GA–A26471

# ACCUMULATED EXPERIENCES FROM IMPLEMENTATIONS OF THE DIII-D PLASMA CONTROL SYSTEM WORLDWIDE

by

D.A. PIGLOWSKI, D.A. HUMPHREYS, M.L. WALKER, J.R. FERRON,
B.G. PENAFLOR, R.D. JOHNSON, B. SAMMULI, B. XIAO,
S.H. HAHN, D. MASTROVITO

JULY 2009

GENERAL ATOMICS

## DISCLAIMER

GA–A26471

# ACCUMULATED EXPERIENCES FROM IMPLEMENTATIONS OF THE DIII-D PLASMA CONTROL SYSTEM WORLDWIDE

by

D.A. PIGLOWSKI, D.A. HUMPHREYS, M.L. WALKER, J.R. FERRON, B.G. PENAFLOR, R.D. JOHNSON, B. SAMMULI, B. XIAO,* S.H. HAHN,† D. MASTROVITO‡

This is a preprint of an invited paper to be presented at the Seventh IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation in Fusion Research, June 15–19, 2009, Aix-en-Provence, France, and to be published in the *Proceedings.*

*Academica Sinica Institute of Plasma Physics, Hefei, China
†National Fusion Research Institute, Daejeon, Korea
‡Princeton Plasma Physics Laboratory, Princeton, New Jersey

Work supported in part by
the U.S. Department of Energy under
DE-FC02-04ER54698 and DE-AC02-09CH11466

GENERAL ATOMICS PROJECT 30200
JULY 2009

GENERAL ATOMICS

# ABSTRACT

The current DIII-D plasma control system (PCS) has evolved through several iterations into a robust platform that has been adopted at several fusion devices around the world. Each installation, as well as each new upgrade at DIII-D, has presented new challenges. Each of these challenges has provided an additional opportunity to expand our understanding of the requirements, alternative operational methods, and differing real-time implementations for tokamak plasma control.

This paper presents a brief historical overview of PCS hardware evolutions and describes some of the design, structure, and techniques that have allowed the PCS to be a productive component at many fusion facilities. It will also discuss some of the major differences between the individual PCS installations and bring to light some of the major challenges that were overcome during integration. The lessons learned from these experiences provide general solutions and can inform control system designs for other next-generation devices. We also describe some limitations of the PCS relative to identified present and future needs at DIII-D and other devices, and discuss planned upgrades to the PCS to address these needs.

# 1. INTRODUCTION

The DIII-D digital plasma control system (PCS) has been controlling plasmas in the DIII-D Tokamak for over 15 years. During this time, it has gone through numerous iterations of both hardware and software [1,2]. Its general benefits have been recognized by others in the fusion community and because of this it has been used as the basis for plasma control systems on several devices across the globe. These devices differ greatly in operational parameters; ranging from small experimental devices to large super-conducting machines. This paper discusses what has been learned during the development and use of the individual PCS implementations.

## 2. DIII-D PCS HARDWARE TECHNOLOGY

The original PCS hardware was SPARC and I860 based, entirely contained on a single VME bus [3]. Approximately eight years ago this was briefly altered to a SPARC and Alpha core hardware set interconnected with a high speed (2.0 Gigabit) dedicated network made by Myricom Inc., the Myrinet [2]. The current hardware set is entirely Intel based and can be remapped into numerous reconfigurations [1]. The DIII-D PCS routinely executes on up to 24 real-time processors or CPUs with feedback cycle times ranging between 11 microseconds to tens of milliseconds. Communication with "plant systems" is performed in many ways of input/output (I/O). In general, most input signals are given through analog-to-digital converters (ADCs). Output commands mostly are delivered to plant subsystems via digital-to-analog converters (DACs). Alternatively, digital I/O is used by the PCS for data and commands that have simple on/off states, such as fault signaling. ADCs, DACs and digital I/O are distributed across several nodes in the PCS. These nodes, located throughout the plant facility, allow the PCS to go to the data source, reversing the model of the first generation PCS where data was brought from various locations to the PCS. Figure 1 shows a general layout of PCS architecture.

