

ANL/RA/CP--80010
CONF-940402--13

To be presented at the International Topical Meeting on Advanced Reactors Safety, April 17-21, 1994, Pittsburgh, Pennsylvania and published in its Proceedings by the American Nuclear Society.

Full Paper for Log Number 57

Development of a Graphical User Interface Allowing Use of the
SASSYS LMR Systems Analysis Code as an EBR-II Interactive Simulator*

P. L. Garner¹, J. D. Staffon², L. L. Briggs¹, K. C. Gross¹, and J.-Y. Ku¹

¹Reactor Analysis Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439-4842

²Integral Fast Reactor Operations Division
Argonne National Laboratory
P. O. Box 2528
Idaho Falls, Idaho 83403-2528

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.

The submitted manuscript has been authored by a contractor of the U. S. Government under contract No. W-31-109-ENG-38. Accordingly, the U. S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes.

*Work supported by the U. S. Department of Energy, Nuclear Energy Programs, under Contract W-31-109-ENG-38.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DEVELOPMENT OF A GRAPHICAL USER INTERFACE ALLOWING USE OF THE SASSYS LMR SYSTEMS ANALYSIS CODE AS AN EBR-II INTERACTIVE SIMULATOR

P. L. Garner¹, J. D. Staffon², L. L. Briggs¹, K. C. Gross¹, and J.-Y. Ku¹

¹Reactor Analysis Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439-4842 USA

²Integral Fast Reactor Operations Division
Argonne National Laboratory
P. O. Box 2528
Idaho Falls, Idaho 83403-2528 USA

ABSTRACT

The SASSYS computer program for safety analyses of liquid-metal-cooled fast reactors has been adapted for use as the simulation engine under the graphical user interface provided by the GRAFUN and HIST programs and the DataViews software package under the X Window System on UNIX-based computer workstations to provide a high fidelity, real-time, interactive simulator of the Experimental Breeder Reactor Number II (EBR-II) plant. In addition to providing analysts with an interactive way of performing safety case studies, the simulator can be used to investigate new control room technologies and to supplement current operator training.

I. INTRODUCTION

Over the past several years, the number of research projects and plant upgrade efforts focused on the Experimental Breeder Reactor Number II (EBR-II) facility has grown steadily. These include

- modern control developments,
- whole-system diagnostic software development and validation,
- control room hardware upgrades,
- plant protective system upgrades,
- sensor validation software development and testing,
- expert system development,
- formulation and testing of new procedures,
- human factors studies of man/machine interfaces, and
- development and safety analysis of transient materials tests.

Since using EBR-II on demand as a test bed for all these projects places an operational burden on the plant, interest has grown in developing an EBR-II interactive simulator to serve as an alternative test bed. An EBR-II simulator would also be useful to

- supplement current operator training,

particularly in the area of response to severe off-normal transients, and

- provide an improved means of conducting safety analysis calculations, allowing interactive input preparation and output visualization.

The development of such a simulator for EBR-II has been undertaken and is described in this paper; although this effort focussed on a specific reactor (EBR-II), the techniques developed are adaptable to other reactors.

To fully satisfy the needs of these diverse projects, the simulator cannot provide a limited-range, low-fidelity simulation, as has been typical of nuclear power plant simulators which are used only for training purposes. Instead, a high-fidelity, real-time simulator able to replicate full accident phenomena is required. The key components of such a simulator are (1) a high-fidelity simulation engine, (2) graphics software to create views to display plant parameters and allow real-time interactive user intervention in the simulation process, and (3) a program to manage the interaction of the user with the screen and the two-way interaction between the screen and the simulation engine. To create a simulator, then, requires finding or generating programs to serve as these components and interfacing them to create a working simulator. Fortunately, the basic requirements of all three components can be found in computer programs which are already in use at Argonne National Laboratory (ANL), as described in the next two paragraphs.

