

1992

Knowledge base expert system control of spatial xenon oscillations in pressurized water reactors

Serhat Alten

Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Artificial Intelligence and Robotics Commons](#), and the [Nuclear Engineering Commons](#)

Recommended Citation

Alten, Serhat, "Knowledge base expert system control of spatial xenon oscillations in pressurized water reactors " (1992). *Retrospective Theses and Dissertations*. 10365.

<https://lib.dr.iastate.edu/rtd/10365>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

6

93

01993

U·M·I

MICROFILMED 1992

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600



Order Number 9801993

**Knowledge base expert system control of spatial xenon
oscillations in pressurized water reactors**

Alten, Serhat, Ph.D.

Iowa State University, 1992

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

**Knowledge base expert system control of
spatial xenon oscillations in pressurized water reactors**

by

Serhat Alten

**A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY**

**Department: Mechanical Engineering
Major: Nuclear Engineering**

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Members of the Committee:

Signature was redacted for privacy.

**Iowa State University
Ames, Iowa
1992**

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ix
CHAPTER 1. INTRODUCTION	1
Characteristics of Xenon Fission Product Poisoning	1
Spatial Xenon Oscillations	3
Expert Systems	6
Scope of This Study	11
CHAPTER 2. CONTROL OF SPATIAL XENON OSCILLATIONS	15
Early Control Strategies	15
Advanced Control Studies on Xenon Oscillations	20
Optimal control approach	21
Adaptive control approach	23
Heuristic Control Studies	24
CHAPTER 3. A PRODUCTION SYSTEM LANGUAGE: OPS5 .	28
Data Representation in OPS5	29
Production Memory in OPS5	31
The OPS5 Inference Engine	32
Organization and Control in OPS5	36

CHAPTER 4. DEVELOPING THE EXPERT SYSTEM ACES . . .	37
Development of Rules for ACES	37
Working Memory of ACES	44
Production Memory of ACES	44
Support Routines Used in ACES	48
Flow Control in ACES	49
CHAPTER 5. SYSTEM MODELING	51
Developing the Model Equations	51
Solution of the System	60
Steady state solution	61
Time-dependent solution	64
Generation of Constants Used in ACES	67
CHAPTER 6. VERIFICATION OF ACES	69
The Results of the Reactor Model	69
Verification of ACES	72
CHAPTER 7. CONCLUSIONS AND RECOMMENDATIONS . . .	107
Conclusions	107
Recommendations for Future Work	110
BIBLIOGRAPHY	111
APPENDIX A: SOURCE CODE OF ACES	116
APPENDIX B: SUPPORT ROUTINES OF ACES	133
APPENDIX C: SUPPORT ROUTINES OF REACTOR MODEL . . .	139
APPENDIX D: STEADY.FOR	151

APPENDIX E: REACTIVITY WORTH ROUTINES 157

LIST OF TABLES

Table 3.1:	A sample conflict set in OPS5	34
Table 4.1:	List of reactor parameters used in ACES	39
Table 4.2:	Modification of boron control amount in ACES	47
Table 6.1:	Reactor constants	71
Table 6.2:	Total reactivity worths of FLCR banks	72
Table 6.3:	Test cases for verification of ACES	104
Table 6.4:	CPU time for initialization of ACES	105
Table 6.5:	CPU time for control cycle of ACES at steady state power .	105
Table 6.6:	CPU time for control cycle of ACES during a power transient	106

LIST OF FIGURES

Figure 2.1:	Bang-Bang control	17
Figure 2.2:	Direct offset control for AO oscillations	17
Figure 2.3:	First overtone control for AO oscillations	18
Figure 2.4:	SRL indirect control for AO oscillations	19
Figure 3.1:	Architecture of production system language OPS5	29
Figure 4.1:	Production memory of ACES	46
Figure 5.1:	Typical control rod layout of a PWR core	58
Figure 5.2:	Iteration Flowchart for Power Method	63
Figure 6.1:	Full power steady state flux distribution	70
Figure 6.2:	Case A: Reactor power and load demand	75
Figure 6.3:	Case A: Axial offset change	75
Figure 6.4:	Case A: Full length control rod positions-1	76
Figure 6.5:	Case A: Full length control rod positions-2	76
Figure 6.6:	Case A: Part length control rod position	77
Figure 6.7:	Case A: Boron concentration change	77
Figure 6.8:	Case A: The xenon distribution change-1	79
Figure 6.9:	Case A: The xenon distribution change-2	80

Figure 6.10: Case A: The flux distribution change-1	81
Figure 6.11: Case A: The flux distribution change-2	82
Figure 6.12: Case B: Reactor power and load demand	84
Figure 6.13: Case B: Axial offset change	84
Figure 6.14: Case B: Full length control rod positions-1	85
Figure 6.15: Case B: Full length control rod positions-2	86
Figure 6.16: Case B: Part length control rod position	87
Figure 6.17: Case B: Boron concentration change	87
Figure 6.18: Case C: Reactor power and load demand	88
Figure 6.19: Case C: Axial offset change	89
Figure 6.20: Case C: Full length control rod positions-1	90
Figure 6.21: Case C: Full length control rod positions-2	90
Figure 6.22: Case C: Part length control rod position	91
Figure 6.23: Case C: Boron concentration change	92
Figure 6.24: Case D: Reactor power and load demand	94
Figure 6.25: Case D: Axial offset change	94
Figure 6.26: Case D: Full length control rod positions-1	95
Figure 6.27: Case D: Full length control rod positions-2	95
Figure 6.28: Case D: Part length control rod position	96
Figure 6.29: Case D: Boron concentration change	97
Figure 6.30: Case E: Reactor power and load demand	98
Figure 6.31: Case E: Axial offset change	98
Figure 6.32: Case E: Full length control rod positions-1	99
Figure 6.33: Case E: Full length control rod positions-2	100

Figure 6.34: Case E: Part length control rod position	101
Figure 6.35: Case E: Boron concentration change	101

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my major professor, Dr. Richard A. Danofsky, for his invaluable guidance, suggestions, help and support during the course of this study. I would like to thank to my committee members, Dr. A. F. Rohach, Dr. R. M. Nelson, Dr. S. K. Gadia, and Dr. E. B. Bartlett, for their valuable suggestions. I am indebted to Dr. Bernard I. Spinrad for supporting me in my first year in USA, and making my dream come true. I am also indebted to Dr. A. F. Rohach for kindly giving his personal computer to me for using at home. With his help, I was able to spend some time with my baby while finishing my dissertation.

I would like to extend my appreciation to my friends and fellow colleagues Ramin Mikaili, Niyazi Sökmen, Taher Al-Jundi, and Yudi Imardyoko for bearing with me when I was pacing the office, and thinking out loud. Their timely comments and friendship were invaluable. I also would like to thank my friend B. Sadık Bakır for his anticipation in endless discussions about this study.

Last but not least, I would like to thank my family for their support during my stay in USA. My special thanks goes to my wife, Gönenç Bilge for her patience, understanding, and support, and for putting her career on hold for my study.

I dedicate this dissertation to my father, M. Turgut Alten, who unfortunately didn't live to see either this document or his first granddaughter Barış Bilge.

CHAPTER 1. INTRODUCTION

The purpose of this dissertation is to develop and demonstrate the use of an Expert System (ES) to aid in the control of xenon induced spatial power oscillations in a Pressurized Water Reactor (PWR) power plant. This chapter presents the important aspects of the xenon poisoning problem, and briefly introduces concepts associated with ESs as they relate to this problem. The main xenon oscillation control techniques, including the heuristic methods, will be introduced in Chapter 2. Chapter 3 will build a background on OPS5, the computer language used in the ES design. The rules of the expert system for the spatial xenon oscillation control will be developed and presented in Chapter 4. To verify the expert system, a pressurized water reactor is modeled in Chapter 5, considering the design characteristics of the ES. The results of the verification process are introduced and discussed in Chapter 6, and conclusions are summarized in Chapter 7.

Characteristics of Xenon Fission Product Poisoning

Having an exceptionally large absorption cross-section for thermal neutrons, xenon-135 (Xe-135) is the most important fission product in reactor control. The decay chain of fission products that leads to the production of Xe-135 starts with tellurium-135 (Te-135). Te-135 decays to iodine-135 (I-135) with a half-life of 11 sec-

onds. Since the half-life of Te-135 is very short, it is a common approach to assume that the fission yield is the only production term for I-135 with an equivalent value given by the sum of the I-135 and Te-135 yields. Also the absorption cross-section of I-135 is small enough to assume that decay is the only depletion term in the I-135 rate equation. I-135 decays to Xe-135 with a half-life of 6.7 hours, which is the main contributor to the production terms in the xenon balance equation. Xe-135 is also produced as a direct fission product in the core, but with a small yield. Xe-135 disappears both as a result of neutron absorption and its β^- decay to cesium-135 with a half-life of 9.2 hours.

Xenon and iodine are practically non-existent in a clean core. Their concentrations increase as the fission process continues. Xe-135 introduces a significant amount of negative reactivity to the core during this build-up. In a PWR, boron dilution from the system compensates for this negative reactivity increase during the early hours following the startup of a clean core [1, 2]. Both iodine and xenon eventually reach a flux dependent steady state concentration. The steady state iodine concentration is linearly proportional to the flux level of the core, while the steady state xenon concentration has a non-linear proportionality.

The flux dependency of the Xe-135 concentration initiates a delayed xenon transient after each power level change. As the reactor power decreases, for example, Xe-135 builds up in the core adding more negative reactivity. Even though the Xe-135 concentration decreases later and reaches a new steady state level, which is lower than the previous value, it creates a continuous reactivity change that has to be controlled. This xenon transient may become important especially after a reactor shutdown.

Following a reactor shutdown, being depleted only by the relatively slow decay process, Xe-135 builds up and introduces a strong negative reactivity into the core. The Xe-135 concentration then begins to decrease as the production due to I-135 decay decreases in time. This negative reactivity accumulation peaks about ten hours after shutdown [2]. The magnitude of the accumulation is flux dependent and may exceed the *Maximum Reserve Reactivity* of the reactor. Consequently, a start up after shutdown will be delayed while waiting for the negative reactivity to drop to the *Maximum Reserve Reactivity* limit. Such a delay may be on the order of hours which means losing a considerable amount of energy production, and therefore money in commercial reactors. The time necessary to wait for a new start-up due to xenon buildup is called the *reactor deadtime* [2].

For high flux reactors, the neutron absorption becomes a very strong component of the Xe-135 dynamics. If the flux level is high enough, even local flux variations will induce a transient in the xenon concentration. As it was mentioned before, each power level change, regardless of the reason, initiates a Xe-135 transient. The Xe-135 concentration will continue to be the driving force of a flux transient during the steady state power operation long after a power transient. The spatial power distribution in the reactor core will be disturbed by spatial changes in the Xe-135 concentration which can in turn lead to an oscillation in the power distribution of the core while the total power remains constant.

Spatial Xenon Oscillations

The spatial power oscillations in nuclear reactors were first discussed during the mid 1950s. These oscillations were attributed to Xe-135 in 1958 by Randall

and St. John [3]. They pointed to "the oscillations in power distribution that are associated with a periodic redistribution of xenon poison." The equations of the oscillations were derived from one-group diffusion theory, and the thresholds for the oscillation were given in this study. Additionally, they proposed a solution to control such oscillations using independent control of local core regions.

A spatial xenon transient typically starts following a local change in the flux distribution. Suppose that the thermal flux is increased in one region while it is decreased in another. Xenon will burn out faster in the high flux region, and the xenon absorption will decrease introducing a local positive reactivity. As a result, the flux will increase further in this region. At the same time, the iodine concentration will increase as the flux increases. The xenon production by iodine decay will eventually compensate for the xenon burn out, and reverse the course of the transient. The xenon concentration will then begin to increase, introducing negative reactivity to the region this time. Therefore, flux will begin to decrease in the region, reducing the xenon depletion rate even further. The course of the transient will be reversed again when the xenon decay term dominates the system.

Similar events will take place in the other half of the core with a 180 degree phase shift. As a result, reactor power can remain constant during the transient, while the power distribution will experience a divergent oscillation. This oscillation may go unnoticed and reach dangerous local flux levels if only the total power of the core is monitored. To detect such an oscillation, power reactors are equipped with devices for independent monitoring of the power production in different regions of the core.

The key reactor parameters in terms of xenon oscillations are the dimensions

and the flux level of the core. In order to sustain a significant xenon oscillation, the average thermal flux of the reactor should be greater than $10^{13} \text{ n/cm}^2\text{s}$ [1, 4]. Together with the high flux, the reactor core should be large relative to the neutron mean free path such that the both halves of the core can be assumed to operate independently [5]. Almost all PWRs meet these conditions. Consequently, they are vulnerable to xenon oscillations.

Both the radial and axial dimensions of a commercial reactor are large enough to sustain xenon oscillations, however, the axial oscillations are the ones of major concern in most cases. By design, control rods are grouped into banks which can be moved together, and the absorber rods in a bank are always evenly distributed along the radial and azimuthal directions in a power reactor. The control rod banks are driven into the core from one side, usually the top of the reactor core. Therefore, a control rod motion will result in a symmetric reactivity insertion in the core in the radial and azimuthal directions, but the reactivity insertion will not be symmetric in the axial direction. This uneven reactivity insertion will eventually initiate an oscillation in the axial flux distribution because of the delayed xenon feedback. Consequently, nuclear reactors are more likely to experience axial xenon oscillations than radial oscillations.

The xenon oscillation is not one of the transients that may end with a catastrophe. It is slow, allowing enough time for a response if the operator correctly follows procedures to damp the oscillation and has adequate instrumentation to observe what is happening in the core. The proper control can be provided if the steady state flux distribution is kept constant as much as possible during the power transient, and restored to that distribution after the power transient.

Expert Systems

Computer technology has developed in the past several decades to the point that one of the favorite science fiction subjects, Artificial Intelligence (AI), has become a reality. The idea of developing *intelligent* machines is as old as computers, despite the fact that there is no unique definition of *intelligence*. According to a commonly used definition, AI is the computer science of designing systems which can simulate intelligent human behavior such as understanding, learning, reasoning, and problem solving [6]. The intelligent behavior attributed to humans in this definition is only a subdomain of the *intelligence* defined by Turing. Turing proposed a test of intelligence where a human and a machine will separately answer the same set of questions asked by human observers. If the observers were unable to tell which one was human, then the machine would be accepted as intelligent. Although none of the existing AI techniques will pass such a test, some game programs would approach this capability, provided that the questions are limited within a given domain [7].

AI includes a broad area of study. The research topics considered as building blocks of AI are searching (for solutions), language processing, machine learning, logical programming, and uncertainty handling (fuzzy-logic) [8]. These blocks add up to the final products of AI such as Expert Systems in reasoning and problem solving, and Neural Networks in pattern recognition. We are interested in one of the final products of AI, Expert Systems, in this study.