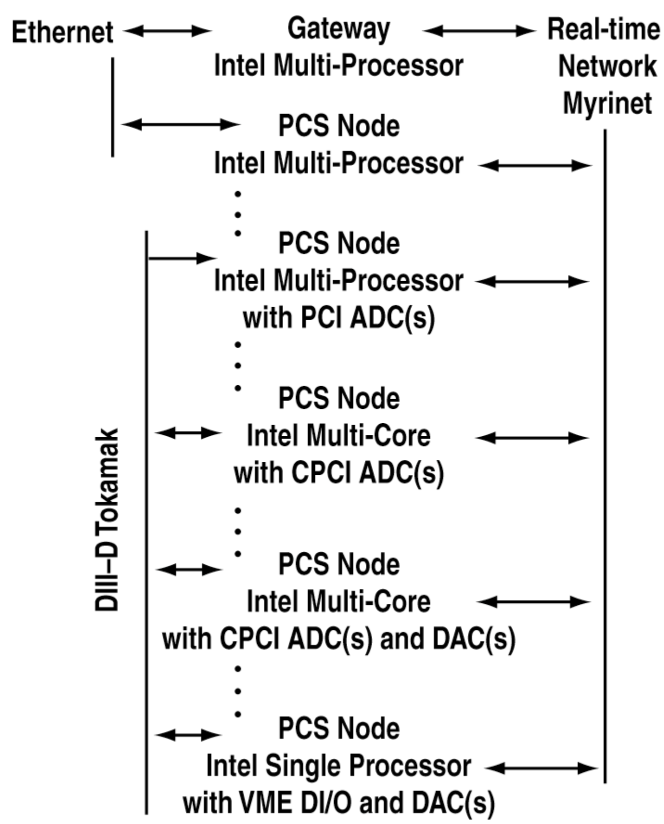


Fig. 1. PCS layout.

The layout of the PCS shows that data may be shared between real-time processes. This may include input data, output command data, and processed data (needed for shared computational purposes). Ideally inter-process communication is easiest when real-time processes reside on the same multi-processor computer, sharing memory and bus resources. When it is not possible for real-time processes to be contained on the same node, communication must be done over the Myrinet. This can lead to exchanges of large amounts of data in real-time. Over 400 channels of raw diagnostic data must be distributed on several different timescales, between 50 microseconds and a few milliseconds, to possibly multiple PCS nodes [1]. For example, the real time EFIT equilibrium reconstruction was until recently transferring several Kbytes of data every few ms between two separate nodes and four CPUs (now performed using shared memory between multiple CPUs on a single computer). Real-time processed data, produced by more complex processes such as Charge Exchange Recombination for ion temperature and rotation, Thompson Scattering for electron density and temperature, and mode spectrum analysis are passed through the real-time network from one node to another. A selected set of diagnostic signals along with EFIT flux contours and intensities of real-time computed MHD modes are passed to computers for plotting of signals and plasma boundary on displays in the control room [1].

The DIII-D model for plasma control is a series of applications or "real-time processes" running in parallel upon a networked cluster of computers. Each process is dedicated to a single CPU on one of the PCS nodes [1]. Real-time performance is achieved by a customized Linux kernel used for the OS of all PCS nodes [4]. Linux was a logical choice for an OS because it was supported for Alpha and Intel-based platforms. The recent upgrades to the PCS have included an increasing number of multi-core systems (giving four or eight real-time CPUs). The multi-core systems allow for an optimum grouping of real-time processes onto a single computer. This increases efficiency because inter-process communication can be handled across the internal bus rather than over the real-time network. It does pose problems when differing processes compete for system resources; organization is key.

# 3. PCS VARIANTS AND COLLABORATION

DIII-D has the unique experience of being involved in the development of multiple real time systems for control of fusion plasmas in devices worldwide, all based on core software developed for the DIII-D PCS [5]. A turnkey hardware and software PCS for the EAST tokamak in Hefei, China, was developed at GA, installed and commissioned on-site, and supported during its first use in experimental operations [5]. This multi-cpu system incorporated a Myrinet real-time network. The DIII-D PCS software for the KSTAR tokamak in Daejeon, Korea [6] was developed in collaboration with the National Fusion Research Institute. GA was closely involved in integrating this system with the reflective memory (RFM) network used by KSTAR for inter-plant communications and supported the commissioning of this system prior to and during the first plasma campaign. The DIII-D PCS software was also provided to the NSTX program at Princeton, U.S., MAST in Culham, UK, and to the Pegasus and MST devices, both at the University of Wisconsin in the U.S. Support for real-time communication was not provided for these installations. Occasional software support has been provided for NSTX. Table 1 shows some major characteristics of the PCS variants.

Adoption of the DIII-D PCS has led to collaboration and remote operation of other devices. Recently, DIII-D personnel participated in run campaigns at both EAST and KSTAR. Originally, participation for both was done on-site. However, during the last campaign at EAST, participation was done entirely remotely. This presented a completely different environment and set of issues to overcome.

