GA-A24815

AUTOMATIC FAULT-CHECKING SYSTEM ON THE DIII-D TOKAMAK

by
J.T. SCOVILLE and M.L. WALKER

AUGUST 2004

DISCLAIMER

GA-A24815

AUTOMATIC FAULT-CHECKING SYSTEM ON THE DIII-D TOKAMAK

by

J.T. SCOVILLE and M.L. WALKER

This is a preprint of a paper to be presented at the 16th ANS Top. Meeting on the Technology of Fusion Energy, Madison, Wisconsin, September 9-16, 2004, and to be printed in the *Proceedings*.

Work supported by
the U.S. Department of Energy
under DE-FC02-04ER54698

GENERAL ATOMICS PROJECT 30200
AUGUST 2004

AUTOMATIC FAULT-CHECKING SYSTEM ON THE DIII-D TOKAMAK

J.T. Scoville and M.L. Walker

General Atomics, P.O. Box 85608, San Diego, CA, 92186-5608,
Scoville@fusion.gat.com

Modern tokamaks are highly sophisticated devices consisting of a large number of state-of-the-art systems that must function in unison to obtain a successful plasma discharge. An unsuccessful discharge can result if one or more systems fail, and diagnosis in an efficient and timely manner can be difficult. The resulting reduction in tokamak availability and productivity can be expensive, justifying a significant effort for automated fault diagnosis.

For the DIII-D tokamak, a software system has been used for the past 5 years to automatically monitor and test the performance of hundreds of tokamak systems. The Fault Identification and Communication System (FICS) is automatically triggered to run immediately after each tokamak discharge and report its results via a simple color-coded graphical user interface. In addition to saving the operator time, the significant advantage of FICS is its ability to detect insipient faults that could lead to future machine failures. It has been estimated that FICS has saved an average of one to two shots per day, which equates to approximately 5% of all DIII-D pulses. The significant experience gained through the development and use of this post-discharge analysis tool also provides insight into future methods for on-line process monitoring of steady state devices

I. INTRODUCTION

Fusion energy research, by its nature, employs the latest capabilities of modern technology. Investigating the physics of the magnetic confinement of a plasma typically involves the use of a large collection of highly sophisticated equipment - the tokamak and all of its auxiliary systems. Routine operation requires that most of the systems function within an acceptable range of operating parameters. Failure of one or more systems can lead to an unacceptable degradation of performance. As the complexity of the interacting machine systems increases, it becomes obvious that an automated fault-checking system is necessary to help ensure maximum productivity. Future devices such as ITER promise to be even more complex, suggesting the importance of incorporating fault diagnosis systems into the basic design of the facility.

The primary mechanisms for protection against equipment failures are mostly conventional and have been integrated into the equipment at research facilities as they have evolved. These protection systems are usually based on hardware, intentionally excluding software, and are as robust and simple as possible to maximize reliability and failsafe operation. Routinely monitoring the hundreds of interacting systems of a tokamak, however, requires a system that can adapt easily to frequent configuration changes. Such a system is most easily employed using intelligent software that has access to the diagnostic data provided by the multiple machine subsystems. A valuable feature of this kind of system is its ability to forecast and issue warnings about impending problems caused by operating near system limits on temperatures, coil torques, and current levels, for example.

The Fault Identification and Communication System (FICS) developed for automatic fault-checking of DIII-D systems is based on the C Language Integrated Production System (CLIPS)[1], inference engine software in the public domain that was developed by the Software Technology Branch at NASA. For each subsystem, the execution of the FICS software is driven by the availability of data following a tokamak discharge, therefore resulting in minimal delay before the start of processing. The data-driven feature also makes it relatively straightforward to add new system fault detection algorithms, with no need to modify the logic structure, keeping FICS current with changes in device operation.

FICS has been in routine use on the DIII-D tokamak since 1999[2]. The system has expanded considerably since its inception and now performs routine tests on a variety of systems including power systems, computer systems, magnetic field coils, vacuum systems, gas injectors, plasma diagnostics, auxiliary heating systems, other fault detection systems, and even plasma parameters such as the shape and impurity content. The diagnosis of an obvious fault after a discharge is often performed more quickly by experienced operators, but FICS detects secondary faults that human operators routinely miss. In addition, the large number of routine tests run by FICS far exceeds the capacity of the operators. Relegation of these routine tests to an automated system also allows the operators to concentrate on other more sophisticated tasks requiring their attention.

