Fusion Engineering and Design 87 (2012) 1917–1920

Contents lists available at SciVerse ScienceDirect

Fusion Engineering and Design

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

MARTe at FTU: The new feedback control

Luca Boncagni[a,*], Yahya Sadeghi[c], Daniele Carnevale[c], Andrea Di Geronimo[c], Gianluca Varano[c], Riccardo Vitelli[c], Critsian Galperti[d], Emanuele Zarfati[b], Daniele Pucci[b]

[a] EURATOM – ENEA Fusion Association, Frascati Research Centre, Division of Fusion Physics, Rome, Frascati, Italy
[b] Department Antonio Ruberti, University of Rome La Sapienza, Rome, Italy
[c] Department of Computer Science, Systems and Production, University of Rome Tor Vergata, Rome, Italy
[d] Istituto di Fisica del Plasma, CNR, EURATOM-ENEA Association, Milan, Italy

HIGHLIGHTS

- ► We show that the MARTe is a candidate for ITER PSH.
- ► We replace the old real-time feedback software using the MARTe framework.
- ► We describe all the work done for the integration.

ARTICLE INFO

Article history:
Received 24 July 2011
Received in revised form 11 August 2012
Accepted 28 August 2012
Available online 9 October 2012

Keywords:
MARTe
Real-time
Control systems
Nuclear fusion

ABSTRACT

Keeping in mind the necessities of a modern control system for fusion devices, such as modularity and a distributed architecture, an upgrade of the present FTU feedback control system was planned, envisaging also a possible reutilization in the proposed FAST experiment [1]. For standardization and efficiency purposes we decided to adopt a pre-existent ITER-relevant framework called MARTe [2], already used with success in other European Tokamak devices [3]. Following the developments shown in [4], in this paper we report on the structure of the new feedback system, and how it was integrated in the current control structure and pulse programming interface, and in the other MARTe systems already in FTU: RT-ODIN [5] and the ECRH and LH [6] satellite stations. The new feedback system has been installed in the FTU backup station (known as “Feedback B”), which shares the input signals with the actual feedback system, in order to simplify the validation and debug of the new controller by testing it in parallel with the current one. Experimental results are then presented.

1. Introduction

A modern Tokamak control system is a complex system from an engineering point of view: it should be easily adaptable to the experimental environment, standardized and simply upgradeable (in order to help new members of the fusion community to give their contribution without wasting too much time in the implementation details), modular (in order to easily add or remove experimental components) and decentralized. With this in mind, and considering the necessity of acquiring know-how to develop and maintain the control system of the envisaged FAST experiment [1], it was decided to upgrade and revamp the feedback control system of the FTU Tokamak. The ideal candidate to become the development framework on which to base the new system was the MARTe framework [2], actively developed and used on several fusion devices such as JET, COMPASS, RFX and ISTTOK [3].

The first part of this paper describes the MARTe framework, then the structure of the new feedback controller, presented in [7] and [4], will be detailed. Finally we present some results obtained with the new control system during standard FTU operation.

2. The MARTe framework

MARTe is a framework for real-time control systems development [2]. It is implemented in C++and relies heavily on object orientation in order to solve the most common problems faced while developing control systems (such as error handling, system configuration, etc.) in a simple and easily understandable way. A MARTe application is made by a series of small components called GAMs (Generic Application Modules), which exchange data through a Dynamic Data Buffer (DDB), and are otherwise completely independent. The GAMs are sequentially run by a RealTimeThread which is usually completely isolated on his own

---

* Corresponding author. Tel.: +39 06 9400 5110; fax: +39 06 9400 5524.
E-mail addresses: luca.boncagni@enea.it, boncagni@frascati.enea.it (L. Boncagni).

0920-3796/$ – see front matter © 2012 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.fusengdes.2012.08.006

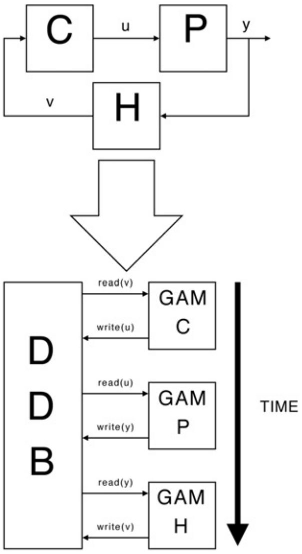

Fig. 1. Conceptual translation from a standard block diagram to a MARTe application.

processor (or core) in order to obtain the best performances with the minimum conceivable latency and jitter, for instance in the order of hundreds of nanoseconds in JET Vertical Stabilization system [8].

In Fig. 1 an example of how a feedback system in the standard block diagram form is translated in MARTe's GAMs/DDB architecture is reported.

The deployment of a MARTe system consists basically in writing the required high-level hardware drivers, the control algorithm itself (i.e. the GAMs), and a configuration file, a text file with a proprietary structure which tells MARTe what GAMs will be used, and how they will be interconnected via the DDB.