Some time ago, the capability for viewing data from the EBR-II data acquisition system (DAS) was developed. The data collected by the DAS are sent across an electronic link from ANL-Idaho to ANL-Illinois and processed by an interface program, which allows the data to be viewed in real time on the screen of a computer terminal connected to the Reactor Analysis Division Computing Network (RADCN). (The DAS data are also viewable by the EBR-II operators and others at ANL-Idaho without the use of the data link between the two

sites.) The capability to store the DAS data in a set of archive files, which can be reviewed at a later time, was also developed. These efforts provided the second and third components needed for the simulator: the basic graphics software (second component) is the commercially available DataViews software package^{1,2,3} and the display management program (third component) is the GRAFUN computer program, which was developed at ANL-Idaho as the interface between the DAS data and the computer terminal screen.

As a separate effort, the SASSYS liquid-metal reactor (LMR) systems analysis computer program^{4,5} has been under development at ANL-Illinois for many years and has been applied to, among other work, EBR-II test and safety analyses; this computer program has been adapted for use as the high-fidelity simulation engine (first component), thus completing the set of three components.

To produce a functioning simulator, these three basic pieces had to be modified and interfaced to allow them to work together as an interactive simulator. The two major tasks were (1) extending the GRAFUN computer program to incorporate management of a simulation engine, allowing for interactions between it, the user, and the screen, and (2) adapting the SASSYS computer program for interactive execution, rather than the batch execution mode used heretofore for test and safety analyses. The result is an EBR-II simulator which can serve as a unique operator training tool, an offline test bed for the variety of projects described above, and an efficient means for analysts to interactively perform calculations for safety case studies. The collection of computer programs which are used for the simulator have been named SASSIM/EBR-II, standing for SASSYS-driven simulation of the EBR-II plant. The simulator can be invoked from any computer terminal which is connected to a UNIX-based workstation on the RADCN and supports the X Window System.

The remainder of this paper provides descriptions of (1) the pieces of software which comprise the simulator, (2) how these pieces are combined to produce the desired EBR-II simulator, and (3) how the user interacts with the simulator. The focus of this paper is on use of this software as a real-time EBR-II simulator; however, usage of just the graphics and display management software to view EBR-II real-time or archive DAS data is also described.

II. DISPLAY COMPUTER PROGRAMS

The EBR-II simulation and visualization project involves a number of computer programs and data sources.

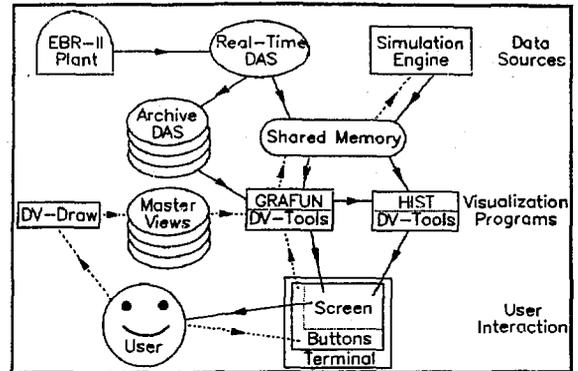


Figure 1. Relationships Between Data Sources, Visualization Programs, and Users

The principal relationships among these items and the user are illustrated in Figure 1; solids lines are used to show data transfer to the user, dashed lines are used to show data transfer from the user, and arrows emphasize the data transfer direction for the two line types. The relationships shown in Figure 1 will now be described.