II. FICS SOFTWARE STRUCTURE

FICS was created to assist the operator in the post-discharge diagnosis of tokamak systems performance. Criteria driving the design included ease of use, automatically triggered analysis, simple point and click interrogation, and unambiguous display of results. These guidelines were met using software modules that reside in one of four categories: the graphical user interface (GUI), the CLIPS inference engine, C and FORTRAN data retrieval and processing, and the plotting package. The hierarchy and interaction of these groups of software is illustrated graphically in Fig. 1.

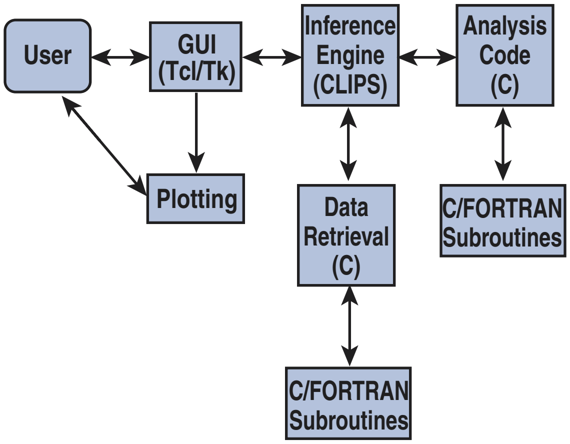
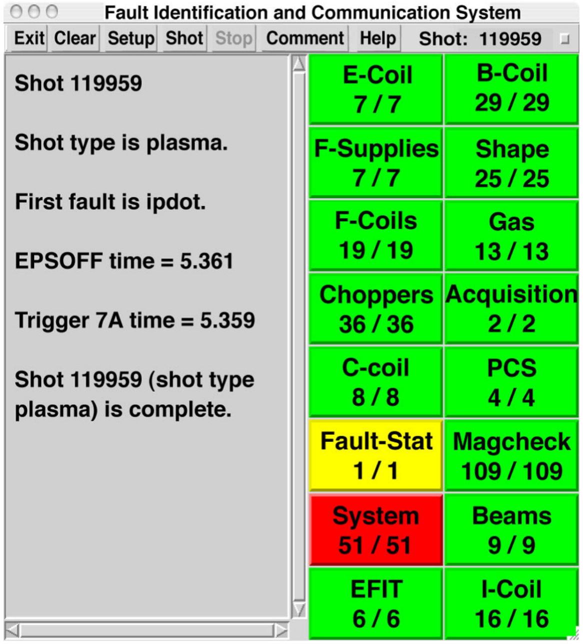


Fig. 1. Hierarchy of the FICS software modules

The GUI employed by FICS is written in Tcl/Tk public domain software[3]. Results of all the automatically spawned data analyses are reported through this interface. The logic used to control the retrieval and analysis of data and infer the presence of system faults from the analysis is handled by the inference engine, written in CLIPS. Data retrieval is triggered by the incrementing of the shot number, which occurs immediately after a plasma discharge. Actual data retrieval and processing is accomplished using the lowest level subroutines, written in C and FORTRAN. Once the data has been processed and analyzed, and results reported to the GUI, the operator can obtain more detailed information by clicking on various parts of the user interface.

II.A. The User Interface

The top-level graphical interface consists of a simple array of boxes that also function as buttons, and a large area for text messages (see Fig. 2). Each box represents one of the primary tokamak systems (or categories), which include various coils and power supplies, computers, diagnostics, auxiliary heating systems, data acquisition, the plasma control system (PCS), and others. A brief description of the subsystems tested and the number of tests run in each category are listed in Table I. The tests range widely in sophistication, from something as simple as checking a thermocouple reading to more complicated tests that fit systems of equations to check for the correct magnetic field topology.



Shot 119959

Shot type is plasma.

First fault is ipdot.

EPSOFF time = 5.361

Trigger 7A time = 5.359

Shot 119959 (shot type plasma) is complete.

Fig. 2 Top-level FICS GUI, showing the 16 major system categories

TABLE I. FICS System Categories

System name	# tests	Subsystems tested
E-coil	7	Ohmic heating coil and plasma current control
B-coil	29	Toroidal field coil systems
F-Supplies	7	DC power supplies that charge choppers on F-coils
Shape	25	Real-time plasma shaping feedback controls
F-coils	19	Poloidal field shaping coils
Gas	13	Gas fueling injectors and density control
Choppers	36	Power converters driving F-coils
Acquisition	2	Diagnostic data acquisition
C-coil	8	External non-axisymmetric radial field coils
PCS	4	Plasma Control System
Fault-Status	1	Machine hardware interlock fault system
Magcheck	109	Magnetic field diagnostics
System-Status	51	Miscellaneous
Beams	9	Neutral beam auxiliary heating systems
EFIT	6	Equilibrium reconstruction software
I-coil	16	Internal non-axisymmetric radial field coils

All FICS analysis and subsystem test results are summarized and reported under one of the 16 major tokamak system buttons and displayed on the top-level GUI shown in Fig.2. Clicking on one of the category buttons opens another array of boxes that represent the subsystems being tested within the category. In some cases the subsystems also have subsystems, and these nested levels can be two or three layers deep. Clicking on a button on the bottom level finally opens a text box with detailed information about all tests run for that specific subsystem.

