GA–A27940

THE ITER PLASMA CONTROL SYSTEM SIMULATION PLATFORM

By
M.L. WALKER, G. AMBROSINO, G. de TOMMASI, D.A. HUMPHREYS, M. MATTEI,
G. NEU, C. RAPSON, G. RAUPP, W. TREUTTERER, A.S. WELANDER,
and A. WINTER

SEPTEMBER 2014

GENERAL ATOMICS

GA–A27940

THE ITER PLASMA CONTROL SYSTEM SIMULATION PLATFORM

By

M.L. WALKER, G. AMBROSINO,* G. de TOMMASI,* D.A. HUMPHREYS, M. MATTEI,† G. NEU,‡ C. RAPSON,‡ G. RAUPP,‡ W. TREUTTERER,‡ A.S. WELANDER, and A. WINTER¶

This is a preprint of a paper to be presented at the Twenty-Eighth Symposium on Fusion Technology, September 29 through October 3, 2014 in San Sebastian, Spain and published in *Fusion Eng. Design*.

*CREATE/Università di Napoli Federico II, Napoli, Italy.
†Seconda Università di Napoli, Napoli, Italy.
‡Max-Planck-Institut fuer Plasmaphysik, Garching, Germany.
¶ITER Organization, Route de Vinon-sur-Verdon, St Paul-lez-Durance, France.

Work supported by
the ITER Organization under ITER/CTS/6000000037
and General Atomics IR&D funding

GENERAL ATOMICS PROJECT 49008
SEPTEMBER 2014

GENERAL ATOMICS

The ITER Plasma Control System Simulation Platform

M.L. Walker[a], G. Ambrosino[b], G. De Tommasi[b], D.A. Humphreys[a], M. Mattei[c], G. Neu[d], C.J. Rapson[d], G. Raupp[d], W. Treutterer[d], A.S. Welander[a], and A. Winter[e]

[a]*General Atomics, PO Box 85608, San Diego, California 92186-5608, USA*
[b] *CREATE/Università di Napoli Federico II, Napoli, Italy*
[c] *CREATE/Seconda Università di Napoli, Napoli, Italy*
[d] *Max Planck Institute for Plasma Physics, Boltzmannstr. 2, 85748 Garching, Germany*
[e]*ITER Organization, Route de Vinon-sur-Verdon, 13115, St. Paul-lez-Durance, France.*

The Plasma Control System Simulation Platform (PCSSP) is a highly flexible, modular, time-dependent simulation environment developed primarily to support development of the ITER Plasma Control System (PCS). It has been under development since 2011 and is scheduled for first release to users in the ITER Organization (IO) and at selected additional sites in 2015. Modules presently implemented in PCSSP enable exploration of axisymmetric evolution and control, basic kinetic control, and tearing mode suppression. A basic capability for generation of control-relevant events is included, enabling study of exception handling in the PCS, continuous controllers, and PCS architecture. While the control design focus of PCSSP applications tends to require only a moderate level of accuracy and complexity in modules, more complex codes can be embedded or connected to access higher accuracy if needed. This paper describes the background and motivation for PCSSP, provides an overview of the capabilities, architecture, and features of PCSSP, and discusses details of the PCSSP vision and its intended goals and application. Completed work, including architectural design, prototype implementation, reference documents, and IO demonstration of PCSSP, is summarized and example use of PCSSP is illustrated. Near-term high-level objectives are summarized and include preparation for release of an "alpha" version of PCSSP and preparation for the next development phase. High-level objectives for future work are also discussed.

Keywords: simulation, tokamak, plasma, plasma control

1. Introduction

The high cost and limited number of discharges planned for ITER, as well as constraints imposed by its nuclear mission, imply both minimal time for scenario and control tuning and a greater level of confidence needed in discharge performance prior to execution. The use of simulation for control development and verification has been well-established in research and commercial applications to support both of these requirements. Several operating tokamaks have made significant use of simulation tools in the development of control algorithms or key components of plasma control systems themselves. The broad success of this approach, both commercially and in the fusion community (references in [1]), led to an IO-funded task to develop such a simulation tool, known as the Plasma Control System Simulation Platform (PCSSP), to aid in development and testing of the ITER Plasma Control System. The current scope of the project is limited to deployment of an initial version of the platform together with support tools and selected modules. It is envisioned that this version may be extended in subsequent efforts to provide a fully capable control simulation tool to also support ITER machine/system design and configuration evolution and discharge scenario development, and to support plant troubleshooting during operations.