Jackson [6] defines an ES as *a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice*. Therefore, an ES must

- simulate human reasoning about a problem domain rather than the domain itself,
- perform reasoning over the representations of human knowledge in addition to numerical calculations, and data retrieval,
- use heuristic or approximate methods which are not guaranteed to succeed,
- deal with the problems that normally require a considerable amount of human expertise,
- perform its task with a reasonable speed and reliability, and
- be capable of explaining and justifying the solution.

To perform these duties, an ES consists of two modules, the knowledge base and the inference engine. The knowledge base stores the information about the subject domain given by an human expert. It is the knowledge engineer's responsibility to transfer and transform the potential problem-solving expertise from some knowledge source to an ES. Transferring the knowledge from an expert, which is currently an interactive process between the knowledge engineer and the expert, has always been an important problem in ES studies. Having different professional jargons, experts have difficulties explaining their experience in a deterministic way that can be programmed. Consequently, knowledge acquisition is quite inefficient as it is currently done, producing two to five units of knowledge per day [6, 9]. Automated knowledge elicitation techniques, and machine learning paradigms are important research subjects aimed at improving efficiency in this aspect of the AI field.

The knowledge engineer must also decide about the language to be used for knowledge representation to transform the expertise into a computer program. The representation language that is used must have

- logical adequacy to have the capability of making all distinctions that have to be made,
- heuristic power to use the representations for solving problems, and
- notational convenience to encode a substantial amount of data.

Among existing conventions of knowledge representation, production systems, structured objects and predicate logic are the most generally used ones [6]. Structured formalisms such as frames are easier to understand and handle, but do not allow for exceptional cases. More relaxed formalisms such as production systems, are obviously more error prone because of the handling and designing difficulties of the knowledge base. Since the production systems will be used in this study, detailed information about them will be given later.

Several computer languages were developed to implement these formalisms. The first AI language, called LISP because of its list processing characteristic, has been developed together with FORTRAN as an alternative programming style. LISP shortly became one of the widely used computer programming languages in the AI area. Then different languages were developed to handle specific formalisms. PROLOG works with symbolic logic to implement the predicate logic formalism. OPS5, on the other hand, implements production systems. Object-Oriented languages, such as Smalltalk, are used for implementation of more strict formalisms. These tools are known as the *fifth generation* of computer languages.

The second module of ESs, the inference engine, performs the reasoning over the knowledge base. It can be designed to use backward or forward chaining methods during the course of reasoning. Forward chaining is the strategy of working from the evidence to hypothesis, sometimes called a data-driven technique. Production systems, having rules in the form of

if condition then action

pairs, is a typical example of forward chaining. The *action* will be performed when the *conditions* are met. Backward chaining, on the other hand, is goal-driven, and works from the hypothesis to the evidence. Predicate logic uses this method during its reasoning process. The goal is divided into subgoals to be satisfied. Subgoaling continues for each subgoal that is not satisfied by the domain knowledge base.

Within the given definitions, game playing programs were one of the first applications of ESs [6]. The fundamental idea behind these programs is state space search. After defining the initial and final states, such as the initial state of a chess board and check-mate, a generate-and-test method is used to reach the final state. A possible solution of the state is generated through the predetermined operations on state and tested for being a solution. If it is not, then a new solution would be generated until all possible alternatives were exhausted. The main search techniques, breadth-first, depth-first, and heuristic search, were developed during what is sometimes referred to as the Classical period of AI research.

The Romantic Period of AI, from the mid 60s to the mid 70s, was focused on trying to make machines *understand*. The main outcome of the period was to learn that AI systems do not have to *understand* a domain in order to solve a problem. In other words, the important point is to get the right answer, not necessarily by

the same steps of reasoning as a human would use [6]. Consequently, researchers put a special emphasis on knowledge representation. They explored the possibilities for encoding particular facts and general principles about the world in a way that they could be used by the computers in the course of reasoning. The basic knowledge representation formalisms, production rules, associative nets, semantic networks, frames, and strips were developed during this time.

The Modern Period of AI started with the applications of AI techniques, usually ESs, to the real world. Several commercial and research ES programs were released after 1975 which served to demonstrate the efficiency of the technique. Among these, DENDRAL, MYCIN, MACSYMA, and R1 were the most successful applications.

DENDRAL was developed at Stanford university during the late 60s. It determines the molecular structure of an unknown organic compound using a modified form of the generate-and-test paradigm, in an interactive process with the user. DENDRAL was based on a heuristic state space search. With MYCIN, also developed in Stanford University, researchers carried ESs several steps further. Having the purpose of assisting a physician for therapy and prescriptions, MYCIN was able

- to reach a conclusion with insufficient data through the uncertainty handling characteristic of the program,
- to explain the reasoning behind the conclusion, and
- to add and modify rules through a knowledge acquisition module.

It was based on the concept of backward chaining, i.e., subgoaling the goal until all subgoals are satisfied [6]. MYCIN has never been sold for commercial use, but

it was updated as a research tool, and descendant of MYCIN do have commercial applications.

MACSYMA, on the other hand, was developed at MIT during 1970s. It was a symbolic processor for matrix, differential, and integral calculus. Having proven its usefulness, MACSYMA has been updated ever since [9]. Maybe the most successful ES application is the R1 system for configuring computer systems, such as VAX. It has been run commercially for over 500 000 cases, and proved that ESs are reliable and powerful tools [6].

Scope of This Study

Despite the fact that the xenon oscillation is not a catastrophic transient, it requires operator attention. The usual way of monitoring for xenon oscillations is based on the Axial Offset (AO), where AO is defined as [10],

$$AO = \frac{P_{top} - P_{bottom}}{P_{top} + P_{bottom}} \quad (1.1)$$

where P is the power produced in a given region (top or bottom half) of the core. Practically, AO is detected by monitoring the power in both halves of the core independently using part length neutron sensitive ionization chambers. The difference between the signals from each chamber is then divided by the sum of the signals to obtain the AO value. Based on the AO, operators follow certain instructions to keep the AO within permissible limits around a target AO. The target AO value is defined as the AO at steady state full power operation, typically -10 to 0 %, and the target band is usually defined as ± 5 % of the target value [10].

There are different control schemes based on the availability of control mecha-

nisms. The most commonly used control mechanism of AO, as proposed by Randall and St. John [3], is based on Part Length Control Rods (PLCRs), which are moved around the center of the core. A shift in AO, either up or down, will then be controlled by moving the PLCRs in the proper direction. The control strategy will be completely different in a reactor if there is no PLCRs. Additional control mechanisms are the inlet temperature, boron content of the coolant, and positions of Full Length Control Rods (FLCRs). In any case, however, the strategy is quite simple and heuristic. The operator must respond to an oscillation with a sensible control action. The problem that must be solved is the quantification and the timing of this action.

Generally, nuclear reactors have AO response procedures in operating manuals, specifically designed for a given reactor. The procedure defines the direction, the magnitude, and the timing of the control rod motion. Responses described in these procedures are categorically heuristic and called bang-bang control [11]. The intent is to damp the oscillation by a single CR motion, preferably using PLCRs. The nature of the problem suggest that it should be possible to design an expert system program to follow the same, or a similar procedure, to control these xenon oscillations.

In a typical situation, the xenon transients are actually started by a power level change in the reactor, and they are not the ideal cases as explained above. When the control rods are moved to change the power level, some local reactivity will be inserted which changes the neutron flux distribution and initiates a transient in the xenon concentration, and therefore an oscillation in the spatial power distribution. Consequently, the reactor power may not remain constant during the transient. Considering the daily load changes that a reactor may have to follow, and the delayed

effect of the xenon oscillation, constant operator attention would be required in a load-follow mode. The long term delay in xenon feedback makes the response timing more difficult and having a reliable control system more necessary.

Sipush [10] et al. proposed the Constant Axial Offset Control (CAOC) strategy for load-follow operations utilizing continuous control to keep the AO within a target band. This heuristic procedure requires constant observation of AO monitors, and response with PLCRs when the AO tends to shift out of the target band. The continuous attention of the operator required for this process may easily be distracted as it happened in a specific AO experiment [10]. Although the target band is not a safety limit, a drifting AO may create unnecessary power peaks. A controller would not miss such a tendency of AO, and would keep it within the target band.

Because of the heuristic nature of the xenon control problem, it seems reasonable to implement the CAOC strategy using an Expert System . Furthermore, it should be possible to implement an ES controller which uses only measurable reactor parameters and is free of any reactor dependent *heuristic* control constant. Such an ES controller should simply simulate the reasoning that a reactor operator would perform during a load-follow reactor operation, and/or xenon oscillation. Basically, he/she would monitor a set of instruments to recognize the reactor status, and decide what to do given the technical specifications of the reactor and the target power level. Forward chaining characteristic of the decision making process led us to use production systems which also allows flexible data representation beside the forward chaining. Therefore, we used the OPS5 tool that implements production systems. An introduction to the OPS5 language will be given in Chapter 3.

An ES control strategy was implemented with a program called ACES (Axial

offset Control using Expert Systems). The idea used in ACES is to sample the reactor status with a certain time interval (every minute in this study) in terms of some readily available variables. The list of signals that ACES uses during the determination of corrective actions are power, reactor period, FLCR positions, PLCR position, AO, inlet temperature, and boron concentration. Using these signals, ACES controls;

- the core power to follow the demand,
- the AO to keep it within a target band, and
- the inlet temperature to keep the average core temperature constant.

The control actions are determined using a knowledge base formed by the differential rod worth curves, boron worth, and load demand for a time period.

CHAPTER 2. CONTROL OF SPATIAL XENON OSCILLATIONS

Almost all nuclear reactor designs are sensitive to spatial xenon redistribution provided that they meet the conditions previously described. For this reason, xenon induced spatial power instabilities have been analyzed for most of the reactor types, including Boiling Water Reactors [12, 13], Heavy Water Reactors [14], High Temperature Gas Cooled Reactors [15], and the Russian design VVER and RBMK reactors [16]. Among these studies, the PWR is the most commonly used reactor type. We will adopt the same approach and use a typical PWR core for the rest of this study.

After the spatial xenon oscillations were identified, the nuclear industry experimentally studied the stability of reactors and the effectiveness of the proposed control strategies [10, 17, 18, 19, 20, 21, 22]. At the same time, there were several occurrences reported to the Nuclear Regulatory Commission [23], regarding controlled xenon oscillations in PWRs. With these incidents, the importance of control strategies was recognized to keep reactors operable in terms of license and safety measures.

Early Control Strategies

It has been shown that the xenon oscillations are convergent for low flux levels [2, 17]. At high flux levels, however, they are divergent because of the strong xenon feedback. Since until recently, reactors were primarily operated at constant power,

early control strategies were based on the constant power assumption. The idea behind these strategies is to attack the first harmonics of the oscillation at certain times using the PLCRs. The magnitude of counter oscillation, created by moving PLCRs, must correspond to the first harmonics of the original oscillation in order to nullify it. The residual, second and higher order harmonics will die away rapidly due to their short time constants. Consequently, the oscillation would be damped effectively in a short time. A detailed review of variations of this idea is given by Bauer [11].

At constant power, xenon oscillations show an explicit trace in terms of the AO, which is necessary for the following control strategies. This trace is used to recognize the transient, and to determine the time and magnitude of corrective action. The basic procedure is composed of two PLCR strokes as shown in Figure 2.1, known as Bang-Bang control. The parameters of control, t_1 , Δt , and $\Delta \Sigma_c$ are determined by using the AO trace. Control initiation time gets closer to the AO peak as core life increases because of decreasing stability of the core. Duration and magnitude of the control depend on the magnitude of the AO at the control initiation time [11].

Direct Offset Control (DOC), First Overtone Control, Savannah River Laboratory (SRL) Indirect Control, and Overstressed DOC are modified versions of Bang-Bang control. The variations are on the time of control and the size of PLCR stroke used for control [11]. In DOC, response starts when the AO curve crosses the average value, and PLCRs are used as often and as much as necessary to keep the AO around this value (Figure 2.2). The First Overtone Control is similar to the Bang-Bang control, and differs only in the magnitude of PLCR strokes. PLCRs will be inserted 1 hour before the positive peak to shift the oscillation phase by 180 degrees, i.e., to shift

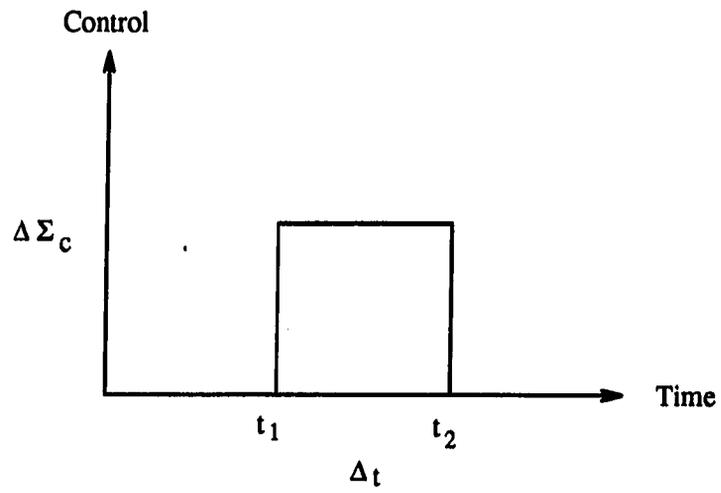


Figure 2.1: Bang-Bang control

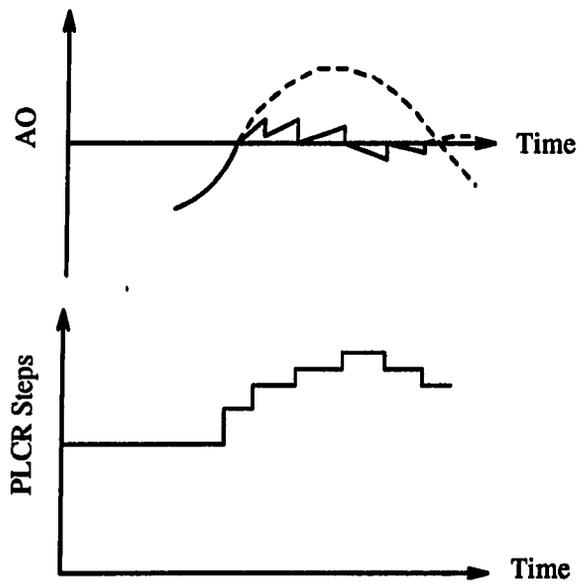


Figure 2.2: Direct offset control for AO oscillations

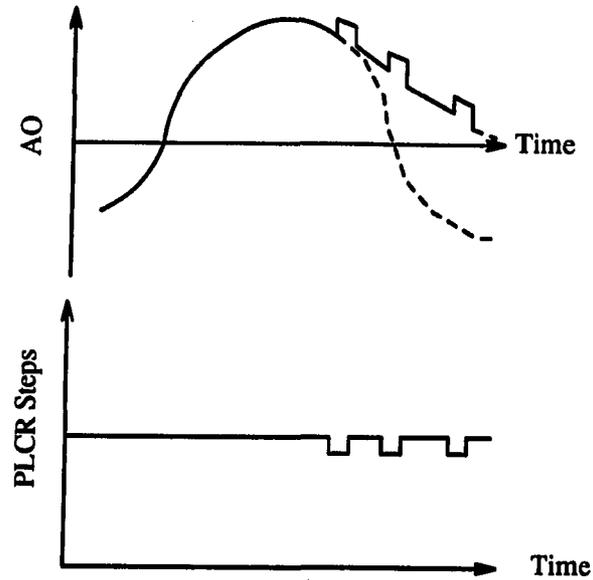


Figure 2.3: First overtone control for AO oscillations

the AO close to a negative peak. Then, the operator waits until the negative peak in the AO, and uses another PLCR stroke to increase the AO to the average value (Figure 2.3). SRL indirect control is a completely different approach for damping the AO oscillations. Control action will be deferred until the AO peak has been passed, and a decline for 1 to 2 hours is observed in a free oscillation mode. Then, PLCRs will be moved to increase the declining AO. After an hour of control, AO will decline to the target value (Figure 2.4).