The DV-Draw program¹ of the DataViews software package is used to initially create the views to be displayed on the workstation screen. This program has the same capabilities provided by many other workstation-based drawing programs, allowing the user to draw basic shapes (e.g. line, circle, polygon) and text using a combination of mouse and keyboard actions. A special feature of DV-Draw is the ability for the initially static drawing to change dynamically; this is accomplished by creating a link between an object on the screen and a data source. Two examples of this dynamic feature are as follows. (1) The corners of a polygon are identified by DataViews programs as "control points"; the x and y positions of a control point may be linked to a data source; as the values provided by the data source change, the control points move and, thus, the polygon's shape and location change. (2) A text string may contain both fixed and variable sections, the latter containing a format code; by linking this text string to a data source, the variable section of the text string changes as the value of the data source changes. An additional feature of DV-Draw is the ability to receive input from the user (e.g. selecting an object on the screen using a mouse). To take full advantage of these dynamic capabilities, the companion DV-Tools package must be used. The DV-Draw program was used to create the 22 main drawing files which may be displayed on the screen and the approximately 120 other drawing files representing input and subdrawing areas on the screen; the use of DV-Draw to create the master view files is shown in the left center portion of

Figure 1. Further descriptions of these drawing files are provided in Section V of this paper.

DV-Tools^{2,3} is a library of routines in the DataViews software package which are callable by a user program to interact with the screens in some dynamic sense. It is through calls to these routines that the user brings a specific view to the screen, forms a connection between screen objects and data sources, and processes inputs from the user. The user program must be written in the C programming language in order to call DV-Tools routines.

GRAFUN is the program which was originally written to manage the interaction between the user, the screen, and the data source when viewing real-time EBR-II DAS data. It contains calls to numerous DV-Tools routines to accomplish its task. The GRAFUN program establishes the default view to be shown on the screen, establishes the connection between dynamic objects on the screen and the data source area, and changes the particular view to be shown on the screen based on user selection. The data source is a shared memory segment, which is discussed in Section IV of this paper; the coding in GRAFUN establishes a connection between approximately 120 of the data items in the shared memory segment and objects on the screen in order to show their values. (As the development process continues, more of the data items in shared memory will be connected to screen objects.) The various views and the way in which the user selects a specific one of these views to be shown on the screen are discussed in Section V of this paper. The GRAFUN program also manages the HIST program and the simulation engine.

The HIST program, shown in the right center of Figure 1, is responsible for maintaining the user-selectable view which contains a set of strip charts which record the changes in six specific key plant parameters over time. Like the GRAFUN program, the HIST program contains calls to numerous DV-Tools routines to accomplish its task. The HIST program is started by, and runs simultaneously with, the GRAFUN program. It maintains this view in a lowered window on the screen, so that it is always an active view but not always visible. When the user wishes to see this view, it is raised to the top of the visibility list by GRAFUN; similarly, it is lowered to the bottom of the visibility list by GRAFUN when the user selects another view. This type of management is used in order for the strip charts to maintain their history when they are not the primary view without incurring excessive processing delays in the GRAFUN program.

The GRAFUN and HIST programs are both written in the C programming language. They must be executed

on a computer workstation under a UNIX-based operating system and the X Window System display environment. These programs have been tested on Sun Microsystems SPARC and IBM RISC System/6000 workstations with the user display being on a Sun Microsystems console or a Network Computing Devices multi-window terminal. The views produced by the GRAFUN and HIST programs are more readable if the display terminal supports color rather than just gray-scale monochrome.

At later points in time, the GRAFUN program was modified to be able to (1) play back a previously saved archive file of EBR-II DAS data, shown on the left center of Figure 1, and (2) exchange data with a reactor simulation computer program or "simulation engine" (i.e. SASSYS), shown in the upper right of Figure 1. This latter change provided a mechanism for the user to alter certain boundary conditions using the screen window managed by the GRAFUN program while the simulation engine was running -- a bidirectional data flow between this program and GRAFUN.

There are several additional important points illustrated in Figure 1. There are three operating modes for this software: (1) viewing real-time DAS data, (2) reviewing archived DAS data, and (3) interaction with a simulation engine. Regardless of the operating mode,

- the user sees the same types of views on the screen and
- these views are managed by the same computer programs (GRAFUN and HIST).

There is, therefore, a uniform look and feel from the user's perspective for all operating modes and there is only a single version of each of the visualization programs, which decreases the computer program maintenance effort.