II.A.1. Color Code

The simple and unambiguous reporting of all algorithm test results is accomplished on all levels using color-coding. The operator can immediately ascertain the success or failure of all tests in each category with a glance at the top-level GUI. The color codes and priorities adopted for reporting test results are shown in Table II. All buttons are initialized to blue, the lowest priority, at the beginning of a discharge analysis. When the inference engine posts a higher priority message, the button turns the color associated with that priority. These priority levels also immediately propagate upward through the subsystem hierarchy. This priority system ensures that a button representing a system category can only remain green (i.e., “good”) if every subsystem within the category is successfully tested and all tests are passed. An error result for any subsystem test causes the top-level button to turn red. If the worst subsystem test result is less severe, a warning color of yellow is reported at the top level. In the event that one or more tests cannot be run (because data could not be obtained, for example), the button will turn gray, unless a warning or error result is reported from another subsystem within the same category. A white button for a subsystem indicates that all tests within that subsystem are intentionally ignored or not needed. (For example, an unused power supply will have a white button). The presence of a blue button indicates that the tests have not yet completed and therefore results are not yet available.

TABLE II. Test Results Color Code

Color	**Priority**	**Meaning**
Red	1	Error
Yellow	2	Warning
Gray	3	No test possible
Green	4	Passed
White	5	Ignored
Blue	6	Not finished

II.A.2. Controlling a FICS Session

In addition to reporting test results for all the system categories, the top-level GUI shown in Fig. 2 has additional features that are invoked by clicking one of the smaller control buttons along the top of the window. The functionality of each button is listed in Table III.

The FICS session can be customized and controlled via the control buttons. For example, the user can launch a FICS run for an old shot or stop the automatically launched session for the current shot. Comments about the FICS code, bug reports, and suggestions for future additional algorithms can be sent to the programmers via e-mail. The system can be set to verbose mode, allowing hundreds of normally hidden messages to be displayed in the text area (primarily for programmer’s use).

TABLE III. Functionality of Control Buttons

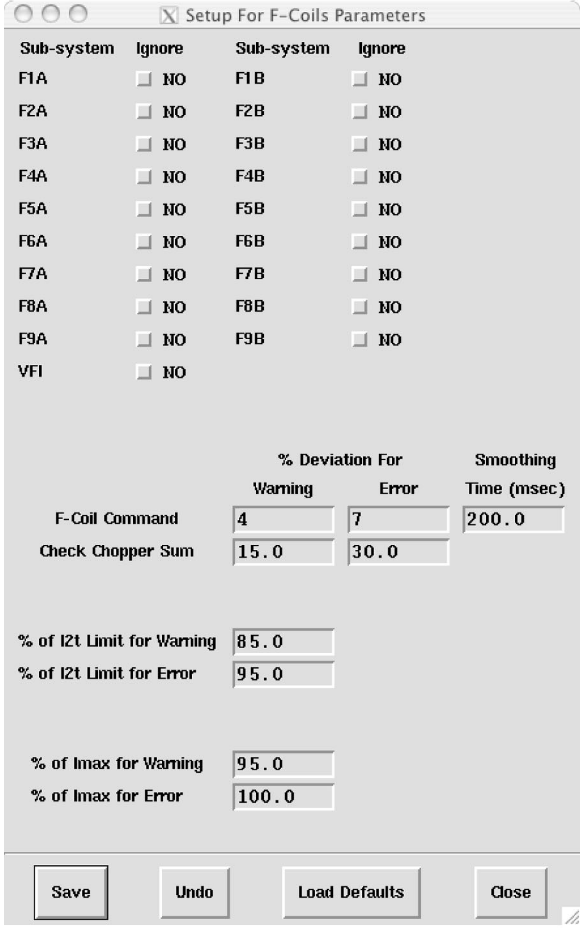
Control Button	**Function**
Exit	Exit FICS
Clear	Clear text box
Setup	Customize test parameters
Shot	Run FICS for an old shot
Stop	Stop processing current shot
Comment	E-mail comments to FICS programmers
Help	Display general help file
Small square, upper right	Verbose mode

The most powerful of the controls is the Setup button. Clicking on Setup brings up a multi-level window that presents the user with a large number of global variable entries, via buttons or text boxes (see Fig. 3). From this page the operator can choose which categories will be tested and specify many of the test parameters, such as the data-smoothing time constants and warning and error levels for various tests. An example is shown in Fig. 3, which displays many of the adjustable test parameters used in the F-coil category. The unique FICS environment created with these controls is saved separately for each machine operator.