These strategies were tested on the Point Beach Plant in Wisconsin, and the First Overtone control was found as the most effective control procedure to eliminate a spatial oscillation [11]. The First Overtone Control strategy was also used during the xenon oscillation experiments in the Maine Yankee Plant proving its effectiveness [22].

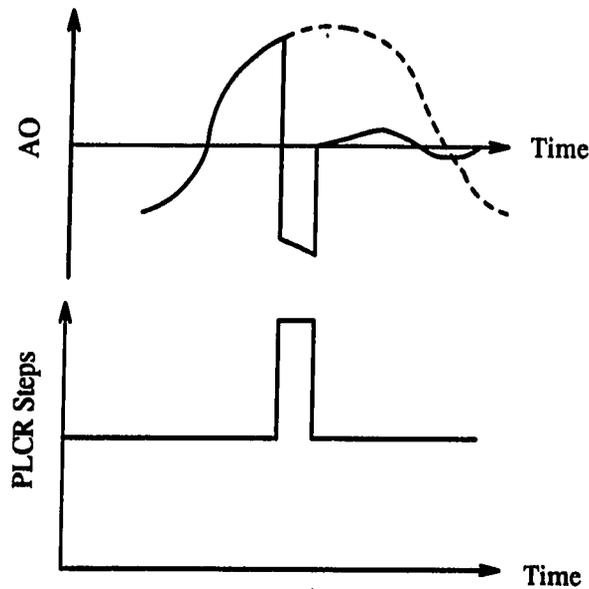


Figure 2.4: SRL indirect control for AO oscillations

The timing of the control is the main problem of these strategies. In most cases, the PLCRs have to be moved approximately one hour before the positive AO peak. Therefore, the operator must closely watch the AO curve to estimate the correct time for insertion, which might take a long time. Although these strategies will be valid only for constant power operations, they are the basis of current AO control procedures used in most plants.

An AO control procedure actually in use in PWRs is as follows: When the oscillation is detected, the operator waits for the next positive peak of the axial offset, and calculates the average AO and the time of the peak. Around an hour before the peak, the operator inserts the PLCRs to reduce the AO to the average value. After reaching the average AO, the operator allows the AO to decrease below

the average by an equal amount to that initially above the average. The AO will be kept constant at this level for several hours by PLCR moves as necessary. Then, the operator withdraws the PLCRs gradually to take the AO to the average value [24].

Nuclear reactors do not usually operate on base load anymore. Most of the reactors follow a daily load change, known as a load-follow mode of operation. When a reactor operates in a load-follow mode, the AO oscillation control problem becomes more complex than it is for steady state operation. Because of the power redistribution during the transients with each CR motion, it is hardly possible to observe free oscillations for making use of the constant power level control methods. As a result, the load-follow mode of operation has become the subject of recent studies of xenon oscillation control. Besides the constant power applications [17, 18, 19, 20], load-follow experiments and analyses were performed [10, 21, 25].

Advanced Control Studies on Xenon Oscillations

As the operating mode for reactors has changed, a need for a better AO oscillation control strategy arose. Several automatic controllers were designed for xenon oscillations, but only a few were actually used because of their simplicity [26]. They usually consist of single input-single output control loops based on oversimplified models, and overall performance was kept high by tuning the individual loops. To improve the operational economy, more advanced controllers were developed to reduce power cutbacks, fuel failures, and to increase the fuel utilization.

Optimal control approach

The first improvements of automatic control were directed to the optimization of the xenon oscillation control problem. An optimal control problem starts with the definition of an objective function, expressing the control objective in a mathematical form. Then, the optimality will simply mean the minimization or maximization of the objective function. For the xenon oscillation control problem, a quadratic function of the spatial power distribution error, the amount of control used, and the control time may adequately form an objective function. It is always possible to use different terms in an objective function which may be useful for a particular purpose, such as the maximum peaking factor of the power distribution. Christie and Poncelet [27] used such an objective function, for improving the manual control strategies described by Bauer and Poncelet [11]. A sensitivity analysis of the main parameters of manual control, such as the duration of control, was the main purpose of their study.

Having defined the objective function, the state equations of the model are derived by introducing a clear distinction between the design and control models of the core. It is very often necessary to use nonlinear design and linear control models. The solution of an optimization problem is classified by Karppinen [26] as Distributed Parameter Control, Variational Calculus, Dynamic Programming, Time Optimal Control, and Linear-Quadratic and Stochastic Control.

Tzafestas [28] gave an excellent survey of the Distributed Parameter approach applied to xenon spatial oscillations in nuclear reactors, and covered various methods of the solution of linearized control problems with a quadratic objective function. Surendran [29] analyzed the xenon spatial oscillation control problem by using a distributed parameter control method. He showed that the spatial approximations

may lead to unexpected results, such as excessive control rod movements. Since the distributed parameter approach requires those spatial approximations, such a controller has not been used for controlling a more realistic model [26].

Variational Calculus provides a more general solution for nonlinear optimization problems provided that continuous variables are used. Stacey [30] and Hanke [31] used this approach in the xenon oscillation problem. Both used nonlinear equations with an objective function that contains quadratic spatial power distribution and control amount terms. Being a sophisticated analytical approach, Variational Calculus is quite inept in handling discrete variables. As a result, as Stacey pointed out in his study [30], a realistic representation of control rod motion remains to be solved in this method.

The Dynamic Programming approach requires the discretization of the system which allows a more realistic representation of a reactor core. Discretization also solves the problems of having nonlinear equations, constraints, and nonanalytical objective functions. The solution of optimal control is obtained as a feedback in discretized space. Being developed to increase computational efficiency, the computation time grows linearly with the number of steps considered in the calculation of the feedback control, and grow faster with the number of control and state variables. Problems with more than three or four variables may become unmanageable [26]. In spite of the computational limitations, the method has been successfully applied to PWRs by Stacey [32], and to Candu reactors by Purandare [33].

The time optimal control method is developed by applying the Pontryagin Maximum Principle to the minimal time problem. The idea is to control the real cause of the oscillation, instead of forcing the spatial power distribution to a desired shape,

in minimal time. Christie applied time optimization to manual control strategies. Together with some others listed in Ref. [26], Schulz and Lee [34] studied the time optimal control of spatial xenon oscillations. Schulz claimed in his paper that time optimal control can also accomplish a similar objective as the Constant Axial Offset Control strategy which will be used in this study. Chiang [35], on the other hand, derived an effective bang-bang control strategy by the minimal time approach which makes use of positive, zero, or negative first harmonic mode control.

The most widely used result of optimal control theory is the Linear Quadratic Gaussian approach [26]. The linearized state equations are used in quadratic objective functions together with control parameters. Optimal control is found in a state variable feedback form if no constraints are used. Cho [36] used this approach to design a controller for xenon oscillations in the load-follow mode. The system can be extended with Gaussian statistics, to estimate the unmeasurable variables by a Kalman filter. Onega and Kisner [37] and Pack and Chambré [38] made use of stochastic methods and Kalman filters in their optimal control studies.

Adaptive control approach

Several characteristics of nuclear systems encourage the use of the adaptive control strategies. The effects of *aging* on some of the system parameters such as fuel depletion, the complexity of a nuclear system having nonlinear feedbacks and time varying constraints, and the dependency of parameters on the power level are some of these characteristics. An adaptive control system minimizes an objective function usually defined as an error array, the difference between the plant and model parameters. The error term is fed into the adaptation mechanism to generate the control

action. Deterministic adaptive control methods lead the plant to follow a model, whereas stochastic adaptive systems generate control signals for the plant based on a set of target values.

Being a relatively new strategy, adaptive control systems have not been actually applied to nuclear reactors yet. There are extensive studies on power control for a load-following reactor using adaptive control schemes [39]. For the spatial xenon oscillation problem, modeling of the system creates some difficulties. In a recent study, Berkan et al. [40] showed the possibility of using adaptive systems on spatial xenon oscillation problems. The Reconstructive Inverse Dynamics concept is based on sending the *mirror image* dynamics back to the plant provided that the image is reasonably accurate. This results in compensation of the forward dynamics and maintenance of a continuous dynamic equilibrium. Meanwhile, the system is forced to follow a desired trajectory by a proportional controller that reconstructs the compensated forward dynamics. An adaptive correction is included in the design to take into account possible defects in the *mirror*. Although this is not a complete adaptive controller, the study shows its potential for applications in this subject.

Heuristic Control Studies

The spatial control methods, which are not based on any formal optimization technique but on intuitive principles, are considered as Heuristic Control. Almost all of the early control strategies introduced previously in this Chapter fell into this category. Bang-bang control, DOC, First Overtone control and others are the heuristic solutions to the xenon oscillation problem developed through the experience and intuition of reactor operators, human experts of the subject domain. These control

strategies proved to be effective control strategies via experimental [22], and theoretical studies [27] for constant load operations.

There are, however, some difficulties in application of these techniques to a load-following reactor. First of all, it is impossible to decouple the global power control from the AO control in load-follow mode. Therefore, the free AO oscillations can hardly be observed to make use of these techniques because of the continuous changes in the spatial power distribution due to CR motions. A heuristic Constant Axial Offset Control (CAOC) strategy is developed for the load-following reactors [10]. The idea is to keep the AO within a target band with continuous control of PLCRs. Although a similar strategy is found to be quite ineffective for constant power operation mode [11, 26], it is relatively effective in load-follow mode considering the continuous changes in power and AO.

On the other hand, most of the analytical methods require a reactor core simulation with a certain degree of accuracy. There is no general answer to the question of how accurate the control model should be to obtain a good performance. The real system is, however, highly nonlinear, and the system parameters are time dependent. Consequently, the simple models won't be reliable enough for real application whereas a complex model will have computational limitations. The simple models are in use for estimating the reactor response in general, while the complex models are used as design-oriented off-line methods [26]. Even the Linear Quadratic method which allows the most realistic models to be designed, has problems with modeling of the control rod motion. Although adaptive control schemes are able to handle some of these difficulties, it has the same modeling problem. Consequently, old and reliable heuristic control strategies are widely used in current reactors.

Recent developments in computer technology related to expert systems allow us to translate these heuristic strategies into a computer program in terms of a knowledge base and rules. An expert system controller would not need any modeling other than the verification process. It is possible to perform this part of the design study without modeling, by coupling the controller with the reactor unidirectionally. By this way, the expert system will get the reactor data, and generate the corrective action as an advisor, which will be applied with the consent of the expert, i.e., the reactor operator.

Advisory expert systems are usually applied to monitoring and diagnostics of reactor operations in Nuclear Engineering [41, 42, 43, 44, 45]. The automation of the next generation of nuclear reactors is discussed by Uhrig [46], in which he emphasized the importance of the application of AI techniques, including ESs, in Nuclear Engineering. On the other hand, Expert System applications in xenon oscillation control problems are quite limited because of the cost of verification of such a program. In order to use an on-line ES, code should be verified thoroughly by testing it on simulators, which may cost a great deal of computer time. There has been a study at Iowa State University, on the implementation of an ES for xenon oscillation control, showing the possibility of designing such a system [47].

Since the ES controllers are usually composed of rules elicited from an expert on a particular reactor, applicability of the same rules to other systems is questionable. This is one of the main drawbacks of ESs. The other disadvantage of an ES control is the lack of optimization. The solution generated by the ES may not be the optimal solution. However, proper selection of an ES language may solve the applicability problem. An ES can be designed to have generic rules which adapt themselves to

system specifications during runtime. Production systems, and particularly the OPS5 language, allow such an approach because of their flexible structure.

CHAPTER 3. A PRODUCTION SYSTEM LANGUAGE: OPS5

An ES xenon axial oscillation controller should recognize the reactor core condition based on available parameters, use this information for determination of control actions, and present the corrective actions to the operator if necessary. The flow of information is clearly a data-driven type for this particular case, which favors a forward chaining ES design. Forward chaining is most appropriate when there are many equally acceptable goals, and a single initial state [48]. This definition fits the xenon oscillation control problem remarkably well. One of the forward chaining formalisms is the production systems. The production system languages additionally have flexible data representation which allows designing adaptive generic systems. Consequently, the xenon control ES will be implemented using the OPS5 (Official Production System) language since it is readily available on the Iowa State University VAX computer system, and has the desired features.

The basic model of the production systems is the same for any familiar procedural languages. Both have data, a program, and an executor. Data store, often called *working memory* in production system terminology, serves as a global database allowing symbols representing physical objects or facts, or conceptual objects. A set of rules constitutes the program part of the production systems. Rules, or *productions* are condition-action pairs stored in the *production memory*. An *inference engine* exe-

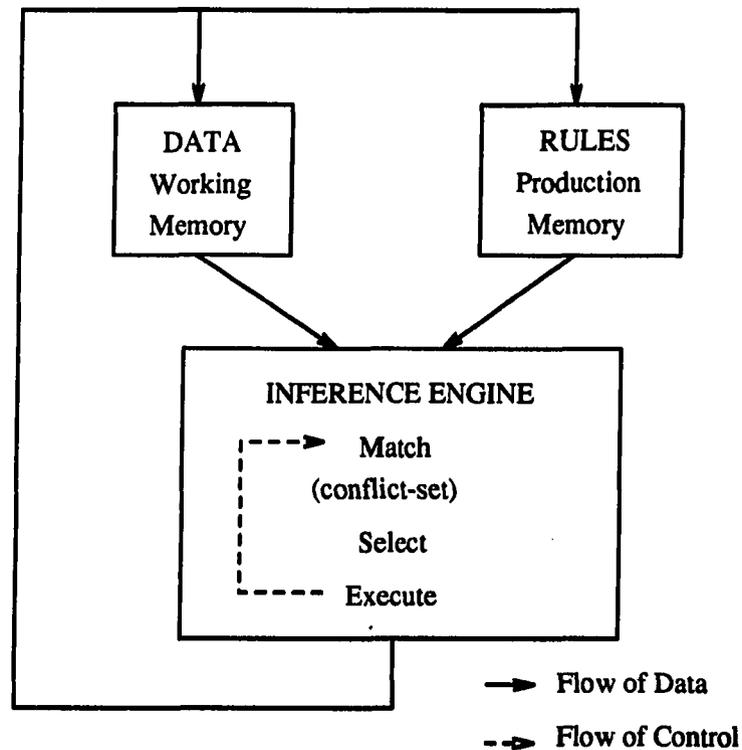


Figure 3.1: Architecture of production system language OPS5

cutes the rules, where the process is referred to as *firing* rules in production systems. The inference engine searches for the applicable rules for the current state, and selects one to fire. The basic architecture of OPS5 is given in Figure 3.1.