2. Overview of PCSSP

The vision for PCSSP and high level use cases and requirements were summarized and a preliminary architecture description was provided in [1]. Briefly, PCSSP is envisioned as the simulation platform that will support the widely-varying activities involved in developing the ITER PCS. These include initial exploratory simulations of proposed control algorithms through evaluation of candidate PCS architectural solutions, exception handling methods, policies, and architecture, methods for dealing with controller interactions, and candidate methods and architectures for control actuator management, to eventual automatic code generation for algorithm implementation in the operational PCS. To have value for control development, simulations must execute fast enough to support the commonly used develop/test iterative process. This implies an emphasis on plant system models that are simple but reasonably predictive, rather than the detailed models used for physics studies. PCSSP must also support validation of plasma and system models with data from existing tokamaks, to allow simulations used in PCS development to be usefully predictive.

There is an emphasis on rapid software prototyping during early development of new PCS algorithms, while later development requires formal change management. Both phases require flexibility and ease of use in constructing simulations from individual plant and PCS components. Variability in choices of components and

author's email: walker@fusion.gat.com

detailed actions depends on the stage of PCS development (Fig. 1). The PCS function may be emulated by a simulated PCS in PCSSP, actual PCS realtime (RT) code running on simulation (non-realtime) computers, or PCS code running on PCS realtime hardware. Algorithm development can use the simplified PCSSP Plant, while more detailed testing and eventual validation of ITER pulse schedules require more accuracy.

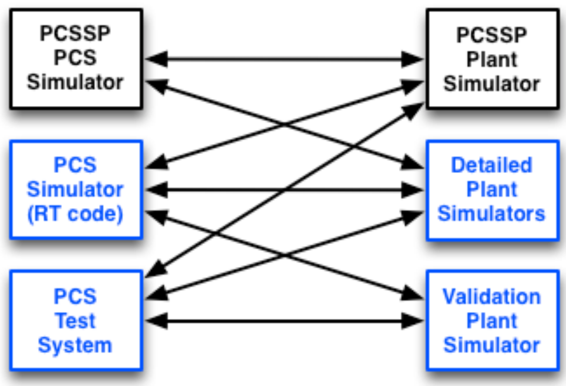

Fig. 1. Expected connections of PCS and plant simulators for closed-loop simulation. Blue blocks indicate objects external to PCSSP.

Matlab/Simulink was selected as the base platform because it already satisfies many of the system requirements [2], which all reflect the fundamental objective of minimizing time, cost, and effort to develop the PCS architecture, algorithms, and control policies.

The goals of PCSSP are fundamentally concerned with simulation of interaction between the ITER plant and PCS, so the PCSSP must include those two main functional blocks (Fig. 2), input processes to manage inputs to the simulation [Simulation Input Managers (SIM)], an output process to archive and display results [Simulation Results Manager (SRM)], as well as the interfaces between them. Pulse Schedule input is a distinct component within PCS SIM to enable replacement by the actual ITER pulse schedule when it becomes available. A functional block represented in Fig. 2 does not necessarily imply that all functions in that block are implemented as a single module.

In the plant, actuator modules simulate actuator responses to commands, diagnostic modules simulate processes involved in transforming physics quantities to real-time measurements, and the Tokamak+Plasma module simulates the combined plasma and device response to actuators. The SDN/CIN module simulates delays in moving measurement data from plant to PCS and commands from PCS to plant. SDN = Synchronous Databus Network (commands & diagnostic data) [5] and CIN = Central Interlock Network (machine protection control data). The Event Generator (EG) modules serve to trigger simulation of user-specified off-normal events.