II.B. The Inference Engine

The core of FICS is the CLIPS code controlling the flow of processing during post-discharge execution. CLIPS was chosen for this function because it was written to facilitate expert system implementation and has two powerful capabilities. *Chaining* is the ability to link the execution of software modules in a logical progression based on decisions made in previous steps. *Data driven* execution means that the “rules” defined by the expert system can be triggered as soon as the necessary data has been retrieved.

In their simplest form, the rules executed by CLIPS have the structure “If A, then B”. The set of facts that define “A” typically include an assertion that the necessary data has become available. The reaction “B” usually involves processing and testing the data by executing some of the lowest level C and FORTRAN codes. CLIPS uses the results returned by the data



Sub-system	Ignore	Sub-system	Ignore
F1A	☐ NO	F1B	☐ NO
F2A	☐ NO	F2B	☐ NO
F3A	☐ NO	F3B	☐ NO
F4A	☐ NO	F4B	☐ NO
F5A	☐ NO	F5B	☐ NO
F6A	☐ NO	F6B	☐ NO
F7A	☐ NO	F7B	☐ NO
F8A	☐ NO	F8B	☐ NO
F9A	☐ NO	F9B	☐ NO
VFI	☐ NO		

	% Deviation For Warning	Error	Smoothing Time (msec)
F-Coil Command	4	7	200.0
Check Chopper Sum	15.0	30.0	

% of I2t Limit for Warning	85.0
% of I2t Limit for Error	95.0
% of Imax for Warning	95.0
% of Imax for Error	100.0

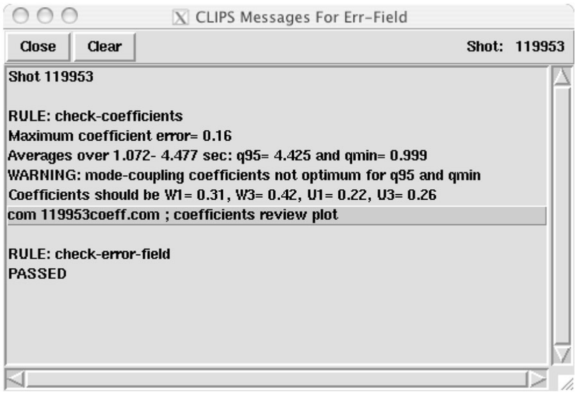
Fig. 3. The setup interface for the F-coils category

processing routines to assert additional facts in the last part of the reaction “B”. These new facts cause the execution of more rules and the process continues until all the tests have been completed.

Since FICS is a large collection of software and must analyze hundreds of sets of diagnostic data serially, it is imperative that it be as efficient as possible. The data driven feature is implemented using a queue to ensure maximum efficiency. Upon algorithm initiation, facts are asserted to request the data sets needed for the system test rules. Each of these data-request facts triggers a separate instance of a data acquisition rule. These rules enter the queue in the order generated, unless they are given differing salience (priority) values. If several rules are ready to execute simultaneously, the salience determines the order of execution. When a data acquisition rule reaches the top of the queue, the rule is triggered and attempts to acquire the requested data. If the result is “data not yet available”, the data-request fact is reasserted, and entered again into the bottom of the queue. If the data acquisition rule retrieves data successfully, a “data-ready” fact is asserted, prompting the execution of the system tests that require that data. When the queue is empty (no more data is needed), FICS terminates normally and waits for the next shot. A time limit is also used, in case some data never becomes available. In that event, the subsystem tests that cannot be performed will cause the display of a gray button at the completion of the FICS run.

II.C. Data Processing

The successful retrieval of data using a data acquisition rule triggers the execution of low-level subroutines in the C and FORTRAN libraries. A large collection of modular data processing routines has been assembled to provide the functionality necessary for efficient analysis. In addition to the many specialized data retrieval routines, there are modules for mathematically manipulating arrays of data, comparing data, windowing, filtering, and sampling the data. The purpose of this library of routines is to maximize efficiency, retrieving and processing quickly only the data that is necessary.



```
Shot 119953

RULE: check-coefficients
Maximum coefficient error= 0.16
Averages over 1.072- 4.477 sec: q95= 4.425 and qmin= 0.999
WARNING: mode-coupling coefficients not optimum for q95 and qmin
Coefficients should be W1= 0.31, W3= 0.42, U1= 0.22, U3= 0.26
com 119953coeff.com ; coefficients review plot

RULE: check-error-field
PASSED
```