Data Representation in OPS5

Data memory, which contains information about the current state of knowledge during the problem-solving process, is accessible from any part of the program. Memory items, referred to as *Working Memory Elements* (WME), are attribute-value systems which can be declared as simple strings, or complex structured objects. The flexibility of allowing simple string attributes provides a basis for representing dif-

ferent forms of knowledge. On the other hand, structural objects helps to store the database in a tabulated form. This characteristic of OPS5 is extensively used in ACES as a knowledge base storage sector in the program. Particularly, the differential reactivity worth of control rod banks are stored in structured WMEs.

All attributes of WMEs and user-defined functions must be declared in OPS5, and the declaration must precede all the rules. The LITERALIZE statement is used for declaration of WMEs. For example, the statement

(LITERALIZE DWORTHS BANK STEP WORTH)

declares a structured WME named DWORTHS (Differential rod WORTHS) with attributes BANK for control rod bank identification, STEP for current position of the bank, and WORTH for the differential worth of the bank at the given position. There is no limit to the number of WMEs created by using the same declaration. In other words, we may have as many DWORTHS WMEs as necessary for storing the knowledge base completely. Therefore, the accuracy of the discretization of the differential rod worth curve will not be limited by the language, but only limited by the accuracy of the system.

When a WME is introduced into the working memory for the first time, it is assigned an integer time-tag, or *recency* number. This number is modified whenever this WME is modified as an action of a rule. OPS5 makes use of the *recency* number during the selection of a rule to fire, and we used this characteristic of OPS5 for providing necessary flow control.

WMEs are building blocks of the condition part of a production in OPS5. Productions are selected to fire provided that there is a set of WMEs that matches with the ones appearing in the condition part of the rule. WMEs can be created, modi-

fied, or removed by the action part of the rules, where these actions provide learning and forgetting features for an ES. Creating WMEs may lead to an overloading of the working memory if unnecessary WMEs are not removed from the active working memory. Since such an overload will considerably slow down the matching, and hence, the decision making process, a production system program must have *garbage collection* rules to avoid overloading of the working memory.

Production Memory in OPS5

Rules of the production systems are similar to formal grammar rules of *if-then* pairs. The Left Hand Side (LHS) of the rules contains the information about the situation, usually a combination of Boolean clauses based on WME values. The use of Boolean operators AND, and OR is implicit in OPS5, while NOT is explicit, simply negation of a condition. Besides the Boolean operators, predicate operators can be used in the LHS of a rule. A simple set of conditions

```
(PWRCTRL ^reactivity < r > )
(CRODS ^bank < b > ^step < s > )
(DWORTHS ^bank < b > ^step < s > ^worth <= < r > )
```

will find the CR banks that may be used for power control via the matching process.

An English version of this set will be;

```
IF the reactivity of the system is  $r$ 
AND the CR bank  $b$  is at step  $s$ 
AND the differential worth of rod  $b$  at step  $s$  is less than and equal to  $r$ 
THEN ...
```

This example explains the use of AND and predicate operators. The OR case is

simulated by writing two separate rules with the same parameters. It is also possible to form conjunction and disjunction within a rule for more specific cases. Design of an ES must involve effective use of these characteristics of OPS5 for an efficient conflict resolution process which mainly defines the length of the decision time. The same example also displays the adequacy of OPS5 for writing generic rules. We didn't use any constants in the condition part, and it may match with any CR bank for an arbitrary value of the reactivity provided that the predicate operation is satisfied.

The Right Hand Side (RHS) of a rule is a list of modifications to be done on WMEs. Primitive actions are MAKE, MODIFY, and REMOVE. It is also possible to call built-in or user-defined external routines, and perform write and read functions. A complete set of functions that can be used in the RHS of a rule can be found in the literature [48, 49, 50]. OPS5 can communicate with a wide range of computer languages including conventional FORTRAN through built-in definitions. Therefore, user-defined functions can be written in many languages. A set of FORTRAN77 routines are used in ACES for numerical data processing.

Rules can be designed for flowing in either direction. In forward chaining, the action part of a rule modifies the working memory making new rules ready to fire. In backward chaining, however, the action part is used to create new subgoals.

The OPS5 Inference Engine

The OPS5 inference engine processes data in three stages, matching, selection, and execution as seen in Figure 3.1. This strategy is also known as the *Recognize-Act* cycle. The cycle repeats itself until an explicit HALT statement is encountered, or the matching process ends with no match. Rules are treated individually during the

recognize-act cycle, with no ordering relationship imposed.

The matching process involves finding sets of WMEs that satisfy the LHS of rules. Each match will be stored in a *conflict set* as pairs of a rule name and a list of recency numbers of matching WMEs. On a given cycle, there may be any number of matching, or *instantiations*, including none. For a consistent match;

each condition element should match with a WME,

there should be no WMEs matching negated conditions, and

each variable must be consistently bound to one value for all occurrences in the LHS [48].

Creation and modification of the conflict set is optimized to avoid making redundant matches. Optimization characteristics are also given by Brownston [48].

Once the conflict set is formed, two alternative strategies can be used for the selection process. A set of tests are performed on the instantiations of the conflict set in both strategies. Each test orders the instantiations and transfers the dominating ones to the next step until only one instantiation is left for firing at any time. In the LEX (LEXicographic ordering) strategy, the *conflict resolution* procedure involves the following steps;

- In the *Refraction* step, all instantiations previously selected and fired are deleted from the conflict set. This avoids firing the same rule with the same matches more than once.
- In the *Recency* step, all remaining instantiations are ordered on the bases of recency of time tags of matching WMEs. All instantiations that do not have the largest recency number are discarded first. Then they are ordered by the second largest recency number, and so on. If there is only one instantiation left

Table 3.1: A sample conflict set in OPS5

I1	Rule-5	1215	1225	1224	12	5
I2	Rule-1	1224	1215	12	1113	
I3	Rule-12	1213	1212	12	5	4
I4	Rule-9	1224	1215	12	1113	
I5	Rule-1	1224	1215	1113	11	

at any step, it is chosen to fire. If there are more than one instantiation left, the next test will be performed.

- In the *Specificity* step, the inference engine performs relational tests on the instantiations that have the same recency numbers. Relational tests are performed against the constants and variables required to compute the match. The instantiation that requires the largest number of tests will be chosen to fire since it is more *specific* than others.
- In the last stage, if there is still more than one instantiation left after the specificity test, the inference engine arbitrarily selects one of them to fire.

The second strategy, MEA (Mean-End Analysis), differs from the strategy LEX at the Recency step. MEA puts an extra emphasis on the first condition element of the rule. The inference engine first orders the first recency numbers of instantiations, keeps the ones that have the largest recency number, and discards the rest. MEA continues the same as LEX for the rest of the conflict resolution procedure. This strategy provides an additional control over the flow of the program. If the user places the goal as the first condition element in a rule, then the program flow won't be distracted by a recent WME which is not a goal. The strategy MEA will be used in this study.

Consider a conflict set with five matches as given in Table 3.1 where I stand for the instantiations, and integers are recency numbers of WMEs in the order of appearance in rules. The conflict resolution procedure of the strategy LEX will immediately select the instantiation I1 in the recency stage, since there is no other instantiation that has a recency number of 1225 or greater.

The strategy MEA, however, discards instantiation I1 since its first condition element carries a recency number smaller than some others. The MEA picks I2, I4, and I5 first because of the largest recency number of the first match element. Then, it compares the second largest number regardless of the order which is the same for all selected instantiations. Third numbers are also the same, 1113 for these instantiations. The instantiation I5 will be discarded later based on the last recency number, which is smaller than the other two. The MEA carries the selection process to the Specificity step with instantiations I2 and I4. The specificity test will decide which instantiations is to be fired at this stage. If they have an equal number of tests for matching, then one of them will be selected randomly. Consequently, instantiation I1 will be fired after the conflict resolution procedure if the strategy LEX is used, while I2 or I4 will be fired if the strategy MEA is used.

Strategy selection is an important part of the design process, and must conform with the ES requirements. The strategy MEA is most adequate for forward chaining programs with an explicit flow control. If there are no clear paths to follow and each rule is equally applicable at any stage, the strategy LEX will help. The ES ACES uses the strategy MEA for providing an explicit flow control.

Organization and Control in OPS5

In OPS5, knowledge can be embodied in three places, working memory, production memory, and external functions. Database knowledge, which is likely to be updated more frequently, is stored in working memory. Although it is also possible to store this type of knowledge in production memory, it complicates the rules, and increases the number of rules dramatically. The second type, Problem-Solving knowledge is considered as a more stable one, and is stored in production memory for most of the cases. Finally, control knowledge can be stored either in working memory or in production memory. Control knowledge is used to direct the sequence of the problem solving steps. It can be embodied into the rules by writing explicit constants, or stored in working memory. In OPS5, the external functions and procedures are designed for tasks that are tedious but not knowledge intensive, however, it is theoretically possible to store some knowledge in these program segments.

The control of an OPS5 program starts with the selection of a conflict resolution strategy. The rest of the flow control depend upon the programming techniques. The designer should effectively utilize the conflict resolution strategy to control the direction of flow. Having a control WME should be considered as an additional way of flow control in OPS5. Consequently, the designer selects the proper means of storing the knowledge, and the control strategy based on the specific case study. The organization of ACES will be explained in the next chapter.

CHAPTER 4. DEVELOPING THE EXPERT SYSTEM ACES

In load-follow operation of reactors, power control can not be decoupled from the AO control. Therefore, an AO controller must have power control as a part of its strategy. In this case, however, the complexity of the system makes the determination of control gains almost impossible in advanced control methods, unless the system is simplified. On the other hand, an ES controller does not require modeling as a part of it. Therefore, the complexity of the system enhances the idea of using an ES controller for the spatial xenon oscillation problem. Additionally, the power and xenon oscillation control strategies are heuristic in nature as shown previously [10, 11, 27, 47]. Since the domain expert, i.e., the reactor operator controls the system based on the information available at the control console, an ES spatial xenon oscillation controller can be designed to utilize these reactor parameters.

Development of Rules for ACES

There are two main goals of the spatial xenon oscillation control problem in load-follow operation: (1.) Keeping the power around a target value, and (2.) Keeping the axial offset within a target band. It is assumed that the user will provide the power schedule to be followed, and the target AO band width prior to use of ACES. The power schedule is determined by the daily load of a particular reactor, and it

usually consists of operating at full power level during the daytime, reducing power to a lower level for night time, and recovering back to full power in the morning. The power levels may vary for each reactor.

The target AO band, on the other hand, can be defined as a band around the AO at full power steady state operation. A characteristic number for steady state AO is around -5% , and it increases to zero as power goes to zero [10]. We also adopted the same power dependency of AO in ACES. It reads the steady state AO value, and generates the power dependent function to calculate the target AO at any time. The target band is defined as $\pm 5\%$ of this target AO value in ACES.

Knowing the targets, it would be easy to find the necessary corrective actions only if the power and AO control were independent phenomena. Unfortunately, any action for power control has some effect on the AO, and vice versa. For example, inserting CRs to reduce the power creates a decrease in AO, pushes the AO down in the axial direction. Or, moving PLCRs to control an AO shift will introduce an unpredictable reactivity to the system. Additionally, long term xenon feedback should be taken into account during the determination of corrective actions.

ACES uses three control parameters to achieve its goals. They are FLCR bank positions, the PLCR bank position, and the boron concentration, and each one is used for a different specific purpose. FLCRs are used specifically for power transient control purposes. The necessary reactivity to change the power level will be provided by using FLCRs. After steady state operation is reached, the only changing reactivity effect will be the xenon concentration change, and it will be compensated by changing the boron concentration. Finally, the PLCR bank will be used for AO control whenever it tends to diverge from the target band. Given the reactor core

Table 4.1: List of reactor parameters used in ACES

Power	Inlet Temperature
Period	Outlet Temperature
Axial Offset	FLCR positions
Boron Concentration	PLCR position

status, the ES controller must answer the following question: Is any control necessary for any particular reason, and if so, how much and which direction should the control variables be changed?

To determine whether a control action is necessary or not, the ES must evaluate the core status. A usual approach to such a problem is to use pattern recognition techniques. However, this technique requires the definition of specific patterns for each reactor, which is against the philosophy of ACES. Instead, ACES samples the reactor status with a certain frequency, and use the data for computing the necessary control actions which may be doing nothing at all. The parameters sampled from the reactor, which are used for determination of the control action are already monitored reactor parameters as it can be seen from Table 4.1. Therefore, the primary task of ACES reduces to the determination of the direction and the amount of control action. In this respect, a simple logical deduction will be enough for determination of the direction of a necessary response. For instance, it should move the PLCRs down if the AO is drifting down, and it should insert FLCRs to reduce the power if the load is decreasing. Xenon control, however, needs a more detailed reasoning for determination of a correct response.

If the reactor power is decreasing, there will be an increase in the xenon concentration which introduces a delayed global negative reactivity to the system. Even though this is a delayed effect, if it is not compensated for, the xenon build up will be

faster in low flux regions, such as around the core boundaries, because of the lower burn out rate. The more the xenon builds up in these regions, the more the flux is depressed. This interactive process leads the spatial flux distribution to peak at the center of the core, a process known as a *power pinching* effect [10]. The power pinching can be avoided by creating a local adverse reactivity effect, increasing local reactivity if a xenon increase is expected, or vice versa. Therefore, ACES changes the boron concentration during power transients to provide this adverse reactivity.

The second problem of the control strategy is to find the necessary amount of control. How much do we need to insert FLCRs to provide the required load decrease? The same question is applicable for moving PLCRs, and changing the boron concentration. Since the idea of this study is to design a controller independent of reactor specifications, a heuristic determination of the amount of control for a particular reactor is unacceptable. Instead, we will use common reactor parameters, such as differential rod worth curves, to determine the required amount of control as a feedback to the system.