The routine user of this assemblage of programs is not expected to create new views using DV-Draw or to modify the coding in the GRAFUN, HIST, and SASSYS programs.

III. SIMULATION ENGINE: SASSYS

Since the SASSYS (version 3.0) computer program^{4,5} is a highly developed LMR simulation tool which has been used extensively for EBR-II safety analyses, it is a logical choice to be installed as the EBR-II simulation engine. SASSYS provides detailed modeling of the reactor core by dividing the core into channels, which provide either single-pin or multiple-pin⁶ representations of one or more subassemblies. The core modeling includes point kinetics and reactivity feedback. In addition, SASSYS simulates components in the primary

and secondary coolant loops and in the balance of plant.^{7,8,9} A control system model¹⁰ based on classical control theory is also available in SASSYS. The code can handle arbitrary arrangements of core, loop, and control system components; therefore, its use is not limited to just the EBR-II design; rather, it is applicable to a wide range of LMR designs and types of transients. The SASSYS code has been applied to cases ranging from anticipated operational transients through severe accident analyses in which coolant boiling and fuel and cladding melting occur; until recently, all analyses were performed in batch mode.

Several features of SASSYS make it an attractive simulation engine for an EBR-II simulator. First, the flexibility of the code allows the user to incorporate whatever degree of complexity is desired in the simulation, from very simple loops and core models which run quickly to very detailed modeling at the expense of longer running times. Second, the high fidelity of the models available in SASSYS make it possible to simulate a wide range of transients. Third, a significant amount of work¹¹ has already gone into validating the SASSYS EBR-II model against data from EBR-II. Fourth, since the same SASSYS algorithms are used for the simulation engine as are used for EBR-II safety analyses, the modeling of EBR-II is consistent for all applications, and the simulator can provide a meaningful demonstration of a particular safety analysis. Finally, the fact that SASSYS has a control system model makes it straightforward to allow the user to interactively adjust limiting safety system settings and configure the plant protective system in much the same way as would be done in the actual plant.

The main change needed to allow the use of SASSYS as the simulation engine was to allow for a two-way exchange of data with the user during execution -- i.e. interactive, rather than the previously used batch, execution mode. During execution, the user is allowed to alter certain boundary conditions, similar to those that the EBR-II plant operator would adjust, using input objects on the terminal screen; these include the reactivity, speed or head of the various pumps, and control system setpoints. Many of the calculated plant conditions are displayed on the terminal screen as the transient proceeds; these include reactor power and temperatures, pressures, and flow rates throughout the plant. The two-way flow of data between the terminal screen and the SASSYS computer program is managed by the GRAFUN program and the values are transferred through the shared memory segment.

IV. USE OF SHARED MEMORY

The GRAFUN computer program associates a specific memory location with each item to be dynamically

displayed on, or input from, the terminal screen; as the value stored in a specific memory location changes, the value displayed on the screen will change; similarly, as the user changes an input value on the screen, the contents of the specific memory location where the value is stored will change. These data values are made available to the GRAFUN, HIST, and SASSYS computer programs by being stored in a special type of memory referred to as a "shared memory segment". The rest of this section defines the concept of a shared memory segment and the specific structure of the shared memory segment used by the EBR-II simulator programs.

Normally, an executing computer program has its own segment of the computer's memory in which the program's instructions and data are stored; the operating system manages memory allocation so that one program does not access or alter the memory segment associated with any other program. Some operating systems (e.g. SunOS and AIX) provide the capability for the user to request that a segment of memory be available to more than one program simultaneously, bypassing the normal separation between programs; such a segment is a "shared memory segment". Each shared memory segment is identified by a "key" having a unique value; each program wanting access to the same shared memory segment must do so by specifying the same value for this key. When one program changes the contents of any portion of the shared memory segment, the change is immediately available to all other programs having access to the same shared memory segment (i.e. there is no need for the user to perform an explicit read or write operation). Shared memory segments have read and write access control permissions analogous to those associated with files.