The PCS simulator will contain multiple components, whose details will be defined during development of the PCS using PCSSP. However, it must include a set of control units, which receive input signals and produce output signals to perform specific functions such as feedback control, exception handling functions that detect and produce responses to any off-normal events that require triggering a change in control action [3], and a reference generator function to interpret the pulse schedule.

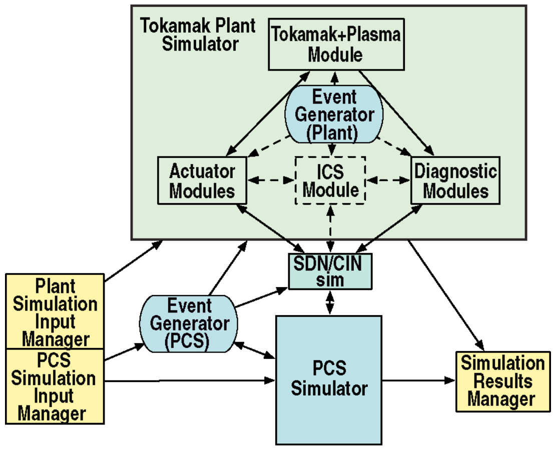

Fig. 2. PCSSP functional block diagram. Major components are plant simulator, PCS simulator, SIM, and SRM. PCS and plant simulators are in Simulink, but part or all of either block can be replaced by an external simulation or source of data to support use cases shown in Fig. 1. SIM and SRM execute in Matlab, outside of Simulink.

The Interlock Control System (ICS) module incorporates the Central Interlock System (CIS), an ITER overall protection circuit that responds to dangerous events [4], and Plant Interlock Systems (PIS), individual plant system (e.g. actuator/diagnostic) protection circuits.

3. Status of PCSSP Development

As of the end of 2013, a description of requirements and use cases [2], a final design and software architecture design [6,7], users guide [8], and a prototype implementation have been delivered. This prototype was demonstrated at IO in December of 2013. The delivered PCSSP design, described here, varies somewhat from the preliminary description provided in [1]. Figure 3 shows the Simulink model template provided with PCSSP, which users can modify to fit their application. The Plant represents models of the device and systems, the PCS represents the plasma control system for the device, and SDN_CIN uses ITER terminology (SDN, CIN) to represent the real-time networks but can be replaced by the communication infrastructure for any device, and EG represents the Event Generator. Signal lines represent classes of signals: "com" represents commands (to actuators), Diags represents diagnostic data, Plantdata can represent any data generated by modules within the Plant subsystem, CIS represents interlock signals generated by plant systems (including the Central Interlock System in ITER). Both the Plant and PCS accept Event Generator triggers and data (EGin) to allow emulation of off-normal events or faults anywhere in the closed loop.

We have chosen to minimize imposition of a priori data standards for the PCSSP architecture, since rigid standards can actually hinder many use cases. There are three classes of data associated with PCSSP simulations [7]: (1) setup data – the data needed to set up simulation to be able to run, including data for model configuration, simulation initialization, and any time-series input needed to drive the simulation, (2) signals data - the data transferred between simulink blocks at each step of simulation, and (3) results data - the data generated and stored by PCSSP during the simulation (includes selected signals data). A significant portion (perhaps all) of the setup data for the PCS is provided by the pulse schedule.

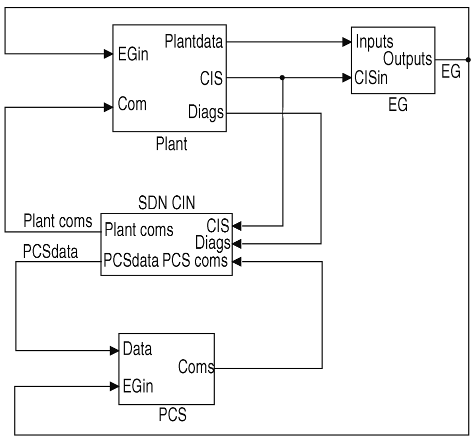

Fig. 3. Simulink template model.