A different kind of knowledge will be used in the determination of the control amount for power control in ACES. Some simple equations, that are used by domain experts in a *back of the envelope* calculations, are stored within user defined functions. Consider an ES, which is able to read the current power and the period of the core. Then, the reactivity of the core can be calculated by,

$$\rho = \frac{0.0848}{T} \quad (4.1)$$

where T is the reactor period. This approximation is valid as long as the reactor period is greater than 80 seconds [4]. Knowing the reactor load for a time period, the amount of reactivity that should be provided to meet the load at the next time

step can be calculated by using,

$$P = P_o e^{t/T} \quad (4.2)$$

where P is the target power for the next step, P_o is the current reactor power, and t is the time step size. The difference between the required and current reactivity, which we will refer to as the target reactivity, has to be provided by moving the FLCRs. If the differential rod worth curves for CR banks are available to the ACES as a knowledge base, it will be easy to find how many steps of CR motion would be required to provide the target reactivity for power control.

The power control is handled by two different strategies for the steady state and transient (intentional power level changes) operations. During the steady state, power and xenon feedback control strategies concur since the only expected reactivity effect is the xenon feedback. The boron concentration is then used to compensate for the reactivity changes during the steady state which solves both control problems. The amount of control can be calculated by using the target reactivity provided that the reactivity worth of a unit boron concentration change is available. A characteristic figure for the reactivity worth of a ppm (part per million) concentration change is approximately $0.01 \Delta\rho/\rho$, and it linearly increases -or decreases- with the change of boron concentration. Having this value in the database, it is possible to calculate the required amount of boron concentration change. However, there is a minimum limit for boron concentration change which is determined by the accuracy of the monitoring devices. ACES uses this minimum controllable amount as a discrete step size for boron concentration changes. Therefore, the user should supply this unit amount and corresponding reactivity worth of boron change in the database. Discrete step changes in boron concentration is especially crucial during the transient power

control.

Transient power control is not as simple as the steady state power. First, FLCR motion has an immediate effect on the spatial power distribution, which complicates the AO control procedures. Second, the net reactivity effect of boron concentration change should be taken into account while calculating the control amount. The xenon control during a transient may or may not help the power control. Therefore, before the calculation of the FLCR motion for power control, target reactivity should be updated for the change of boron concentration of the system. The corresponding reactivity worth of a boron change required for xenon control should be added to the target reactivity.

Changing the boron concentration for xenon control has a completely opposite effect on power during a level change after a steady state operation. For instance, as power level decreases, an increase in xenon concentration should be expected. Therefore, the negative reactivity insertion by the increase in xenon absorption must be compensated for by decreasing the boron concentration. This means adding positive reactivity to the system during a power decrease. This strategy works effectively for relatively short power transients. However, if the power continues to decrease when the xenon concentration reaches its peak and begins to decrease, the ES should also switch to boration of the system from deboration.

It is a problem to find the correct amount of boron that should be changed for xenon control. There is no simple equation which can give the xenon reactivity feedback during a transient based on the signals coming from the reactor. Consequently, we adopted the following approach; the amount of control is initially set to zero based on the minimum step change of boron concentration. Then, based on

the expected response from the reactor, we increase the control amount by steps if the reactor response is opposite to what is expected, and decrease the amount if the reactor responds as expected but in excess. Specifically, we expect an AO shift to the bottom of the core as a result of FLCr insertion while the reactor power is decreasing, or vice versa. Therefore, if ACES observes that the AO is shifting upward during a power decrease, it increases the boron control amount, and decreases back if AO tends to shift down again. As a result of this strategy, the core AO will be kept around the target value during a power transient. This strategy is successfully used for the short term power transients. In long term transients, however, the expected response may change its direction as a result of the xenon feedback. Then, the same strategy is utilized by switching the expected reactor responses when the change in xenon response is sensed indirectly. If the AO tends to shift upward even though the boron control is increased, and if it exceeds the upper control limit set for the AO control, ACES switches the direction of response.

With the given power control strategy, the spatial power distribution control becomes easy. Since the xenon feedback is kept under control from the very beginning of a transient, a simple approach like moving PLCRs a step in the direction of AO shift is enough to control the residual power distribution changes, in compliance with heuristic CAOC strategy. The problem of using PLCRs is the determination of reactivity insertion to the system as a result of its motion. It is not practical to use the differential rod curve approach as we did for FLCRs, since any change in power distribution will change the differential worth of PLCRs. Therefore, the reactivity effect of PLCr moves will be detected and corrected through the power control rules as illustrated in the following cycle of control.

Working Memory of ACES

Working memory elements are the variables of the OPS5 language, and they will be used for two purposes. The database of differential rod worth curves will be stored in a set of a WME having the bank name, the position, and the corresponding reactivity worth as its attributes. Another WME will keep the information about the boron concentration. It has attributes for the minimum unit of change, worth of unit change, number of units that should be used during power transients, and the amount of change done at each time step. The other two WMEs that contain the database are for the average temperature and the power requirement. ACES asks for a file name that contains all of these data but the power history. In order to initialize and make changes in load demand, a subroutine requesting these data from the user is written in FORTRAN77 and coupled with ACES.

The working memory of ACES contains a set of temporary WMEs to provide information flow among the rules. Most of these WMEs are created and deleted at each cycle of control. Among these WMEs, one named GOALS stores the current goal of the system to control the direction of information flow. A detailed explanation of this WME will be given in the following section. As it is usual for most computer languages, these WMEs are declared at the very beginning of the program which can be seen in App. A.

Production Memory of ACES

The production memory of ACES can be discussed in parts since the rules are grouped, based on their context. The first set of rules carries out the initialization

process. As a default option of OPS5, attributes of created WMEs are set to *nil*, regardless of data type, if they are not initialized. Therefore, special attention should be given to attributes that will be bound to numerical atoms later. Additionally, the database should be added to the active working memory. These problems are solved by having a set of rules that initialize the necessary WMEs.

The basic structure of the production memory of ACES is given in Figure 4.1. Basically, ACES samples the reactor with a given time interval, evaluates the data, finds the necessary corrective actions, and sends this information back to the reactor. Receiving the necessary data from the reactor, a compound rule is used for creating temporary WMEs for future evaluations, and for displaying the information received. Although this rule seems to be overloaded with conditionals and actions (see Rule # 1.5 in App. A), having a set of rules for this purpose requires extra conflict resolution process.

After the data are processed, and necessary WMEs are created, control is passed to the second group of rules. This group determines the corrective action for AO. There are only two rules in this set, taking care of the AO problems. The main control rule moves the PLCRs one step in the direction of AO shift if AO is out of the control band. The control band is defined as a band around the target value which determines the time of response. Obviously, the control band is smaller than the target band of AO. By responding when the AO is out of the control band, we keep the AO within the target band. The width of the control band is a user defined value, and is set at $\pm 2\%$ in this study. If AO is within the control band, the second rule will simply pass the control to the next set of rules.

The third set of rules relates to the utilization of boron in power control, and

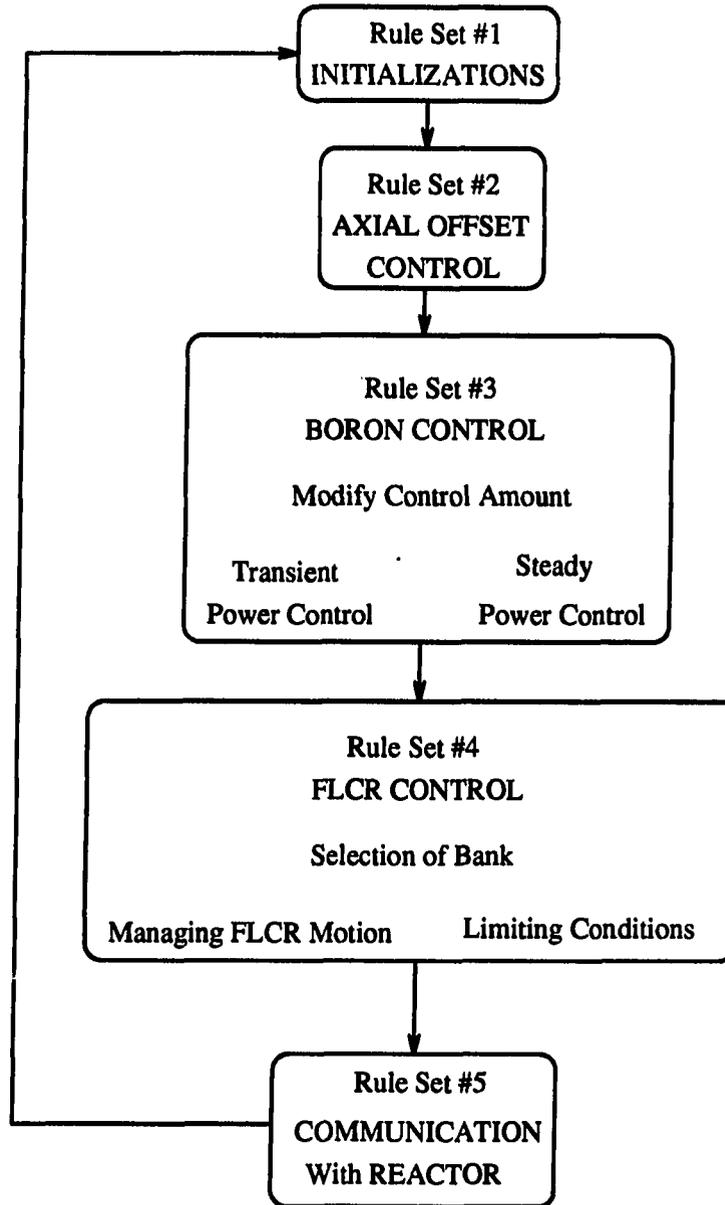


Figure 4.1: Production memory of ACES

Table 4.2: Modification of boron control amount in ACES

Power	AO	Response
increasing	> target AO	decrease one step
increasing	< target AO	increase one step
increasing	> control AO	set to zero
decreasing	< target AO	decrease one step
decreasing	> target AO	increase one step
decreasing	< control AO	set to zero
steady	not important	keep as it is

has two subgoals to accomplish. First, the amount of the boron change is modified if necessary. The criteria for modification of the control amount are the load demand, and the AO value. The strategy was explained previously, and is summarized here in Table 4.2. Notice that the amount of boron control is not determined by this method at steady power operation. The information given in Table 4.2 is translated into a set of generic rules. Instead of six distinct rule, and six more for long term transient cases when the xenon concentration behavior changes, only three generic rules are designed for increasing, decreasing, and setting the response to zero. A fourth rule changes the direction of response when the xenon behavior change is detected indirectly. Finally a fifth rule passes the control to the next step if modification is required.

A second group of rules in this set decides whether to use boron or not. During a power transient, the determined amount of boron change will be used. At steady state, however, the target reactivity value will be used to determine the control amount. The total of three rules complete this part of the ES, and transfer the control to the fourth set of rules.

The fourth set consists of power control rules. Selection of which FLCR bank

to use is the main concern of the power control strategy. Even though it is possible to use only one bank for power control, as was done in most of the previous studies [10, 36], we chose to use all FLCR banks, of course one at a time, for power control. As a result, we never have an FLCR bank inserted too much into the core, resulting in an excessive AO shift. A preference number is assigned to each bank based on their relative positions, -1 for lowest, $+1$ for highest positions, and 0 for the rest of the banks. ACES will use this number during the selection of which control rod bank to use. Therefore, the lowest position FLCR will be withdrawn and the highest position FLCR will be inserted when necessary, to keep the relative positions as close as possible.

If the absolute value of the local differential worth of the selected control rod bank is smaller than the target reactivity, then a temporary WME is created to manage the proper CR motions until the target reactivity is satisfied. There are some limiting conditions on FLCR motions, such as being totally withdrawn. These conditions are handled with separate rules for stopping to use that particular rod bank.

The last set of rules are the data communication rules of ACES. Currently calling a FORTRAN77 subroutine that simulates a PWR core, this rule is supposed to communicate with the reactor control console, receive the necessary set of data, and create a WME for the state of the core.

Support Routines Used in ACES

External functions and routines are generally used for detailed numerical calculations in OPS5. Although it is possible to write some rules that calculate the

absolute value or logarithm of a number for example, it is much more reasonable to do this kind of work by calling an external routine written in one of the procedural languages. We used FORTRAN77 for this purpose, and wrote routines that solve Eqns. 4.1 and 4.2, and that process some raw data.

The external function TREND returns 1 if the first argument is greater than the second, 0 if they are equal, and -1 otherwise. The function AOTREND, however, returns 1 if the first argument is greater than the second, 0 if it is in between the second and third arguments, and -1 otherwise. The function BORON performs integer division while the function ABSOLUTE returns the absolute value of its argument. The external functions that are part of ACES are given in App. B.

Finally, a PWR simulator is coupled with ACES as an external routine for validation purposes. ACES communicate with this routine to get the core parameters listed in Table 4.1, and send the corrective action to this routine. The simulator recalculates these core parameters for the next time step using the given control actions. Listings of support routines that are a part of ACES, are given in App. B, and routines that are part of the simulators are given in App. C.

Flow Control in ACES

There is no specific preference attributed to rules in OPS5 unless the programmer designs such a mechanism. Each rule that has a complete match with WMEs is included in the conflict set regardless of its position in the program or its context. Therefore, information flow should be provided by other means. First, strategy MEA is selected for the conflict resolution strategy to give an emphasis to the first WME that appear in the condition part. Then, a WME that contains the information about

the direction of data flow is used as the first WME in most of the rules. The attribute *goal* of this WME is set to various values which passes the control among the rule sets defined above.

For example, initializations are done in the absence of this WME. When the reactor data are sampled and processed, it is created with an attribute value that gives the control to the AO control set of rules. After one of the rules of this set is fired, the attribute will be modified to transfer the control to the next set. This process repeats at each set, and finally, the WME will be deleted from the active working memory when the control actions are sent to the reactor. By deleting this WME, ACES will be able to use initialization rules if necessary, and be ready for the next cycle starting with reactor data sampling.

After the control is passed to one of the sets, another flow control problem appears. Within a set of rules, there may be several possible paths for data flow. Usually, each rule is an individual path within a set. Flow control is managed by using the properties of the conflict resolution process. The recency rule of conflict resolution is used as the primary tool for flow control in this level. Creating a temporary WME may be used to disable a rule while it ables another one. This strategy is extensively used within the power control set of rules. The other tests of conflict resolution are used whenever necessary.

