

**GA-A23246**

**AUTOMATED FAULT DETECTION FOR DIII-D  
TOKAMAK EXPERIMENTS**

**by**

**M.L. WALKER, J.T. SCOVILLE, R.D. JOHNSON, A.W. HYATT, and J. LEE**

**NOVEMBER 1999**

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

GA-A23246

# **AUTOMATED FAULT DETECTION FOR DIII-D TOKAMAK EXPERIMENTS**

by

**M.L. WALKER, J.T. SCOVILLE, R.D. JOHNSON, A.W. HYATT, and J. LEE<sup>†</sup>**

This is a preprint of a paper to be presented at the 18th IEEE/NPSS Symposium on Fusion Engineering, October 25–29, 1999, Albuquerque, New Mexico, and to be published in the *Proceedings*.

<sup>†</sup>University of California, San Diego

**Work supported by  
the U.S. Department of Energy  
under Contracts DE-AC03-99ER54463, and  
Grant DE-FG03-86ER53266**

**GA PROJECT 30033  
NOVEMBER 1999**

# Automated Fault Detection for DIII-D Tokamak Experiments\*

M.L. Walker,<sup>1</sup> J.T. Scoville,<sup>1</sup> R.D. Johnson,<sup>1</sup> A.W. Hyatt,<sup>1</sup> and J. Lee<sup>2</sup>

<sup>1</sup>General Atomics, P.O. Box 85608, San Diego, California 92186-5608

<sup>2</sup>University of California, San Diego, California

## Abstract

An automated fault detection software system has been developed and was used during 1999 DIII-D plasma operations. The Fault Identification and Communication System (FICS) executes automatically after every plasma discharge to check dozens of subsystems for proper operation and communicates the test results to the tokamak operator. This system is now used routinely during DIII-D operations and has led to an increase in tokamak productivity.

## I. INTRODUCTION

During experimental operation of the DIII-D tokamak, hundreds of subsystems must operate correctly and simultaneously for a completely successful plasma discharge (shot). In the past, verifying proper operation of the subsystems which were most prone to failure was done manually by human operators after the shot. Because of the tedious nature of this task and the large number of systems, inoperative or malfunctioning systems were sometimes not detected until several experimental discharges had passed. Occasionally, problems were not detected until days later if the failure was not catastrophic.

During 1998, an automated fault detection software system was developed and used during 1999 plasma operations. The fault identification and communication system (FICS) executes automatically after every plasma discharge to check dozens of subsystems for proper operation and communicates the test results to the operator. It attempts to identify the source of any experimental discharge failure as well as system anomalies which could cause a future plasma discharge to fail.

The only commercial software required to implement this system are C and FORTRAN compilers. The user interface and the expert system which comprises the core of the fault detection software use freely available software packages. An optional plotting package can be invoked from within FICS which uses DISSPLA™ graphics. The FICS software system to be described here is now used routinely during DIII-D operations and has led to an increase in tokamak productivity.

## II. FICS OVERVIEW

The primary purpose of FICS is fault detection and fault prediction. Fault detection refers to determining which systems were not working properly during an experiment, even if they do not cause the loss of the experimental discharge (shot). For example, bad diagnostics will not cause

the loss of a shot but they can severely reduce its value for physics purposes. Fault prediction refers to determining which systems look like they are having trouble and may cause a future fault. This includes detecting programming errors, i.e. determining whether the operator specified setup for a shot is self-consistent. A secondary use of FICS is as a problem evaluation tool; its ability to automatically generate plots of detected or suspected problems facilitates the process of diagnosing problems by tokamak operators. A long term goal of the FICS effort is fault identification, i.e. determining the root source of problems rather than just detecting their symptoms.

FICS provides a graphical user interface which shows the status of selected systems with warning and alarm indicators identifying potential problems. It operates autonomously, with operator interaction only upon receipt of an alarm or trouble indicator. Tests of various tokamak subsystems are written so that false alarms, i.e. alarms which do not correspond to real problems, are minimized. This means that occasionally real problems will be missed. This is the classical tradeoff in detection problems. In the first version of FICS, a multitude of simple system checks are made but there is no active reasoning to identify root causes of problems.