As an example, consider the use of the GRAFUN and HIST programs with the SASSYS program as the simulation engine and refer to Figure 1. The data values to be displayed are computed by SASSYS and placed in the shared memory segment at the end of every main time step. The GRAFUN and HIST programs access these data by executing the commands needed to reference the same shared memory segment being used by the SASSYS program and display the values on the screen. Similarly, any values input via the screen by the user are placed by GRAFUN into the shared memory segment; the SASSYS program will process these changed values before taking the next main time step. A similar approach is used if GRAFUN and HIST are to display data from the real-time EBR-II DAS or an archive file of DAS data; in the former case, the GRAFUN and HIST programs reference a special shared memory segment which contains the real-time EBR-II DAS data, which is updated every second with new values sent across an electronic link to the

plant's DAS; in the latter case, the GRAFUN program reads a file and maps its data into the shared memory segment; the SASSYS simulation engine is not used for either of these two cases. The assignment of a proper value for the key is handled by the script which the user invokes to start this set of programs.

The arrangement of data in the shared memory segment is shown in Table I. The shared memory segment is 1032 words (32 bits/word) long. The first four words are a time stamp; the time stamp consists of the wall time (mountain zone) and date when the segment contains real-time or archive-file DAS data; the time stamp consists of the current time step number and the case time when the segment contains data from a simulation engine (denoted SimEng in Table I); the time stamp is shown on almost every view which GRAFUN displays. The next two words and the final three words are not currently used. Words 7 through 1029 are the actual data values; the ordering is the same as that used by the EBR-II DAS, where each word stores the value of one DAS channel (denoted DAS#*nnn* in the table); the values for approximately 120 of these 1023 DAS channels are displayed by the GRAFUN and HIST programs; more values may be displayed in the future.

Table I. Shared Memory Data Arrangement

Word	Description
1-2	DAS: wall time (hh:mm:ss) SimEng: time step number (8-digit integer as characters)
3-4	DAS: date (MM/DD/YY) SimEng: case time (hh:mm:ss)
5-6	Not currently used.
7	DAS#001 (floating point)
...	...
1029	DAS#1023 (floating point)
1030-1032	Not currently used.

V. VIEWS AND OPERATION

The GRAFUN program presents its view for output and input as a "window" under the X Window System. Figure 2 is the default view shown when GRAFUN initializes. (Some features of these color screen views are not distinct in the grey-scale figures used in this paper.) The various areas of this window will now be discussed.

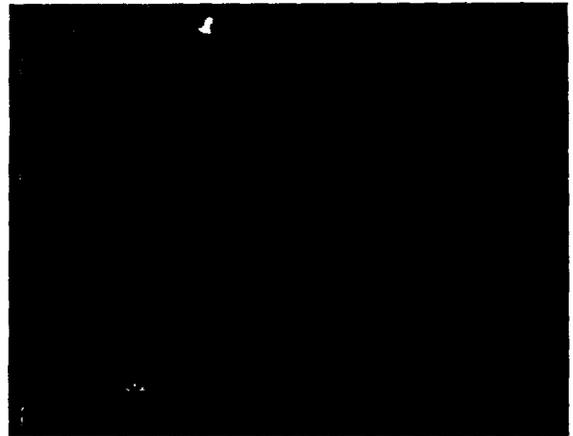


Figure 2. Default GRAFUN Display Window (Plant View)

The majority of the window is occupied by a user-selectable view. The default view in Figure 2 provides an overview of many plant conditions; temperatures in the primary sodium (upper left), secondary sodium (upper right), and steam (lower right) systems are shown on an analogue basis by the dynamic locations of polygon corner points relative to the scale on the left; reactor power is shown digitally and its direction of change (i.e. decreasing, steady, or increasing) is indicated by a text box; relative power and primary sodium flow rate are shown using digital and analogue barometer displays in the lower left; the steam drum and condenser hot well pressures and water levels are shown in the lower right; the upper right corner shows the status (currently ONLINE but could also read PAUSED, NOT UPDATING, READING, or COMPLETE) and the time stamp (indicating time step number 10 at 5 s into a transient simulation).