CHAPTER 5. SYSTEM MODELING

Developing the Model Equations

Modeling of the spatial xenon oscillations starts with the definition of the problem. For the constant power operation mode, several different methods were developed starting with a historical study by Randall and St. John [3]. A common approach among these studies is the two node reactor dynamics model. The point kinetic equations are written for two nodes of a reactor core where each region is subcritical by itself. Core criticality is satisfied by a diffusion term between the two regions. These equations are then supported by the xenon and iodine rate equations. A successful two point xenon oscillation model is developed by Onega and Kisner [51], and stability analysis of this approach is performed by Josephson [52].

For the load-follow mode, however, a more detailed approach is necessary for modeling of the reactor core. For this mode of operation, the spatial diffusion equation is coupled with the rate equations of the xenon and iodine concentrations for modeling. A three-dimensional core model with a multi-group approach is the most accurate solution that can be achievable within a reasonable computation time. On the other hand, the characteristics of the xenon oscillation control problem lead to a set of assumptions that can be used during the solution of the diffusion equation.

First, there may not be a need for that much of accuracy. While there are control

gains to be selected in most of the optimal control methods, it is unreasonable to model the core in that detail. Second, since the xenon oscillations are more likely to occur in the axial direction, the core model can be reduced to a one-dimensional study. Teachman and Omega showed that a one-group and a one-dimensional linear model is adequate for practical control purposes, and there is an inherent margin of safety because of the overprediction of the xenon effect in a one-energy group approach [53]. Consequently, we will use a one-dimensional one-group reactor core model for verification of the expert system control code ACES developed in this study.

Further details of the model are determined by the design of ACES. In order to comply with the characteristics of ACES, the reactor model should be able to calculate reactor power, instantaneous period and axial offset, and simulate discrete CR motions for different banks, discrete PLCR motions, boron poisoning, xenon feedback and temperature feedback. A one-dimensional solution of the diffusion equation is a reasonable approximation for some of these requirements, in which power and AO can easily be calculated by integrating the axial flux distribution. At the same time, the CR motions, boron poisoning, xenon feedback and the temperature feedback can be simulated by writing the absorption coefficient of the system in a more detailed form.

The diffusion equation is supported by the rate equations of the xenon and iodine concentrations for an accurate representation of the nonlinear xenon feedback effect. In addition, a one-group neutron precursor concentration is considered in the diffusion equation for representation of power transients. Therefore, another rate equation is used for the precursors in the model. Consequently, the general equations that govern the spatial xenon oscillations are written as,

$$\frac{1}{v} \frac{\partial \psi}{\partial t}(r, \theta, z, t) = \nabla \cdot D \nabla \psi + (1 - \beta) \nu \Sigma_f \psi + \lambda_C C - \Sigma_a \psi - \Sigma_c \psi - \sigma_B B \psi - \sigma_X X \psi - \delta \Sigma_a \psi \quad (5.1)$$

$$\frac{\partial I}{\partial t}(r, \theta, z, t) = \gamma_I \Sigma_f \psi - \lambda_I I \quad (5.2)$$

$$\frac{\partial X}{\partial t}(r, \theta, z, t) = \gamma_X \Sigma_f \psi + \lambda_I I - \lambda_X X - \sigma_X X \psi \quad (5.3)$$

$$\frac{\partial C}{\partial t}(r, \theta, z, t) = \beta \nu \Sigma_f \psi - \lambda_C C \quad (5.4)$$

where

v is the velocity of thermal neutrons,

ψ is the neutron flux,

t is the time,

D is the diffusion coefficient,

β is the delayed neutron fraction,

ν is the number of neutrons per fission,

Σ is the macroscopic cross-section of a material,

λ is the decay constant of an isotope,

σ is the microscopic cross-section of a material,

$\delta \Sigma_a$ is the absorption term due to the temperature feedback,

B is the boron concentration,

I is the iodine concentration,

X is the xenon concentration,

C is the one group neutron precursor concentration,

γ is the fission yield of a fission product,

B, X, I, C are subscripts for boron, xenon, iodine, and neutron precursors, and

f, a, c are subscripts for fission, absorption, and control rods.

For further analysis of the system, the following assumptions were used,

- prompt jump approximation,
- azimuthal symmetry for the flux distribution,
- first order Bessel function distribution along the radial direction, and
- constant properties along the core unless specified otherwise.

The finite difference numerical method is a very common technique used for solving the time dependent diffusion equation. The step size in the time domain is determined by the shortest time constant of the set of equations to be solved. This time constant is on the order of hours for the xenon and iodine rate equations where it is less than a second for the diffusion equation. Therefore, the system should be solved with a step size much smaller than a second, and considering the spatial dependency of the problem, the solution would take considerable computation time. By adopting the prompt jump approximation, the diffusion equation reduce to a steady state equation, and the minimum time constant of the system is determined by the precursor equation which is on the order of ten seconds. Consequently, solving the system with one second time steps will be enough for stability of the finite difference method, and will save computation time.

The second and third assumptions are used for reducing the system to a one- dimensional model. Based on these assumptions, the radial and azimuthal dependency of the variables will be separated from the axial and time dependency as follows,

$$\psi(r, \theta, z, t) = J_0(B_r r) \phi(z, t)$$

$$\begin{aligned}
\mathcal{I}(r, \theta, z, t) &= J_0(B, r)I(z, t) \\
\mathcal{X}(r, \theta, z, t) &= J_0(B, r)X(z, t) \\
\mathcal{C}(r, \theta, z, t) &= J_0(B, r)C(z, t)
\end{aligned}
\tag{5.5}$$

The last assumption additionally simplifies the system by neglecting spatial and time dependency of the material cross-sections. Using this assumption, the boron concentration will be written in the convenient form of ppm (part per million) units of coolant on a weight basis.

$$B = c_B B N_w \tag{5.6}$$

where

B is the boron concentration in ppm units

c_B is the conversion factor, and

N_w is the average coolant concentration in *molecules/cm³* units.

The conversion factor c_B carries out the unit conversion from ppm on a weight basis to an atom basis, and it is equal to $0.33161 * 10^{-6}$.

The last assumption will also be used for the derivation of the reactivity feedback term for the moderator temperature, which was written as a change in the absorption cross-section of the system in Eqn. 5.1. However, the moderator temperature reactivity feedback can be written in terms of known parameters by using first-order perturbation theory. The reactivity change as a result of a change in the absorption of the system is given in perturbation theory as [4],

$$\Delta\rho \cong -\frac{(\phi^+, \delta\Sigma_a\phi)}{(\phi^+, \nu\Sigma_f\phi)} \tag{5.7}$$

where ϕ^+ is the adjoint flux. The same reactivity change can be written in terms of

the reactivity feedback coefficient of temperature as,

$$\Delta\rho = \alpha_m(T(z, t) - T_r(z)) \quad (5.8)$$

where

$T(z, t)$ is the temperature distribution along the core, and

$T_r(z)$ is the reference temperature distribution.

Combining Eqns. 5.7 and 5.8, and using the fact that the flux is self-adjoint in the one-energy group approach,

$$\alpha_m(T(z, t) - T_r(z)) = -\frac{\int_V \delta\Sigma_a J_0^2(B, r) \phi^2(z, t) r dr d\theta dz}{\int_V \nu \Sigma_f J_0^2(B, r) \phi^2(z, t) r dr d\theta dz} \quad (5.9)$$

Considering the constant property assumption, the absorption term can be solved from the above equation.

$$\delta\Sigma_a = -\alpha_m \nu \Sigma_f (T(z, t) - T_r(z)) \quad (5.10)$$

The temperature terms in the above equation will be derived from the basic heat transfer equation;

$$\dot{m}c_p(T(z, t) - T_{in}) = \epsilon\Sigma_f \int_0^z \int_0^{2\pi} \int_0^R \psi(r, \theta, z', t) r dr d\theta dz' \quad (5.11)$$

where

ϵ is the energy conversion coefficient,

\dot{m} is the mass flow rate of the coolant,

c_p is the heat capacity of the coolant, and

T_{in} is the inlet temperature of the core.

Therefore, the temperature can be written as,

$$T(z, t) = \frac{\epsilon\Sigma_f}{\dot{m}c_p} \int_0^z \int_0^{2\pi} \int_0^R \psi(r, \theta, z', t) r dr d\theta dz' + T_{in} \quad (5.12)$$

We assumed that the almost all cross-sections are constant except the control rod absorption cross-section. The control rod absorption cross-section must preserve its axial and time dependency throughout the solution in order to provide a response for the corrective action proposed by the expert system. The following equation will be used together with a typical control rod distribution scheme along the core, given in Figure 5.1,

$$\begin{aligned} \Sigma_c(r, \theta, z, t) = & \Sigma_{c2} A_r \sum_j^M \left[\sum_i^{m_j} \frac{1}{r} \delta(r - r_{ij}) \delta(\theta - \theta_{ij}) \right] [H(z - z_i(t)) - H(z - z_i(t) - l)] \\ & \Sigma_{c1} A_r \sum_j^N \left[\sum_i^{n_j} \frac{1}{r} \delta(r - r_{ij}) \delta(\theta - \theta_{ij}) \right] H(z - z_j(t)) \end{aligned} \quad (5.13)$$

where

Σ_{c1} is the absorption cross-section of full-length rods,

Σ_{c2} is the absorption cross-section of part-length rods,

δ is the Kronecker delta function,

H is the Heaviside step function,

A_r is the effective area of the control rods,

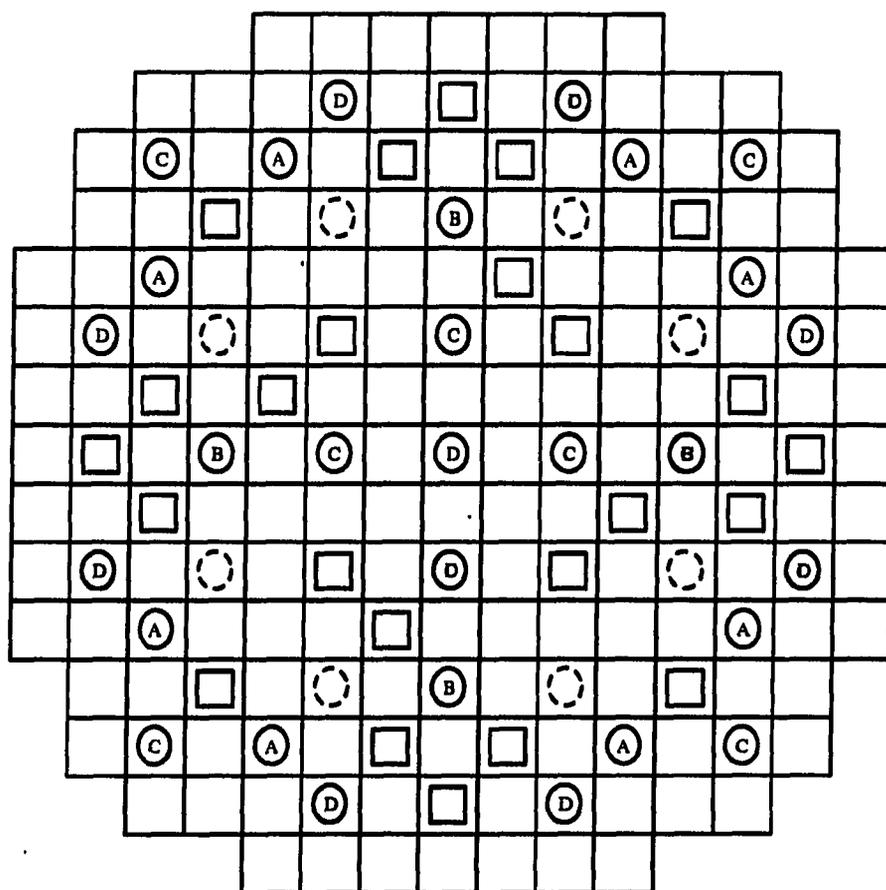
l is the length of the part-length control rods,

N, M are the number of full and part-length control rod banks, respectively, and

n_j, m_j are the number of clusters in each bank.

Introducing Eqns. 5.6, 5.10, and 5.13 into Eqns. 5.1 through 5.4, we will reduce the system of equations into one-dimensional form by using the radial and azimuthal average values given as,

$$\begin{aligned} \phi(z, t) &= \frac{1}{\pi R^2} \int_{2\pi} \int_R \psi(r, \theta, z, t) r dr d\theta \\ I(z, t) &= \frac{1}{\pi R^2} \int_{2\pi} \int_R \mathcal{I}(r, \theta, z, t) r dr d\theta \end{aligned} \quad (5.14)$$



- | | | | |
|---|-----------------|---|-------------------|
| Ⓐ | FLCR Bank A (8) | Ⓓ | FLCR Bank D (9) |
| Ⓑ | FLCR Bank B (4) | ⊖ | PLCR Bank (8) |
| Ⓒ | FLCR Bank C (8) | □ | Safety Banks (24) |

Figure 5.1: Typical control rod layout of a PWR core

$$X(z, t) = \frac{1}{\pi R^2} \int_{2\pi} \int_R \mathcal{X}(r, \theta, z, t) r dr d\theta$$

$$C(z, t) = \frac{1}{\pi R^2} \int_{2\pi} \int_R \mathcal{C}(r, \theta, z, t) r dr d\theta$$

and by using the following definition

$$\nabla \cdot D \nabla \psi(r, \theta, z, t) = D \frac{d^2 \phi(z, t)}{dz^2} - DB_r^2 \quad (5.15)$$

to consider the radial leakage term in the one-dimensional case.

Therefore, the final equations will be as follows,

$$-D \frac{d^2 \phi(z, t)}{dz^2} = [(1 - \beta) \nu \Sigma_f - \Sigma_a - DB_r^2 - c_B \sigma_B B N_w] \phi(z, t)$$

$$- \bar{\Sigma}_c(z, t) \phi(z, t) - \sigma_X X(z, t) \phi(z, t)$$

$$- \alpha_m \nu \Sigma_f [T(z, t) - T_r(z)] \phi(z, t) + \lambda_C C(z, t) \quad (5.16)$$

$$\frac{dI(z, t)}{dt} = \gamma_I \Sigma_f \phi(z, t) - \lambda_I I(z, t) \quad (5.17)$$

$$\frac{dX(z, t)}{dt} = \gamma_X \Sigma_f \phi(z, t) + \lambda_I I(z, t) - \lambda_X X(z, t) - \sigma_X X(z, t) \phi(z, t) \quad (5.18)$$

$$\frac{dC(z, t)}{dt} = \beta \nu \Sigma_f \phi(z, t) - \lambda_C C(z, t) \quad (5.19)$$

which are to be solved based on the boundary and initial conditions of,

$$\phi(0, t) = 0$$

$$\phi(H, t) = 0 \quad (5.20)$$

$$\phi(z, 0) = \phi_0$$

The rate equations also have similar initial conditions.