Perhaps the largest hurdle to overcome was the inability at times to evaluate shot data. Access to remote data was painfully slow. Network bandwidth was limited to a tiny fraction of the large high-energy physics "pipe.” It became necessary to segment this data and use parallel sftp sessions to provide more bandwidth. The data sets were then reconstructed locally with the complete task taking several minutes. Another hurdle was a separate MDSplus tree containing the power supply data, which was not in a common or obvious format, which meant that certain signals were funneled through a narrow set of utilities, remotely executed at EAST. Access to data, stored in a standardized format and perhaps distributed to many peer facilities could have lessened the impact on the central site.

# 4. MAJOR LESSONS AND OBSERVATIONS.

The PCS computing hardware at most facilities centers around low cost, off-the-shelf (OTS) Intel-based processors. Availability and ruggedness have proven to be excellent. Highly commercialized CPUs, like the Intel family of CPUs, are in high demand. This guarantees continued upgrade paths and better performing CPUs with each subsequent upgrade. Widely used CPUs also allow for more widely available peripheral hardware such as motherboards, memory, disk drives, etc. Common commercialized equipment offer the widest array of available free/inexpensive software; i.e. OS, compilers, etc. Also, widely used hardware and software tend to have large global communities of users which can be tapped for help from things as simple as configuration to trouble-shooting.

Recent experience shows that IEEE floating-point processors, such as the Intel Xeon and Core family of processors, give the highest performance to cost ratio. Their successful use has been proven for years in many PCS variants. More generally, floating-point general purpose processors (FPs) are significantly easier to use for development and maintenance than specialized digital signal processors (DSPs). The most difficult processors to work with are those that require development of code on some other processor and then cross-compiled to download onto the real-time processor. DSPs are optimized to take advantage of “long vector” operations, of which there are relatively few in existing fusion control algorithms. It is therefore likely that general purpose processors are faster than the special purpose DSPs for the mix of computations used in real time plasma control. In addition, FPs can be accessed by standard operating systems and programmed with widely used languages and compilers. The flexibility that this gives the PCS allows the software to be easily altered for ongoing development of control codes. Fusion plasma control is expected to continue to evolve over the lifetimes of existing devices (and also during ITER’s lifetime). It is therefore expected that a significant reduction in manpower cost is provided by the choice of FPs in the PCS.

As increases in computing power are more widely used, more capable micro-processors filter down for use in front and back end peripherals of the PCS. Smart and programmable equipment, such as the D-tacq ADCs and DACs [7] have replaced the older more rigid CAMAC and VME counterparts at DIII-D. Simple programmable firmware alterations can be done on these devices to include simple tasks like baseline corrections, physic unit conversion or on the fly filtering. Work is underway in collaboration with KSTAR and D-tacq Inc. that will parallel input data from PCS ADCs onto a connected RFM network, thereby allowing other plant sub-systems to share the ADC data from the PCS in real-time. Such a technique allows for more easily distributed data to plant subsystems by reducing the need to digitize analog signals at multiple locations.

Networking is becoming an even more critical element of PCS design. A weak link here can affect unrelated systems and diminish overall performance. The network is also one of the most difficult items to replace once established. Because of the physical logistics, it may not be possible to swap the complete network or even sub-nets. DIII-D evaluated several possibilities before deciding upon the Myrinet. RFM gigabit Ethernet and Infiniband were also considered at the time. The decision in the end to go with Myrinet was due mainly to it's deterministic nature, fiber optic and copper cabling, supported form factors (PCI and PMC), supported on a Linux interface, and its use elsewhere in high speed networked facilities; e.g. super-computer centers. Although the quantity of data being transferred between real time processors in the DIII-D PCS is not as large as specified for the ITER synchronous data bus network [8], the data rates are generally more stringent, as is the latency requirement. However, the DIII-D PCS does not rely entirely on the real time data network for minimizing this latency. Shared memory between multiple CPUs on a single computer is being used more frequently now for processes that require large amounts of inter-processor communication.

The real-time connectivity of each PCS variant is different. For multi node systems, such as EAST and DIII-D, inter-process communication is handled via the internal system bus and an external real-time network. EAST and DIII-D use a Myrinet star configuration. RFM is also employed for real-time communication at KSTAR and EAST. Although output data is only sent over the RFM for EAST, both input and output data are received/transmitted on the KSTAR RFM. Stock device driver software from the manufacturer needed to be customized to remove the dependency upon interrupt driven communication handling. For EAST, a mixed type of real-time networking is used. The Myrinet is utilized for point-to-point communication between PCS nodes and the RFM is utilized for output commands to plant sub-systems. Only point-to-point communication is needed when two processing nodes share data. Broadcasting out over a general network would needlessly waste bandwidth. A general broadcast networking schema, such as RFM, may best serve information required by many nodes in ITER. Each type of network has strengths and ideally the PCS could be best served by having access to both. Additionally, segregated networks for raw diagnostic data and command data, i.e. outputs to actuators, may also be optimum from the standpoint of control. Command data should have higher priority, which may be impeded by secondary systems utilizing bandwidth on a general network.