Several of the other user selectable views for this main window area are as follows. The primary sodium system is shown in Figure 3; the major components are shown along with digital displays of temperatures, pressures, flow rates, and power. The secondary sodium system is shown in Figure 4; inlet and outlet temperatures for the ten heat exchanger units are shown on an analogue basis by the size of the shaded rectangles relative to the scale on the left; the flow rate is shown digitally. The steam system is shown in Figure 5; the major components are shown along with digital displays of temperatures (T), pressures (P), qualities (x), water levels (L), and flow rates (G); dotted lines denote components not currently included (although they could be) in the SASSYS model of EBR-II.



Figure 3. Primary System View

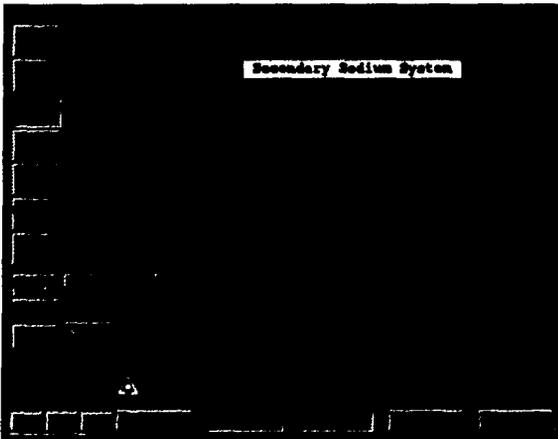


Figure 4. Secondary System View

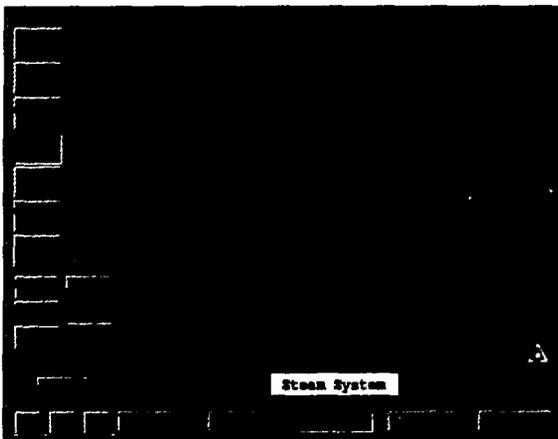


Figure 5. Steam System View

Above this main window area is a title area which indicates the mode of operation: interaction with a

simulation engine (which is the mode for Figures 2 through 5), display real-time DAS, or display archive-file DAS.

Several areas of the window contain buttons or other objects which may be selected by the user to (1) change the view shown on the screen or (2) adjust one of the boundary conditions if using a simulation engine. Selection of a particular item or value is accomplished by using the mouse to position the screen cursor on the desired object and clicking the left mouse button; the appearance of the object changes after it has been selected.

The user can change the view in the main window area at any time by selecting a button from the two columns in the upper left part of the window; the left column contains the main view selection buttons, which are always displayed; the right column of buttons depends on which button has been selected in the left column (cf. differences among Figures 2 through 5); these buttons give the user a choice of 21 different views as described in Table II. (The views selected by buttons 11, 12, and 13 are identical to those selected by buttons 2, 3, and 4, respectively. The group of five buttons below these two columns are not fully functional; in the future, they will be used to change operating modes.) The five buttons below the main window area may be selected to (left to right) force a redraw of the screen, superpose dashed grid lines on the main view area which align with the temperature scale (cf. Figure 4), pause from displaying new data on the screen, restart the simulation engine computer program, and terminate this display session.