Because of the constant properties assumption, average cross-section values will be used for all materials other than the control rods. Considering the radial and

azimuthal dependency of the control rod absorption cross-section, a weighted average of Σ_c is calculated from,

$$\bar{\Sigma}_c(z, t) = \frac{\int_{2\pi} \int_R \Sigma_c(r, \theta, z, t) J_0(B, r) r dr d\theta}{\int_{2\pi} \int_R J_0(B, r) r dr d\theta} \quad (5.21)$$

which can be solved as,

$$\begin{aligned} \bar{\Sigma}_c(z, t) = & \frac{\Sigma_{c2} A_r}{2\pi R J_1(B, R)} \sum_j^M [H(z - z_j(t)) - H(z - z_j(t) - l)] \sum_i J_0(B, r_i) \\ & + \frac{\Sigma_{c1} A_r}{2\pi R J_1(B, R)} \sum_j^N H(z - z_j(t)) \sum_i J_0(B, r_i) \end{aligned} \quad (5.22)$$

For the numerical solution of Eqn. 5.22, we will approximate the radial positions of each cluster, the variable r_i , from the control rod layout of a typical PWR core given in Figure 5.1.

Solution of the System

There are several approximations for the solution of the diffusion equation given in Eqn. 5.16. Teachman and Onega solved the system by using the Galerkin's weighted residual method in which the flux is written as the multiplication of a time dependent coefficient and a shape function [53]. The main drawback of this approach is the necessity to lump all control rod banks into one. For the same reason, Chung limited his ES design for axial offset control to one control rod bank [47].

The importance of having independent control on different control rod bank positions led us to use finite difference methods in the spatial domain as well as the time domain of the solution. Additionally, nuclear reactors usually have discrete position indicators for control rod banks which enhances the validity of using the finite difference methods in the spatial domain.

The system of equations given in Eqns. 5.16 through 5.19 is a boundary value problem coupled with a set of initial value equations. Such a system requires the solution of the initial conditions to start the time-dependent solution of the system. Considering the nature of the problem and the system equations, the steady state solution of the system will be used as initial conditions of the problem. Therefore, the steady state case will be solved first.

Steady state solution

The steady state version of the system equations are

$$0 = D \frac{d^2 \phi(z, 0)}{dz^2} + \left[\frac{(1 - \beta)}{k_{eff}} \nu \Sigma_f - \Sigma_a - DB_r^2 - c_B \sigma_B B N_w \right] \phi(z, 0) - \bar{\Sigma}_c(z, 0) \phi(z, 0) - \bar{\sigma}_X X(z, 0) \phi(z, 0) + \lambda_C C(z, 0) \quad (5.23)$$

$$0 = \gamma_I \Sigma_f \phi(z, 0) - \lambda_I I(z, 0) \quad (5.24)$$

$$0 = \gamma_X \Sigma_f \phi(z, 0) + \lambda_I I(z, 0) - \lambda_X X(z, 0) - \bar{\sigma}_X X(z, 0) \phi(z, 0) \quad (5.25)$$

$$0 = \frac{\beta \nu \Sigma_f}{k_{eff}} \phi(z, 0) - \lambda_C C(z, 0) \quad (5.26)$$

where k_{eff} is the effective multiplication factor of the reactor. Since the reference temperature distribution is defined as the temperature distribution at full power steady state, the reactivity feedback term does not appear in the steady state equations.

The spatial derivative term will be replaced with the finite difference equations by using the following equation based on the Taylor expansion,

$$\frac{d^2 \phi}{dz^2} = \frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{h_z^2} + \mathcal{O}(h_z^2) \quad (5.27)$$

where h_z is the spatial step size. The term $\mathcal{O}(h_z^2)$ means that the truncation error is of order h_z^2 [54].

Additionally, the xenon concentration which appears in Eqn. 5.23 can be solved in terms of flux by using Eqns. 5.24 and 5.25 as,

$$X(z, 0) = \frac{(\gamma_I + \gamma_X)\Sigma_f}{\lambda_X + \sigma_X\phi(z, 0)}\phi(z, 0) \quad (5.28)$$

Similarly, the precursor concentration can be solved from Eqn. 5.26 as,

$$C(z, 0) = \frac{\beta\nu\Sigma_f}{k_{eff}\lambda_C}\phi(z, t) \quad (5.29)$$

which cancels the effect of delayed neutrons from the steady state equations. Therefore, the system reduces to,

$$D\frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{h_z} + \left[\frac{\nu\Sigma_f}{k_{eff}} - \Sigma_a - DB_r^2 - c_B\sigma_B BN_w\right]\phi_i - \bar{\Sigma}_{ci}\phi_i - \sigma_X\frac{(\gamma_I + \gamma_X)\Sigma_f}{\lambda_x + \sigma_X\phi_i}\phi_i^2 = 0 \quad (5.30)$$

with

$$\phi_0 = 0$$

$$\phi_N = 0$$

where $i = 1, \dots, N$ is the mesh points along the z-direction. In order to achieve reasonable accuracy in the spatial domain, the solution mesh size must be consistent with the minimum step size of the control rod position indicator. Therefore, the number of mesh points depends on the minimum step size of the control rod movements, and total height of the core. Usually, full scale of an ordinary control rod position indicator varies between 150-250 in nuclear reactors [10]. Based on this information, using approximately 200 mesh points will be adequate to model control rod motions in the axial direction.

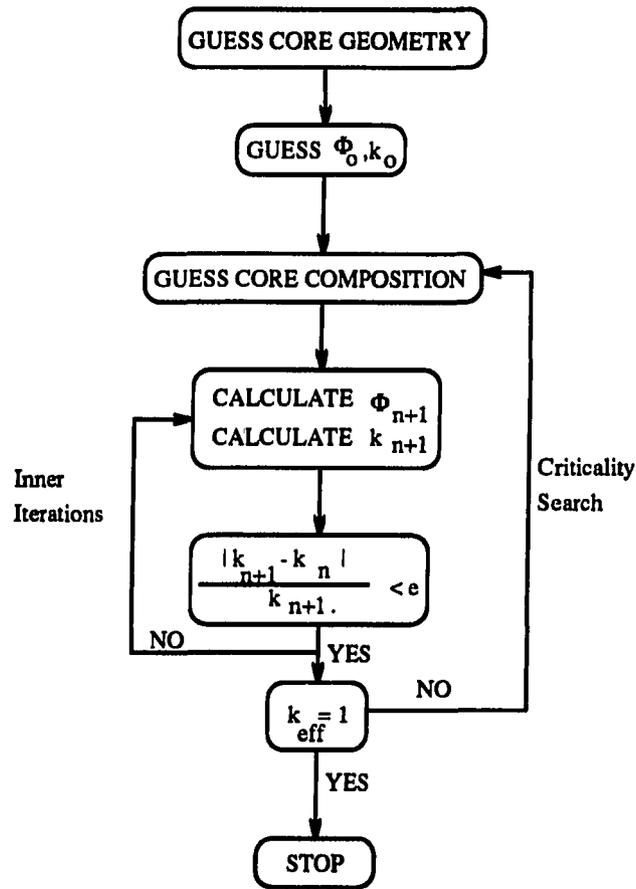


Figure 5.2: Iteration Flowchart for Power Method

The diffusion equation given in Eqn. 5.30 is a non-linear eigenvalue problem. There is not a general analytical solution for a nonlinear eigenvalue problem that gives all eigenvalues and corresponding eigenfunctions. On the other hand, it is known that a system is dominated by the fundamental eigenvalue at steady state, and the spatial distribution assumes the shape of the fundamental eigenfunction [54]. Therefore, it will be enough to solve for the fundamental eigenvalue of the system. Duderstadt and Ames suggest using the iterative power method in such a case [1, 54].

The particular application of the power method to the diffusion equation includes two nested iteration loops (See Figure 5.2). The inner iteration solves for the effective multiplication factor, the eigenvalue of the system, based on an initial guess of the flux distribution. After the effective multiplication factor is solved, an outer iteration will be performed on the boron concentration to achieve criticality, ie. $k_{eff} = 1$. The shape of the flux distribution will be determined at this stage according to the power method [1]. However, it does not contain any information about the magnitude of the flux, and simple scaling with a given power is not a solution because of the system nonlinearities.

Therefore, the power iteration procedure will be repeated after rescaling of the flux for a given reactor power. When the flux is converged also for power, the solution will contain all information necessary for the rest of the steady state solution. Having solved the flux distribution along the core at steady state, iodine, xenon, and precursor concentrations can be solved by using Eqns. 5.24, 5.25 and 5.26, respectively. A FORTRAN77 program, STEADY.FOR was written to implement the given solution, and a copy of which is provided in App. D.

Time-dependent solution

It is already determined that the finite difference equations will be used in the time domain. The first-order difference equation for a first-order derivative can be written as,

$$\frac{du}{dt} = \frac{u_{n+1} - u_n}{h_t} + \mathcal{O}(h_t) \quad (5.31)$$

where h_t is the time step size. Applying the finite difference equations given in Eqns. 5.27 and 5.31 to Eqns. 5.16 through 5.19, the time-dependent equations can

be obtained as,

$$\begin{aligned}
D \frac{\phi_{i-1,n} - 2\phi_{i,n} + \phi_{i+1,n}}{h_z} + \left[\frac{(1-\beta)}{k_{eff}} \nu \Sigma_f - \Sigma_a - DB_r^2 \right. \\
\left. - c_B \sigma_B BN_w \right] \phi_{i,n} - \alpha_m \nu \Sigma_f [T_{i,n} - T_{ri}] \phi_{i,n} \\
- \bar{\Sigma}_{ci,n} \phi_{i,n} - \sigma_X X_{i,n} \phi_{i,n} + \lambda_C C_{i,n} = 0 \quad (5.32)
\end{aligned}$$

and,

$$I_{i,n+1} = h_t [\gamma_I \Sigma_f \phi_{i,n} - \lambda_I I_{i,n}] + I_{i,n} \quad (5.33)$$

$$X_{i,n+1} = h_t [\gamma_X \Sigma_f \phi_{i,n} + \lambda_I I_{i,n} - \lambda_X X_{i,n} - \sigma_X X_{i,n} \phi_{i,n}] + X_{i,n} \quad (5.34)$$

$$C_{i,n+1} = h_t \left[\frac{\beta \nu \Sigma_f}{k_{eff}} \phi_{i,n} - \lambda_C C_{i,n} \right] + C_{i,n} \quad (5.35)$$

where i , and n subscripts for the axial and time steps, respectively. The effective multiplication factor will be determined during the steady state solution of the problem and kept constant in the time domain solution.

Having the solution of variables at time step n , Eqns. 5.33 to 5.35 can be solved for the next time step. Then, Eqn. 5.32 will be solved for the flux at time step $n + 1$. However, the xenon concentration and the temperature terms in this equation are also time dependent, and creates nonlinearities. The finite difference method will also help in handling of these nonlinearities of the system. We used Picard linearization for nonlinear terms, which is simply using the temperature and xenon concentration solved at time step n while solving the flux at time step $n + 1$ [54].

Solving these equations requires determination of the time step based on the time constants of the system equations. The system of rate equations given in Eqns. from 5.33 to 5.35 have two dominant time constants. The time constant of the precursor equation limits the step size of the time domain to seconds for a stable

finite difference solution, while the xenon and iodine rate equations can be solved with much larger step size, such as minutes. Considering the increased size of the problem due to the application of the finite difference method in the spatial domain with 200 mesh points, two different step sizes are used in the time domain to reduce the computation time. Eqn. 5.32 is solved every second together with the precursor equation (Eqn. 5.35) to satisfy the stability conditions of the finite difference solution of the precursor equation, and the xenon and iodine concentrations are updated every minute by solving Eqns. 5.33 and 5.34 with current flux values.

Consequently, both stability conditions are met with a gain in computation time. Since in this study we wish to have the reactor simulator run for at least 24 hours of simulated reactor time, it is quite important to keep the CPU time as small as possible for flexibility of the debugging process. On the other hand, the important computation time is the CPU time consumed by the Expert System during the decision process rather than the CPU time consumed by the reactor model unless the model is slower than real time. The time-dependent solution of the system is implemented as a FORTRAN77 subroutine for the expert system, and the source code of REACTOR.FOR is listed in App. C.

The communication between the subroutine REACTOR and ACES is provided by a set of built-in functions which converts OPS5 variables into floating point or integer numbers or vice versa. The control parameters are received from ACES through these functions. Then, the system of equations are solved for every second up to a minute. The flux distribution is numerically integrated to solve for the power and axial offset of the system. Since the average flux value is used during the calculations, the following trapezoidal numerical integration equation is used for

calculation of global or local power,

$$P = \epsilon \Sigma_f \pi R^2 h_z \left[\frac{\phi_j + \phi_N}{2} + \sum_{i=j+1}^{N-1} \phi_i \right] \quad (5.36)$$

by proper setting of the upper and lower limits j and N . In Eqn. 5.36, the mesh size is assumed to be the same for all mesh points.

The instantaneous period of the reactor is calculated by using the instantaneous power at two successive time steps, which is also a finite difference solution of Eqn. 4.2 given in the previous Chapter.

Generation of Constants Used in ACES

The ES developed in this study requires a knowledge base of reactivity worth values for control rods and boron concentration change as discussed previously. These values are strongly reactor dependent parameters, and they are usually determined by a series of experiments for each reactor. These experiments measure the power change as a result of a rod drop, and uses inverse kinetic equations to derive the integral rod worth curves for each bank.

We will use a similar approach to calculate the total reactivity worth of a control rod bank, based on the basic definition of reactivity given in the following equation,

$$\rho = \frac{k_2 - k_1}{k_2} \quad (5.37)$$

With the steady state solution of the system, we already have $k_1 = 1$. Assuming the flux will immediately change its shape after a rod motion, we will solve the steady state equations for each rod bank twice. First, we will withdraw one of the banks totally out of the core, and solve the eigenvalue problem without the criticality search.

The new effective multiplication factor of the system gives the reactivity worth of the corresponding control rod motion. Then, we will insert the same bank totally, and repeat the solution. The total reactivity worth of a control rod bank is then calculated by adding the two reactivity worth values.

At this point, a very common approach is to assume that the control rod worth is proportional to the square of undisturbed flux at a certain point [55]. A FORTRAN77 program, WORTH.FOR was written to generate the differential reactivity curves of all FLCR banks. The results of the program is written into a file which is read by ACES in free format.

The reactivity worth of a boron concentration change can be calculated by repeating the first part of this routine since the boron change is a global reactivity insertion. Additionally, the reactivity worth of a boron concentration change increases linearly as the amount increases. Therefore, only one solution will be enough to calculate the reactivity worth of boron. For practical purposes, however, the differential rod worth curves and boron reactivity worth are solved by two separate programs. The source codes of these programs are also provided in App. E.