## III. USE OF FICS DURING TOKAMAK OPERATIONS

The FICS code is started up every morning during experimental operations. It waits until a new shot is completed, then begins testing for faults. Figure 1 shows the front panel display for the FICS system. The colored buttons on the front panel represent broad categories of tokamak subsystems. The system tests represented by each button are summarized in Table 1.

Buttons along the top row control FICS execution. *Exit* causes the FICS execution to stop and the program to exit. *Clear* is used to clear the main message display window (left half of display in Fig. 1). *Setup* opens up a system of menus in which parameters which help define how tests on various subsystems are performed may be modified. The *Shot* button allows the operator to execute the FICS code on a past shot. *Stop* causes the currently executing shot to stop. It is not active in Fig. 1 because execution for the shot has already completed. The *Comments* button provides a mechanism for an operator to report a bug in the software or to request a new capability or system test.

Initially all system test buttons on the FICS display are blue. Alarms are raised in the form of color changes to these front panel buttons. The meaning of the various button colors

\*Work supported by U.S. Department of Energy under Contract No. DE-AC03-99ER54463 and Grant No. DE-FG03-86ER53266.

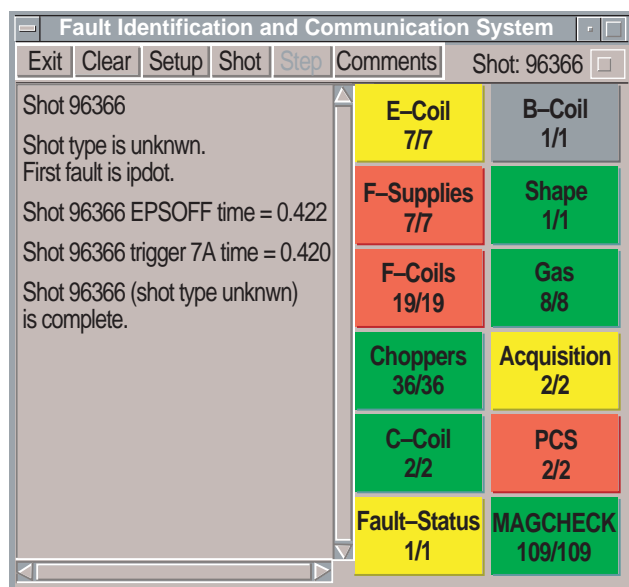


Fig. 1. FICS front panel display for shot 96366.

are summarized in Table 2. The colors are listed in order of increasing priority, i.e. if there are multiple decisions (e.g. based on multiple subsystem tests) which would produce more than one distinct color, the color with highest priority is the one actually displayed. For example, if a minor problem (yellow) was detected in an F-Coil subsystem by one FICS test, but another test detected a serious (red) problem, the F-Coils button would turn red. Numbers on each button in Fig. 1 represent the number of tests completed out of the total number of tests for that system.

Each major tokamak system category represented on the front panel display is comprised of a number of smaller subsystems. A click of the mouse on a front panel button brings up another panel of buttons representing the collection of subsystems within the major system. Figure 2 shows the panel of buttons which appears after pressing the main panel *F-Supplies* (red) button. This is taken from a particular shot which can be used to illustrate a problem that FICS detected.

Table 2. Button colors used to represent various alarm levels.