The PCS variants utilize a few differing types of archival schemes. For collaboration purposes it has been found that a widely used scheme such as MDSplus offers some key benefits. Most new PCS sites use MDSplus for archiving PCS, general diagnostics and power systems data. Utilizing MDSplus allows the facility to devote efforts on other fronts while relying on a schema that has been proven and debugged by others. A widely used archiving scheme, like MDSplus, also provides the facility with software utilities that can expedite the creation of codes that insert and extract data from the archive. Work is currently underway to extend the MDSplus scheme to allow for long pulse experiments. If implemented, it may offer ITER a near turnkey archiving mechanism [9].

Although hardware at the onset of a PCS installation may be well defined to be a given type and configuration, events occur which may force changes in the original hardware. Changes to the system are inevitable, because stronger, more capable hardware is available or older hardware is no longer available, or because control needs of the device have grown. The longer the machine is in use, the more change it may have to endure. DIII-D has a varied use of mixed legacy hardware, e.g. VME and CAMAC, which is the result of evolving hardware standards. As time progresses, the most valuable part of the PCS is the time and effort that has gone into the evolution of algorithm software. As the hardware system has changed, it is ideal to preserve the algorithm codes with as little alterations as possible in order to maintain their integrity. The software architecture of the PCS tries to accomplish this preservation by de-coupling hardware dependent codes from algorithm coding as a general rule; especially with regards to I/O dependent devices. A modular software design helps accomplish this task. Standardization on a largely portable programming language, in the PCS case C, has also eased the maintenance of PCS software through its many cycles of hardware upgrades.

# 5. CURRENT ISSUES AND FUTURE UPGRADES

Porting the DIII-D PCS to additional devices has exposed the original design to issues that the DIII-D PCS would never have encountered given its original conceived usage. The PCS staff is addressing each of these issues as needed and future upgrades are planned. The most pressing near-term issue is a solution for long pulse archiving. The DIII-D tokamak is a short pulse device. Even as added demands for data acquisition have increased, these have been addressed with increasing memory usage. For long pulse devices, such as EAST and KSTAR, data will need to be streamed out of the PCS at near real-time speed because the current archiving scheme (storing data in processor memory until post-shot download) eventually will force a limitation either in total length or in desired sample interval. Real-time streaming of PCS data will also allow operators to evaluate shot situations at a faster pace while not having to wait for post-shot downloads. A planned solution to this issue is to dedicate a new process or CPU to handle solely the need of archiving. This process, connected to the PCS via a chosen real-time network, will not have to run in real-time but will need to perform at near real-time speeds in order to keep pace with archival demands. Because the process is not a real-time process, it does not have to operate under the same constraints of other PCS processes, which could give it disk and/or Ethernet access. It will act as a portal to the world outside the PCS passing data to users and the main archive. If demands on this single process grow too large, multiple parallel processes could be implemented. The current real-time data displays, used at DIII-D, EAST, and KSTAR are an example of just such a scheme. Data is streamed out of the PCS in near real-time speed, in all cases, for visual display. The real-time display at DIII-D routinely handles approximately one megabit of data per second with cycle times of 1 ms. This represents a relatively slow download given current DIII-D PCS operational standards. Other DIII-D real-time processes transfer larger amounts of data over the real-time network at faster cycle times with the maximum being 8 megabits per second from one process (used solely for ADC data collection, running at 250 µs.) to seven other PCS processes.

Another major issue brought about by recent implementations is the need for the PCS to not only monitor itself but also detect fault conditions of the various subsystems whose data the PCS is acquiring in it’s normal course of execution. Smart algorithms allow for programmable fault detection as well as dynamic setting of fault condition levels. Low-level work in this field has been ongoing at DIII-D but even though the PCS will monitor itself for internal fault conditions it was never intended to be the primary device safety system for monitoring other subsystems. This is due to the needed reliability of such a monitoring system, which is much higher than can be expected by the PCS. However, the PCS can assist in trying to prevent fault conditions and can provide a programmable set of actions to take in response. The benefit of the PCS monitoring, as demonstrated at EAST and KSTAR, is to have the PCS be a first level detection system, which could mitigate the situation at an early stage saving both equipment and experimental time. Expanded work is in progress to extend the PCS capabilities in the area of detection and mitigation [10].

