTU, Fusion Engineering and Design (2013).
[10] L. Boncagni, Y. Sadeghi, R. Vitelli, C. Centioli, S. Sinibaldi, V. Vitale, et al., Progress in the migration towards the real-time framework MARTe at the FTU tokamak, Fusion Engineering and Design 86 (6–8) (2011) 1061–1066.
[11] F. Alladio, F. Crisanti, Analysis of MHD equilibria by toroidal multi-polar expansions, Nuclear Fusion 26 (1986) 1143.
[12] L. Zaccarian, Dynamic allocation for input-redundant control systems, Automatica 45 (2009) 1431–1438.
[13] L. Boncagni, S. Galeani, G. Granucci, G. Varano, V. Vitale, L. Zaccarian, Plasma position and elongation regulation at FTU using dynamic input allocation, IEEE Transactions on Control Systems Technology 20 (3) (2012) 641–651.
[14] G.D. Tommasi, S. Galeani, A. Pironti, G. Varano, L. Zaccarian, Nonlinear dynamic allocator for optimal input/output performance trade-off: application to the JET tokamak shape controller, Automatica 47 (5) (2011) 981–987.

[14] The allowable error should be increased or the maximum iteration threshold should be reduced if we want to go fast, vice versa if we want to slow down.