Color	Meaning
blue	tests for the system have not completed
white	tests for the system not applicable for this shot
green	system has passed all tests
gray	some tests of the system were unable to complete (e.g. required data unavailable)
yellow	a problem with the system was found
red	a serious problem with the system was found

DIII-D has seven dc supplies called F-Supplies which charge the capacitor banks of the plasma poloidal shape control power convertors (choppers). In Fig. 2, the D1 button is red because that power supply failed to come on for the shot 96366. When the operator presses the D1 button, the message window corresponding to the D1 F-Supply, shown in Fig. 3, comes up. The D1 system actually failed two tests (rules) as the message on the D1 button in Fig. 2 indicate. More detail about these failures is shown in the message window in Fig. 3. In most cases where a system test has failed, a command file is generated for a subsequent plot to illustrate the problem for the operator. Double clicking on the second line beginning with “com” in the message window (Fig. 3) brings up the plot shown in Fig. 4. This plot shows that the D1 supply never came out of inversion (D1 voltage never > 0).

The main window display (Fig. 1) indicates major problems (red buttons) also for the F-Coils category and the PCS category. The F-Coil test failures were caused in part by the failure of the D1 supply and in part by a problem in the timing of a chopper power convertor turn on time. The PCS (Plasma Control System) test failure was due to an operator error in programming one of the PCS waveforms. There are also three yellow buttons on the main panel indicating less severe errors. We will not discuss these here. Note the gray button for the B-Coil category. This shot was terminated by the D1 fault at about 400 ms into the discharge; this provided insufficient data for the B-Coil test, hence the gray button. The D1 power supply fault was an obvious problem which was easily identified by DIII-D operators, while the problems

Table 1. Tokamak system tests represented in FICS front panel display.

System Category	Represents
E-Coil	systems used in developing and maintaining plasma current
B-Coil	systems used in developing and maintaining toroidal field
F-Supplies	dc power supplies used to charge voltage in chopper capacitor banks
Shape	systems used to produce the correct plasma shape; tests conducted are to verify correct shape is produced
F-Coils	plasma shaping coils
Gas	systems used to fuel plasma and maintain density
Choppers	power convertors used to drive plasma shaping coils (F-coils)
Acquisition	systems used to acquire data from plasma diagnostics
C-Coil	systems used to produce radial fields needed for correction of inherent error fields and non-axisymmetric plasma control
PCS	Plasma Control System software and hardware
Fault-Status	machine interlocks, hardware fault reporting system
MAGCHECK	results of an independent program which looks for magnetic diagnostics which are zero, have saturated, or have the wrong sign

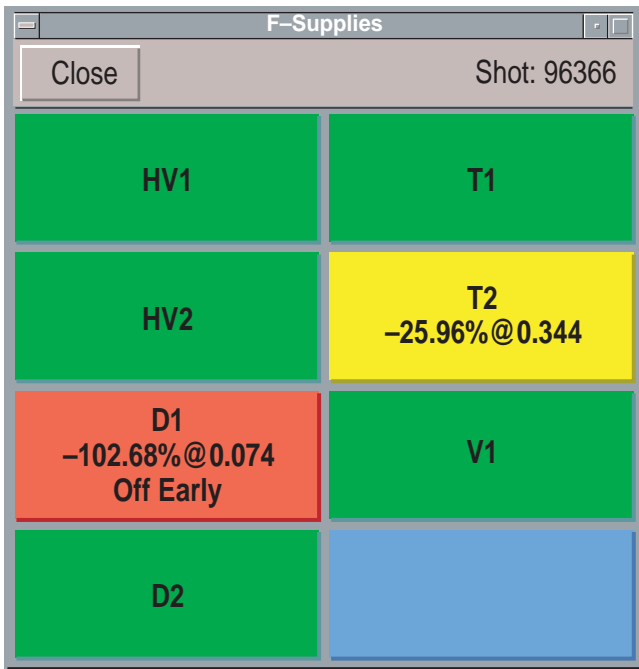


Fig. 2. Button panel for the F-supplies category.