One of the largest drawbacks among new PCS sites is the dependence of the PCS upon an expensive third party software package called Interactive Data Language (IDL). This package is mainly used by the PCS graphical user interface (GUI). At DIII-D, there is a long and extensive use of IDL. This spreads out the developmental costs in IDL amongst many applications and justifies its use. DIII-D also benefits from a licensing deal, which perpetuates its use in the PCS at essentially no cost. New sites however, that need to invest in IDL, are surprised by the cost, especially at a time when free or inexpensive GUI builders and languages abound. A new PCS site often will show no interest in an expanded use of IDL other than by the PCS, which could mitigate the cost. The use of IDL is contradictory to efforts in moving away from proprietary technology and towards more OTS and open source products. Work was started this last year to create a simple non-GUI interface to the PCS. This interface, currently only textual, was originally developed to aid remote support because of the extreme inconvenience of running the GUI on a distant host. In the long term, additional work may be done in this area to move away from IDL and perhaps even towards interfaces developed in more mainstream languages such as Java or Tcl/Tk.

During the most recent support of EAST and KSTAR, network security at endpoints of the remote collaboration hindered access for data and control by placing obstacles in the path of remote users. For example, all users were funneled through a privileged machine with a login destination perhaps several hops away. Simple tasks, such as copying files to remote systems became a lengthy multi-step process. In general, “ssh tunneling” was used for connecting to remote servers. Extra steps had to be taken well in advance to verify access, proper ports were open and security keys in place. The “operations request gatekeeper” is a security interface for ITER, which is intended to address these security issues while allowing remote users to gain access to internal plant systems [11]. A prototype of such an interface will be implemented at one of the PCS sites to support remote collaboration.

# 6. SUMMARY

The DIII-D PCS is a proven model for Tokamak control. It has been used successfully at DIII-D for almost two decades and has survived through multiple cycles of hardware upgrades. It has been ported to several other Tokamak devices. In doing so, it has proven itself upon multiple types of hardware and under varying control scenarios; the most recent being plasma start-up with super-conducting tokamaks. The expanded implementation of the DIII-D model has given DIII-D personnel increased experience and understanding of plasma control issues at multiple Tokamak sites. This paper has identified many practices, which over time, have proven to be wise choices: the use of OTS computing hardware, floating-point processors and open source software usage and design. The continued use of the PCS has also brought about insight into other areas touched upon by the PCS: real-time networks, remote collaboration, and data archival. Work is ongoing in additional areas where clear needs are present: most notably in open-source GUI development, long pulse archiving, and remote participation.

# REFERENCES

[1] D.A. Piglowski, et al., Enhancements in the second generation DIII-D digital plasma control system, 24th Symp. on Fusion Technology, 2006.

[2] B. Penaflor, et al., Real-time control of DIII-D plasma discharges using a Linux alpha cluster, 21st Symp. on Fusion Technology, 2000.

[3] J.R. Ferron, et al., A flexible software architecture for tokamak discharge control systems, Proc. 16th IEEE/NPSS Symp. on Fusion Engineering, 1995 (Institute of Electrical and Electronics Engineers, Inc., Piscataway, 1996) vol. 2, p. 870.

[4] B.P. Penaflor, et al., Real-time data acquisition and feedback control using Linux Intel computers, Proc. 5th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, 2005.

[5] B.G. Penaflor, et al., Worldwide collaborative efforts in plasma control software development, Proc. 6th IAEA Technical Committee Meeting on Control Data Acquisition and Remote Participation for Fusion Research, 2007.

[6] Sang-hee Hahn, et al., Plasma control system for "day-one" operation of KSTAR tokamak, 25th Symp. on Fusion Technology, 2008.

[7] Intelligent Data Acquisition Boards and Systems, D-TACQ Solutions Ltd, East Kilbride, Scotland UK; www.d-tacq.com.

[8] CODAC Conceptual Design, February 2008, www.iter.org

[9] T. Fredian, et al., MDSplus extensions for long pulse experiments, Proc. 6th IAEA Technical Committee Meeting on Control Data Acquisition and Remote Participation for Fusion Research, 2007.

[10] B. Sammuli, et al., Approaches to tokamak off-normal event detection and response at DIII-D, KSTAR, and EAST, this conference.

[11] D.P. Schissel, et al., An investigation of secure remote instrument control, this conference.

# ACKNOWLEDGMENT

This work supported in part by the U.S. Department of Energy under DE-FC02-04ER54698 and DE-AC02-09CH11466.