There are two areas of the window which allow the user to alter boundary conditions in the simulation engine at any time during its execution. (1) The lower left corner of the window contains a scrollbar and three buttons. A new value for the indicated quantity may be chosen using the point-and-click selection process within the scrollbar outline, on the ◀ symbol (decrease), or on the ▶ symbol (increase). Selecting a button at the bottom of this area determines which quantity's scrollbar is shown in this area; scrollbars are available for the adjustment of reactivity (cf. Figure 2), speed of primary pumps M1 (cf. Figure 3) and M2, and head of the secondary pump (cf. Figure 4). (2) Selecting button 7 from the upper left column fills the main window area with another group of input controls as shown in Figure 6. The top row of scrollbars, menus, and buttons allows the user to alter the functionality statuses and setpoints for control system trips associated with (left to right) power, period, primary sodium flow rate, and subassembly outlet temperatures. The menus in the left portion of the bottom part may be used to manually trip one or both of the primary sodium

Table II. View Selection Buttons

Left Column	Right Column
1. Plant: overall conditions	11. Primary sodium system
	12. Secondary sodium system
	13. Steam system
	14. Key parameters on kiviart chart
	15. Key parameters on dials
	16. Key parameters on strip charts
2. Primary sodium system	21. Pump M1
	22. Pump M2
	23. Pumps M1 and M2
	24. Subassembly outlet temperatures
	25. Neutron flux by subassembly
	26. Neutron flux as 3D plot
	27. Neutron flux as contour plot
3. Secondary sodium system	
4. Steam system	
5. Fuel handling system	
6. Radiation levels: overall plant	
	61. Operating floor
	62. Mezzanine floor
	63. Basement floor
	64. Subbasement floor
7. Input for simulation engine	

system pumps. The scrollbar on the right side of the bottom allows the user to set a reactivity insertion rate, simulating a function provided by the Automatic Control Rod Drive System in the plant, and the button below this will simulate a manual scram. Buttons are provided in the lower right corner to separately kill and (re)start the simulation engine computer program.

VI. CONCLUDING REMARKS

An EBR-II simulator has been created by adapting existing programs (GRAFUN and HIST) for data visualization with an existing program (SASSYS) for safety analyses. The resulting EBR-II simulator can run in real time with the latest SASSYS version (Release 3.0) on current generation Sun Microsystems (e.g. SPARC2) and IBM (e.g. RS6000/350) workstations. (The actual

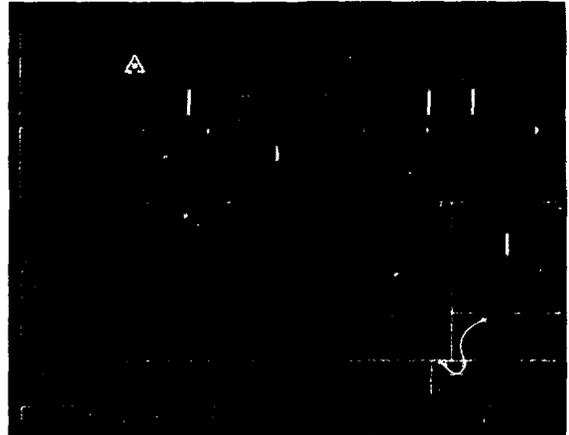


Figure 6. Simulation Engine Control View

execution speed varies with the complexity of the EBR-II model in SASSYS and the specific workstation, and its other load, upon which the simulation is run.)

The SASSYS EBR-II simulator provides a high-fidelity, real-time simulation of the EBR-II plant which can be used as a demonstration tool in conjunction with EBR-II safety analyses of plant modifications and in-core transient tests, as an operator training tool, and as a testing ground for new component models and control room upgrades. Future work on the simulator will include expanding the amount of information displayed in the simulator views, adding views which more closely replicate current control room displays, and modifying views to reflect upgrades in the EBR-II control room.