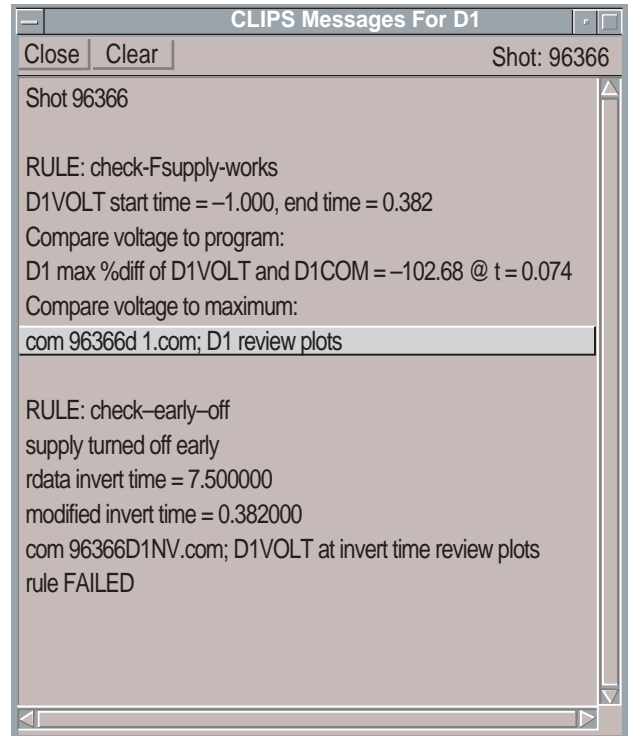


Fig. 3. Message window for the D1 subsystem.

with the PCS and those with the F-Coils which were unrelated to the D1 fault likely would not have been detected until they had caused subsequent discharges to fail.

#### A. Operator Customization

Operators have the capability of telling FICS not to execute tests for certain subsystems. This may be done for example, if the use of a particular subsystem for an experimental day is unusual so that the FICS test is no longer meaningful, or they may decide to operate with known system problems and don't wish to be continually reminded of those problems. Pressing the "setup" button (Fig. 1) brings up an interface with which an operator can customize the execution of FICS. This includes turning off certain sets of system tests, determining whether to produce plots when tests fail, and setting global variables which influence how particular rules operate. For example, threshold values and parameter ranges for many warning and error indications are user-settable.

#### B. Examples of System Tests

In this section we list several examples of system tests performed by FICS immediately after an experimental discharge (shot). Only those tests which are simple to explain, and which have counterparts for all tokamaks are presented here.

- Tests in which FICS compares requested or programmed values of tokamak or plasma parameters with values actually achieved: Checking that measured plasma current is following the programmed waveform, that power supplies follow their commands, and that gas valves follow their commands.
- Tests that hardware is operating correctly: Check that multiple choppers in parallel on each F-coil share current

- through the coil approximately equally. Check that choppers are actually working, i.e. putting out current and voltage. Check that power supplies don't turn off earlier than programmed.
- Tests for systems approaching limits: Typically, when a system exceeds a limit, it will trip a hardware fault detection circuit so often no alarm is needed. However, those near limits will often cause a later shot to fail. Check for coils near or over limits and for power supply currents near or over limits. Check that temperatures at various machine locations do not approach or exceed safety limits.
- Tests that plasma shape achieved is what was expected: Check if plasma went vertically unstable. Check that shape is approximately achieving programmed values. Check that shape is well away from outer wall.
- Check for operator programming errors: Check that shape has not been programmed to bump into a wall. Check that at least one gas valve has been programmed to turn on. Check that gas commands are in a valid range.
- Check for problems with data: Check for acquisition errors (data never shows up or is bad), or for data that is not acquired at the proper time or with the proper sampling rate for the experiment of the day.

## IV. SUMMARY OF FICS SOFTWARE

The FICS software consists of four main parts:

1. The user interface and supervisory control code, written in Tcl/Tk [1].

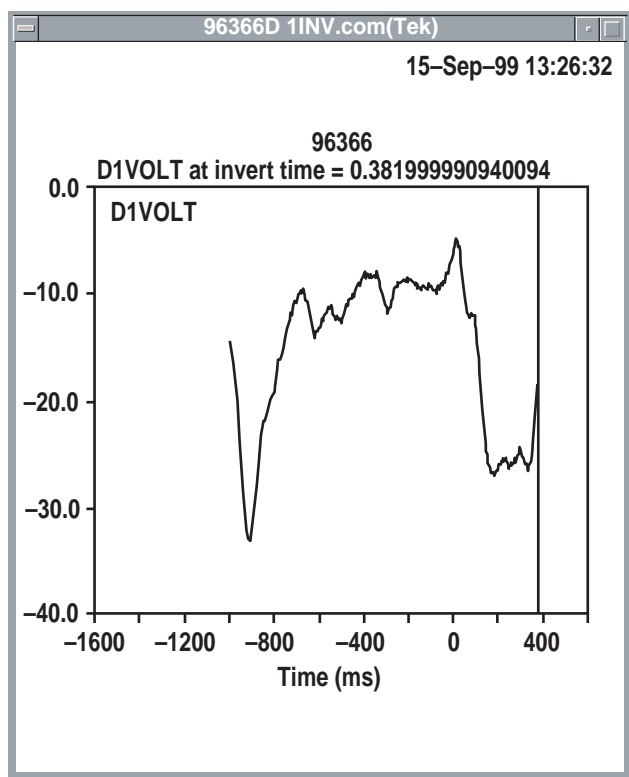


Fig. 4. Plot of data used to detect D1 subsystem failure.

2. The production expert system written in CLIPS [2].
3. The low level data extraction and manipulation routines written in C and FORTRAN.
4. The graphics display software. In this case, an existing standalone in-house graphics program called REVIEW was used. This package requires use of the DISPLA™ plotting software, but could be replaced by any (e.g. public domain) graphics package with some additional work.

The core of FICS uses the public domain software package called CLIPS (C language integrated production system), originally developed by NASA [2]. CLIPS is a computer language designed for implementing expert systems. The CLIPS shell is that portion of CLIPS which performs inferences, i.e. executes rules of the form “if A, then B.” One can think of CLIPS in its simplest form as consisting of facts and rules. Rules are executed when specified facts are asserted (“A” in the clause “if A, then B”). The consequences of a rule execution are to assert other facts (“B”) which can then execute other rules, and so on. A very important side effect of rule execution is the ability to activate functions

which extract and manipulate data and return the results of those manipulations to CLIPS. This information can then also be used to assert more facts to drive other rules, and so on.

The order of rule execution can be influenced by a priority assigned using a “salience” statement. Rules are fired (executed) according to their relative priority (salience) and according to when their data becomes available. The salience values can be used to provide a description of major rules in *approximate* sequential order provided the data arrives in approximately the order in which it is expected.

As an example, many of the rules in FICS represent system tests to be executed and many of the facts represent information regarding the existence and quality of data needed to conduct those tests. Existence of necessary data causes a rule to fire which then conducts the system test.

Use of the CLIPS production language provides two powerful capabilities not provided by conventional languages such as C or FORTRAN: (1) chaining, i.e. the ability to emulate human reasoning in software, and (2) data driven execution. The chaining capability was used only sparingly in the initial implementation. The data driven execution capability has been invaluable however, since FICS begins execution as soon as the shot completes. It would not be useful if it waited until all data was available – a process that often takes several minutes. Thus the code doesn’t know apriori when or even *if* the data needed to check for faults will be available. Since CLIPS is data driven, each rule executes if and when the data shows up. If a particular piece of needed data is not available, those tests which require that data don’t run and an alarm is raised saying the data was not acquired.

## V. SUMMARY

The FICS automated fault detection software system has been developed and was used routinely in DIII–D operations during 1999. It’s purpose is to improve the reliability of DIII–D experiments by providing early detection of a number of both common and uncommon system problems. It has also allowed the routine evaluation of *many* more systems between experimental discharges than could practically be done before. It has already proven its value by frequently detecting system problems which had not been noticed by tokamak operators.

## REFERENCES

- [1] J.K. Ousterhout, Tcl and the Tk Toolkit, Addison-Wesley, 1994.
- [2] J.C.Giarratano, G.Riley, Expert Systems Principles and Programming, PWS Pub. Co.