No constraints are imposed on internal structure of individual modules but some standardization is imposed on the structure of data used to configure and initialize those modules. Although data content for a module is determined by the module provider, the data is required to be represented by a Matlab data structure with some required fields representing data for configuration, initialization, time series inputs, and module documentation. The documentation field itself has some prescribed content, such as descriptions of module input and output signals.

It is not actually required that users follow this standard to use PCSSP, but use of the standard provides several advantages. Use of a single data structure provides a method of grouping all data needed to use a module in simulation. This helps the user to identify the data needed to prepare the module for simulation and facilitates development and maintenance of SRM methods to archive and restore this data. It also prevents variable name conflicts in the Matlab workspace when individual modules are produced by independent modeling groups. It enables PCSSP to automatically archive together all input (e.g. setup) data and output (e.g. results) data for a simulation run. An auto-backup facility prevents accidental deletion of setup data while working. In addition, no functional limitations due to following this standard have been identified.

The wide variation of intended use cases makes standardizing module connections challenging and in some cases undesirable. Use of a predefined set of allowed signal names and content would make connections easier, but would require centralized definition and management and would severely constrain flexibility of the anticipated distributed ITER PCS development. The delivered prototype provides tools to support module signal standards when agreed upon, but also to support the unstructured signal use more common in exploratory development. To manage the expected large number of signals in larger simulations, a publish/subscribe mechanism has been developed. A connection map between modules is generated automatically from configuration data files, and can be inspected by users. Use of this mechanism saves effort compared to redrawing signal lines during iterative PCS development and can greatly reduce the visual clutter of multiple signal lines. One signal interface that will be standardized is the SDN/CIN, but this interface will be defined by the PCS design rather than by PCSSP.

A collection of sample modules were delivered in two Simulink libraries with the prototype of PCSSP and several were employed in the demonstration simulations. A generic Tokamak modeling library contains reusable Simulink blocks that can be used to simulate operation of any tokamak by configuring with input data. These blocks are grouped under the four categories Tokamak_Plasma_Modules (modules representing a generic Tokamak and Plasma), Utilities (to assist in constructing simulations), Tsdata_Generations (for use in creating time series input for a pulse schedule or the Event Generator), and Module_Utilities (to assist developers in creating new modules for PCSSP). A separate library contains ITER-specific modules grouped according to whether they are Plant modules (modeling a particular portion of the device and subsystems), PCS modules (representing a functional component of the ITER PCS), or Other (e.g., the real-time networks). Plant modules are further organized according to whether they represent Actuator, Tokamak+Plasma, or Diagnostic systems. There is also a module representing the ITER interlock systems. PCS modules are likewise separated according to whether they represent measurement processing, continuous control, command processing, exception handling, reference signal generation, or supervisory control. Maturity of the various modules varies widely, with the most mature being those that were used in the prototype demonstration. Similar device-specific libraries can be constructed for currently-operating devices for the purpose of validating plant modules and (if desired) control development.

Figure 4 illustrates one of the scenarios simulated using PCSSP as part of the demonstration in 2013. Early in time, a programmed change in plasma boundary is simulated. During this time, continuous controllers for gap and vertical control operate simultaneously with kinetic (electron density and internal energy) control and neoclassical tearing mode (NTM) control, which attempts to maintain electron cyclotron current drive (ECCD) mirrors to align with the 2/1 q-surface. This is followed by a requested beta rise. The requested value was not achieved because of limited available power. Later in the simulation, growing 2/1 island "events" are simulated ($w_{2/1}$=width of 2/1 island). NTM control responds by turning on gyrotrons (peaks in EC power) and aligning the EC mirrors to inject power at the 2/1 island surface (EC misalignment tends toward zero). When an island is not present the alignment can drift, but when the mode grows it provides additional information to the controller that allows better alignment. Finite mirror response time causes some delay in returning to good alignment, at which time the mode is suppressed ($w_{2/1}$ returns to a low value). The fourth time the island grows, the NTM control responds but a large portion of the EC power is not delivered due to a gyrotron failure, and the $w_{2/1}$ signal remains high. Exception handling logic in the controller detects this situation (NTM exception) and triggers the Disruption Mitigation System (DMS), which activates the massive gas injector (MGI), causing a sudden rise in density (n_e). The subsequent disruption is not simulated in detail, but n_e exhibits decay due to a finite confinement time, β_p collapses, and the plasma moves inboard as is typically seen during disruption.