Fig. 4. Text box example showing test details for the Error Field category

II.D. Data Visualization

Clicking on the lowest level GUI colored button for a subsystem brings up a text box with detailed messages about the tests run for that subsystem (see Fig. 4). Each rule is represented by at least two lines of text, one containing the name of the rule and one listing the result of the test (e.g., “passed”, “warning” or “error”). Clicking on the first line (the rule name) opens a help file, displaying a detailed description of the test performed by the rule and explaining what the test results mean. If a rule reports a warning or error, more lines of information are displayed describing the problem. Fig. 4 shows an example of a warning level discrepancy found in the Error Field subsystem that is contained within the C-coil category. In this example, the mode-coupling coefficients used for multi-mode error field correction (C-coil current control) were not set to optimum values for the measured range of safety factor, q. For most rules, an additional final line of text is written listing the plot-file that was created. Clicking on this last line for the rule (shown highlighted in Fig. 4) opens the plot-file and spawns a process to graphically display the data in a conventional X-Y plot, where X is usually time. The plot feature makes it very easy for the operator to see graphically the nature of the problem reported by FICS.

III. FUTURE DIRECTIONS

FICS was installed on the DIII-D tokamak in 1999 and has been continuously revised and upgraded since then to keep pace with machine modifications. Planned future upgrades, for example, include monitoring and testing the new automatic diagnostic data fitting and analysis routines that are currently being brought online.

FICS is strictly an asynchronous between-shot analysis and does not run continuously in real-time. This mode of operation is acceptable for a machine like DIII-D that is only pulsed for approximately 10 seconds every 12-15 minutes. Next generation devices like ITER, however, and future steady-state machines, will be at much greater risk of damage or prohibitively expensive loss of performance from system failures. The occurrence of off-normal events such as catastrophic loss of plasma or failure of cooling systems, for example, must be mitigated by on-line safety systems and interlocks. The detection of slow performance degradation or an accumulating variance in system response, however, will require constant monitoring of all systems by an intelligent, interactive fault-checking system. Such a system could be achieved by an extrapolation to real-time of a post-shot analysis software package similar to FICS.

The explicit modularity of FICS means the addition of a new system test rule almost never requires any code modifications beyond addition of the rule itself. FICS already provides the mechanism for real-time unambiguous communication of errors requiring human intervention. The use of saliences allows the highest priority rules to execute first and the color priority enables immediate propagation of the most serious alarms to the operator. The FICS software does not presently run as a continuous process however, and it is explicitly asynchronous because it is intended as a post shot analysis code. The primary modification required for an extension to steady state process monitoring would be the synchronization of FICS with the real-time data acquisition system, to allow repetitively re-acquiring and applying the various system test rules to this data.

IV. SUMMARY

The continuously increasing complexity of the DIII-D tokamak has led to the development of the FICS automatic fault-checking software system. Hundreds of tests are performed during between-shot analysis to check for erroneous or degraded system performance. Results are reported to the user interface in a simple color-coded array of panels that also function as category buttons. Clicking on a category exposes layers of subsystems and more detailed information about the test results for each subsystem. Help files, plot files, and customization of the FICS environment are available with the click of a button.

Routine use of FICS has relieved the tokamak operators of the burden of tediously checking system performance between shots. FICS performs a much larger number of routine system tests than can be accomplished by the operators and, as a result, failures, operation near limits, and deviations from optimum performance are detected that would otherwise have gone unnoticed. The use of FICS on DIII-D has led directly to higher machine availability and productivity. The knowledge and experience acquired from the implementation and use of the FICS post-discharge fault detection software is directly applicable to the perceived future requirement for on-line process monitoring of steady state tokamaks.

ACKNOWLEDGMENTS

Work supported by the U.S. Department of Energy under DE-FC02-04ER54698.

REFERENCES

[1] J.C. GIARRATANO and G. RILEY, *Expert Systems: Principles and Programming*, PWS publishing co., Boston (1998).

[2] M.L. WALKER, J.T. SCOVILLE, R.D. JOHNSON, A.W. HYATT, and J. LEE, "Automated Fault Detection for DIII-D Tokamak Experiments", *Proc. 18th IEEE/NPSS Symp. on Fusion Engineering*, Albuquerque, New Mexico, October 25-29, 1999, p. 539, IEEE (1999).

[3] J.K. OUSTERHOUT, *Tcl and the Tk Toolkit*, Addison-Wesley Professional (1994).