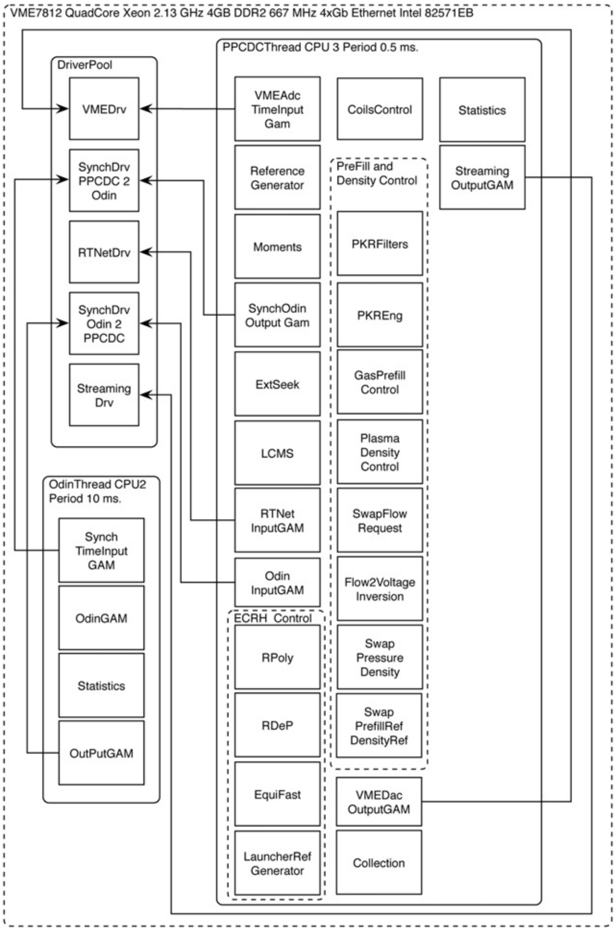

Fig. 2. Block diagram of the MARTe FTU feedback system.

3. MARTe at FTU

Considering the structure of the feedback system explained in [4], we tried to build a more organic structure for the system using various GAMs and limiting as much as possible the responsibilities of each one of them.

The main feedback control system structure is reported in Fig. 2, it is made by two real-time threads that run on a Quad Core VME Industrial PC hosted in a VME crate together with three ADC 12 channels 12 bits and two DAC 8 channels 12 bits. The Plasma Position Current Density Control thread (PPCDC) runs on the isolated CPU 3 with a period of 0.5 milliseconds. The Odin thread runs on the isolated CPU 2 together with the other MARTe's auxiliary threads (HttpServer threads, non rt Streaming thread) but it has higher priority and it has period of 10 milliseconds.

In the PPCDC thread, the ADCTimeInputGAM manage the communication with the sensors and actuators (ADC and DAC) using the VMEDrv, that also triggers the start of cycles. The ReferenceGenerator provides the preprogrammed references requested by the physicists.

The 32 magnetic measurements, together with the toroidal current and the V_{loop} integral, are passed to the MomentGAM, which reconstructs the pick-up toroidal field offset, or use the one stored during the Zero Shot[1], and removes it from the acquired data. After cleaning up the signals, it preforms the multi-polar expansion described in [9]. The 12 output signals (external and internal moments) are first forwarded to the Odin thread[2] using the SynchOdinOutputGAM and then are passed as inputs for the LCMSGAM together with the requested internal, external, upper and lower plasma radius. Starting from reflected power percentage measurement, if ExtSeekGAM is active, it evaluates a correction on the external preprogrammed radius in order to move the plasma and consequently maximize the coupling with the LH antennas [6]. The LCMSGAM evaluates the poloidal flux on the limiter contact points: the absolute maximum value for the flux among the contact points is the one of the last magnetic surface. The reconstruction of the LCMS is then carried out by iterating this process for all mesh points and stopping each time that the evaluated flux is greater or equal to the contact point one. Finally, the LCMSGAM calculates the plasma position error in terms of $\Delta\Psi$ using the preprogrammed radii (DEP and DEZ signals, for horizontal and vertical position error respectively). The plasma current together with the DEP and DEZ errors, are then sent to the CoilsControlGAM, which holds the controllers for the three poloidal field amplifiers, and the plasma current one. The CoilsControlGAM embeds four PID controller objects[3].

---

[1] The Zero Shot is a particular shot carried out before a set of experiments which share the same toroidal field. During this Zero Shot, the MARTe system gathers the signals from the magnetic probes in order to evaluate the coil pickup offsets. These offsets will then be used in the following experiments to compensate the magnetic signals in order to obtain a better reconstruction of the plasma shape and a faster convergence of the ODIN algorithm.

[2] The SynchDrv1 triggers the Odin thread that starts its calculation cycle.

[3] Note that it was necessary to merge the PIDs and the various nonlinear controllers in a single GAM, as their signals were heavily intertwined.