Although specifically developed for the EBR-II plant, the simulator is adaptable to other LMR plants for which SASSYS is an appropriate simulation engine; this would entail creating new versions of views which show plant-specific component arrangements. Similarly, the GRAFUN and HIST data visualization programs are adaptable for use with other simulation engines, such as the DSNP-based simulation¹² of EBR-II routinely used at ANL-Idaho. The overall approach could also be adapted for non-LMR type designs.

ACKNOWLEDGMENTS

This work was supported by the U. S. Department of Energy, Nuclear Energy Programs, under Contract W-31-109-ENG-38.

AIX and RISC System/6000 are trademarks and IBM is a registered trademark of International Business

Machines Corporation. The X Window System is a trademark of Massachusetts Institute of Technology. Network Computing Devices is a trademark of Network Computing Devices, Inc. SPARC is a registered trademark of SPARC International, Inc. Sun and SunOS are trademarks and Sun Microsystems is a registered trademark of Sun Microsystems, Inc. UNIX is a registered trademark of UNIX System Laboratories, Inc. DataViews, DV-Draw, and DV-Tools are registered trademarks of V.I. Corporation.

REFERENCES

1. "DV-Draw User's Guide (Release 9.0)", V.I. Corporation, Northampton, Massachusetts (1991).
2. "DV-Tools User's Guide (Release 9.0)", V.I. Corporation, Northampton, Massachusetts (1991).
3. "DV-Tools Reference Manual (Release 9.0)", V.I. Corporation, Northampton, Massachusetts (1991).
4. J. E. Cahalan and T. Y. C. Wei, "Modeling Developments for the SAS4A and SASSYS Computer Codes," *Proceedings of the 1990 International Fast Reactor Safety Meeting* (Snowbird, Utah, August 12-16, 1990), American Nuclear Society, La Grange Park, Illinois, II:123-132 (1990).
5. J. E. Cahalan, "Advanced LMR Safety Analysis Capabilities in the SASSYS and SAS4A Computer Codes," *Proceedings of the International Topical Meeting on Advanced Reactors Safety* (Pittsburgh, Pennsylvania, April 17-21, 1994), American Nuclear Society, La Grange Park, Illinois (1994).
6. F. E. Dunn, "Verification of the Multiple-Pin Treatment in the SASSYS-1 LMR Systems Analysis Code," *Proceedings of the International Topical Meeting on Advanced Reactors Safety* (Pittsburgh, Pennsylvania, April 17-21, 1994), American Nuclear Society, La Grange Park, Illinois (1994).
7. P. A. Pizzica, "An Improved Steam Generator Model for the SASSYS Code," *Trans. Am. Nucl. Soc.*, 60, 712-713 (1989).
8. Loral L. Briggs, "A New Balance-of-Plant Model for the SASSYS-1 LMR Systems Analysis Code," *Trans. Am. Nucl. Soc.*, 60, 709-710 (1989).
9. J.-Y. Ku, "SASSYS-1 Balance-of-Plant Component Models for an Integrated Plant Response," *Trans. Am. Nucl. Soc.*, 60, 716-717 (1989).
10. R. B. Vilim, T. Y. C. Wei, and F. E. Dunn, "Generalized Control System Modeling for Liquid-Metal Reactors," *Nucl. Sci. Eng.*, 99, 183-196 (1988).
11. J. P. Herzog, "SASSYS Validation with the EBR-II Shutdown Heat Removal Tests," *Trans. Am. Nucl. Soc.*, 60, 730-731 (1989).
12. J. T. Madell, E. M. Dean, D. Mohr, and T. C. Hung, "Dynamic Analysis of the EBR-II Plant with the DSNP-ND Simulation Package", *Proceedings of the International Topical Meeting on the Safety, Status, and Future of Non-commercial Reactors and Irradiation Facilities* (Boise, Idaho, September 31 - October 4, 1990), American Nuclear Society, La Grange Park, Illinois, II:705-716 (1990).