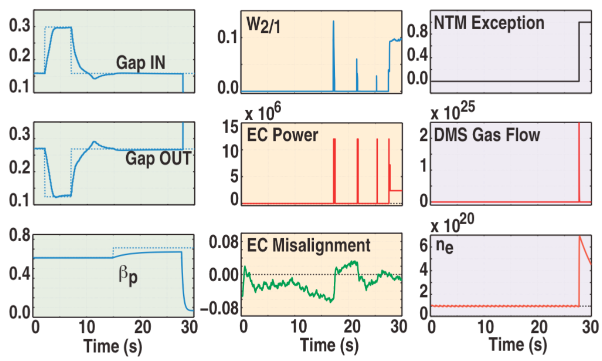

Fig. 4. Results of a control scenario simulated using the PCSSP environment. The selected signal plots shown were updated as the simulation executed.

4. Conclusions and Future Work

Development of a prototype of the PCSSP, previously described in [1], was completed in 2013 and demonstrated at ITER. A description of one of these demonstrations has been provided. The reference [1] described an initial design, significant portions of which were delivered in the prototype. In particular, the prototype system contains two Simulink libraries containing multiple Plant component simulation modules, both generic and ITER-specific, and PCS component simulation modules. A description of these libraries and modules has been provided.

Some changes were necessary to the initial design to better align with the anticipated high variability in use cases. These included a movement away from strict standards for overall user simulation model structure and signal exchange mechanisms. Instead support tools are provided to allow standardized methods when desired, while simultaneously supporting variability in intended use. For example, a generic template model is provided that automates adherence to the few standards imposed on Simulation models. A publish/subscribe signal data exchange tool is provided to facilitate definition and easy reconfiguration of signal paths.

The next development phase has as its primary objective preparing PCSSP for deployment for alpha testing by IO, the development group, and selected other ITER partners. It is also intended to enable simulation and exploration of evolution and control of plasma boundary, basic kinetics, and NTM, use of an event generator to create simulated events, and evaluation of basic continuous control, exception handling, and candidate PCS architectures.

Beginning during alpha testing and continuing throughout the PCSSP lifetime, additional modules for ITER and other tokamak plasma and device systems will be developed and incorporated into PCSSP. It is expected that following initial development with IO funding, PCSSP will transition to a mixed funded/open-source development model. In particular, modules needed specifically to simulate ITER PCS or plant systems will typically be developed under IO funding. PCS or plant modules for existing devices are expected to be developed as part of the support of those programs, but can help inform ITER module development. A rough development timeline and notional broad milestones are summarized in [9].

Acknowledgments

This work was supported in part by the ITER Organization under ITER/CTS/6000000037. The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

References

[1] M.L. Walker, *et al.*, Fusion Eng. Design **89**, 518 (2014)
[2] PCSSP Final Requirements Document v.1.2, ITER IDM ref. 4FK397 (2012).
[3] G. Raupp, *et al.*, Fusion Eng. Design **89**, 523 (2014)
[4] A. Vergara Fernández, *et al.*, Fusion Eng. Design **86**, 1137 (2011).
[5] H.G. Kim, E.J. Lee, SDN Software Architecture Design Description v.1.4, ITER private communication, B7AWBK (2013).
[6] Final Design Document for PCSSP, ITER private communication MUCS8C, v.1 (2014)
[7] Final Architecture Document for PCSSP, ITER private communication LAANVV, v.1 (2013)
[8] User's Guide for PCSSP, ITER private communication MUD4V8, v.1 (2014)
[9] A. Winter, Implementation Strategy for the ITER Plasma Control System, Proc. of 28th Symposium on Fusion Technology, San Sebastion, Spain, 2014, Paper P2.042.