CHAPTER 6. VERIFICATION OF ACES

The Results of the Reactor Model

A PWR core model is developed for the verification of the rules of ACES. Here, we introduce the results of the model first, and compare them with previous studies. The constants used in the solution of the PWR core are based on a typical B&W core [1], and listed in Table 6.1. The macroscopic cross-sections are all generated from microscopic cross-sections by necessary corrections given by Lamarsh [4]. Since we assumed a homogeneous reactor for the unit volume of each mesh points, cross-sections are smeared out over the cross-sectional area of the core. The constants related to the control rod cross-sections are adopted from a previous study of xenon oscillation control [36].

Among the given constants, the boron concentration is determined by the solution of the steady state equations, and it is a typical number for a core at the beginning of its life [4]. Since we didn't consider any fuel burnup during the calculation of the fission cross-section of the fuel, the critical boron concentration shows the reliability of the solution. Additionally, the steady state flux distribution (Figure 6.1) at full power gives an AO value of -4.65% which is also a very typical AO value of a PWR core [10].

During the solution of the steady state equations, the control rod cross-section

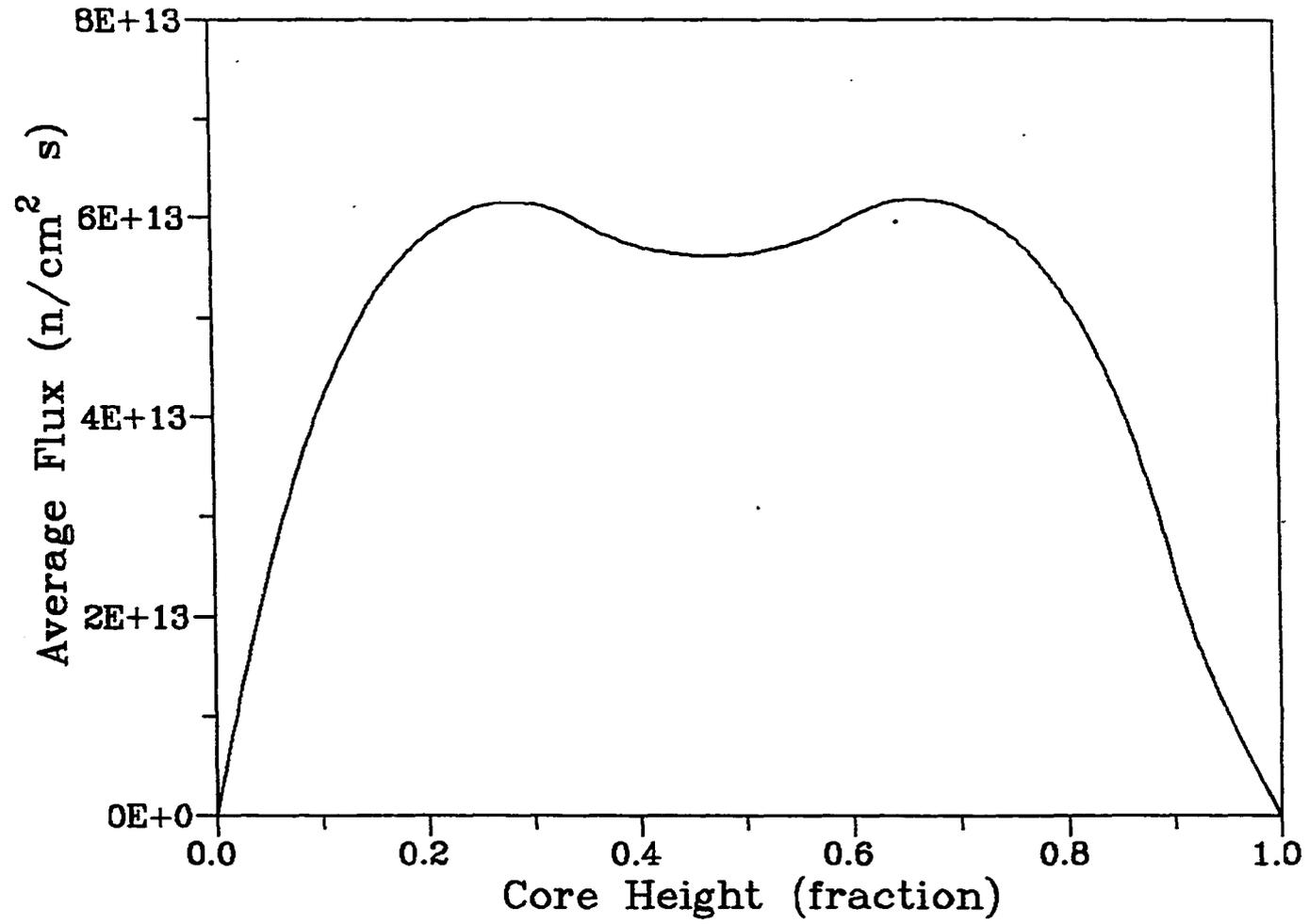


Figure 6.1: Full power steady state flux distribution

Table 6.1: Reactor constants

Thermal power	3400 MW_{th}
Active core height	370 cm
Active core radius	170 cm
Mass flow rate	15944 kg/s
Specific heat of water	6060 W/kgK
ν	2.418
Σ_f	0.06617 cm^{-1}
Σ_a	0.1285 cm^{-1}
D	1.2 cm
Σ_c	1.8636 cm^{-1}
σ_B	3.838*10 ⁻²¹ cm^2
σ_X	1.232*10 ⁻¹⁸ cm^2
λ_X	0.20917*10 ⁻⁴ s^{-1}
γ_X	0.00228
λ_I	0.2875*10 ⁻⁴ s^{-1}
γ_I	0.06386
λ_C	0.0767 s^{-1}
β	0.0065
α_m	1.0*10 ⁻⁴ $\Delta\rho/\rho K$
FLCR positions	180 steps (90 %)
PLCR position	70 steps (35 %)
Boron concentration	915.95 ppm

given in Table 6.1 was used for FLCRs. PLCRs are assumed as long as a quarter of the FLCRs, and therefore, the macroscopic absorption cross-section for the PLCR is one-fourth of the value given in that Table. In order to find the control absorption at any mesh point, Eqn. 5.22 is solved for approximate positions of each cluster shown in Figure 5.1 in radial and azimuthal directions. Then, CRWORTH.FOR is used for calculating the differential worth of each bank. The total reactivity worth values of each FLCR bank are tabulated in Table 6.2. They are in accordance with the values given in the literature [56]. This also shows the validity of the finite difference solution approach.

Table 6.2: Total reactivity worths of FLCR banks

Bank	$\Delta\rho/\rho$	Dollars
FLCR-A	$4.479 * 10^{-3}$	0.689
FLCR-B	$3.833 * 10^{-3}$	0.59
FLCR-C	$6.346 * 10^{-3}$	0.976
FLCR-D	$5.353 * 10^{-3}$	0.824

Verification of ACES

The verification of an expert system involves two different sets of tests. Each rule in an expert system should be tested for the proper performance for which it was designed. This process, however, may require fictitious data, and corresponds to the debugging process of a program written in a conventional language. As one of the goals of this study, the expert system is designed by using generic rules, which reduced the number of rules, and therefore, the time spent for debugging of the program.

The second test is related to the overall performance of the expert system, and requires a set of test cases which should be designed carefully. The test cases were designed with an intention of uncovering any deficiency of the ES design. The test cases, listed as a daily power load of the reactor in percents of full power in Table 6.3, are designed based on the following criteria:

- Considering the daily cyclic behavior of a power load, each case should be run for at least 24 hours.
- The time dependent solution of the model should be verified.
- ACES should be able to keep the power and AO under control at different power levels for time periods long enough to experience the xenon oscillations.

- It should be able to handle different rate of power level changes.
- And finally, it should be able to follow an arbitrary power load demand for at least 24 hours period.

Table 6.3: Test cases for verification of ACES

Time (hours)	Case A	Case B	Case C	Case D	Case E
0	100	100	100	100	100
1	100	100	100	100	100
2	100	100	90	75	90
3	90	100	80	50	80
4	80	100	70	50	75
5	80	100	60	50	70
6	80	90	50	50	65
7	80	80	50	50	60
8	80	80	50	50	55
9	80	80	50	50	50
10	80	80	50	50	60
11	80	80	50	50	70
12	80	80	50	75	80
13	80	80	60	75	90
14	80	80	70	75	100
15	80	80	80	75	100
16	80	90	90	100	100
17	80	100	100	100	100
18	80	100	100	100	100
19	80	100	100	100	100
20	80	100	100	100	100
21	80	100	100	100	100
22	80	100	100	100	100
23	80	100	100	100	100
24	80	100	100	100	100

Case A is designed to observe the axial xenon oscillations more clearly. In a load-follow mode, two separate xenon transients occur at the same time. The first transient is the change of the xenon concentration because of a change in the power

level. The xenon concentration increases for instance, following a decrease in power, and settles to a new steady state concentration value. This transient will simply be called the xenon feedback during the following discussions. The second transient is the xenon oscillation that occurs due to uneven reactivity insertion in the core. At constant power, however, the xenon oscillation will be the only transient after the xenon concentration changes to the new steady state value.

To study the xenon oscillation separately, the reactor power is reduced to 80 % of full power, and kept at this level for 22 hours (Figure 6.2), which is enough time to observe the xenon oscillations. To reduce the power, FLCR banks are inserted into the core as much as 4-5 steps (Figures 6.4 and 6.5) which corresponds to 2-2.5 % of total core length. ACES uses its power control rules (Rule set # 4) to control the power, while it uses the boron control rules (Rule set # 3) for the xenon feedback. As seen in Figure 6.7, the boron concentration is changed with an increasing rate by using rules 3.1 for increasing the response, and 3.7 for using the boron for xenon feedback during a transient.

The insertion of control rods was enough for stimulation of a xenon oscillation as seen in Figure 6.3. During the steady state operation at 80 % of full power, the xenon concentration continued to increase because of the xenon feedback. This transient was recognized by ACES from the change of system reactivity, and a boron concentration change was proposed as the corrective action (Figure 6.7) by using rule 3.8 given in App. A. The smoothness of the boron concentration change in close proximity to the 4th hour, the time that the power transient ends, shows that separate boron control rules for steady state and transient power operations are in agreement with each other in terms of the amount of boron control.

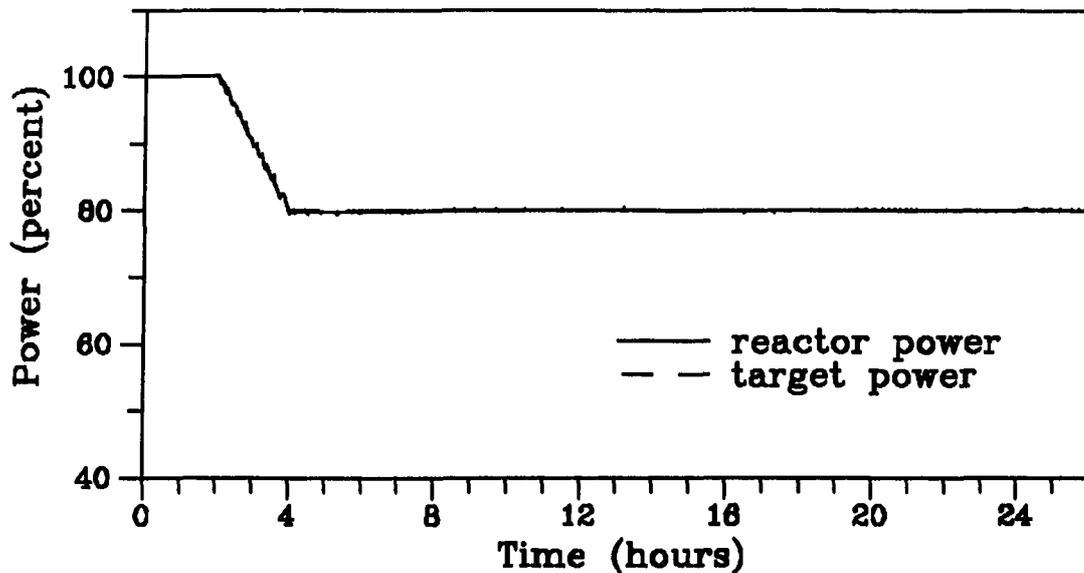


Figure 6.2: Case A: Reactor power and load demand

The xenon oscillation was observed in this case, following the xenon feedback, and recognized by ACES from two separate points. The AO shift was one of the signals, and PLCRs were used (Rule # 2.2) to compensate for this shift as shown in Figure 6.6. As seen in Figure 6.3, ACES kept the target AO power dependent during this case. The AO converges to zero from its steady state value at full power while the power decreases [10]. Hence, ACES follows the power changes, and determines the new target AO, and the target and control bands around it at each time step in the rule 1.5. The other signal for the recognition of the xenon oscillation was the net reactivity effect of the oscillation, and as it can be seen from Figure 6.7, the rate of boration was increased to compensate for this effect. At the end, the AO was kept well within the target band and the goal was achieved for test case A (Figure 6.3).

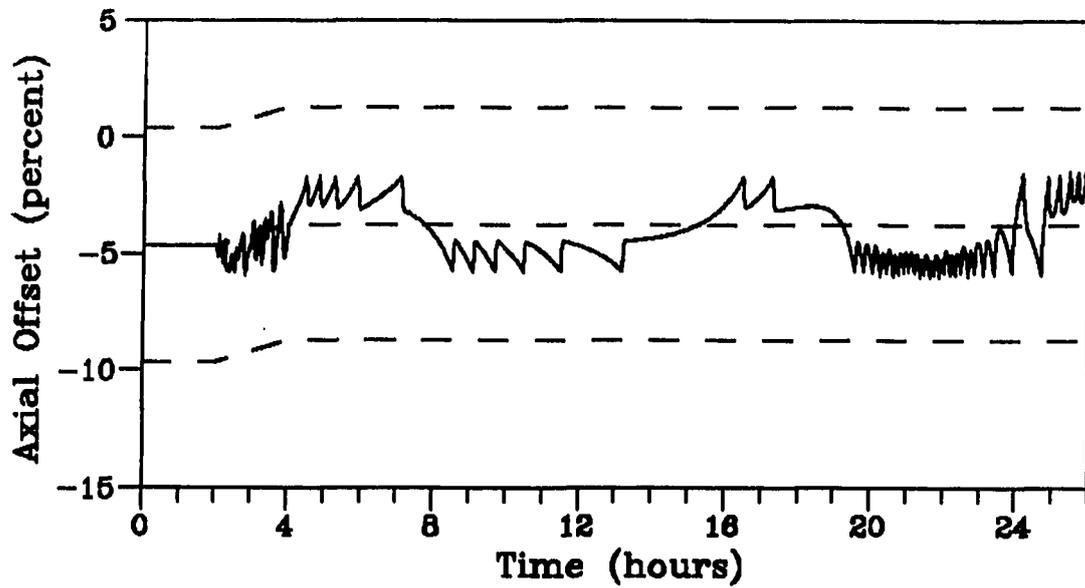


Figure 6.3: Case A: Axial offset change

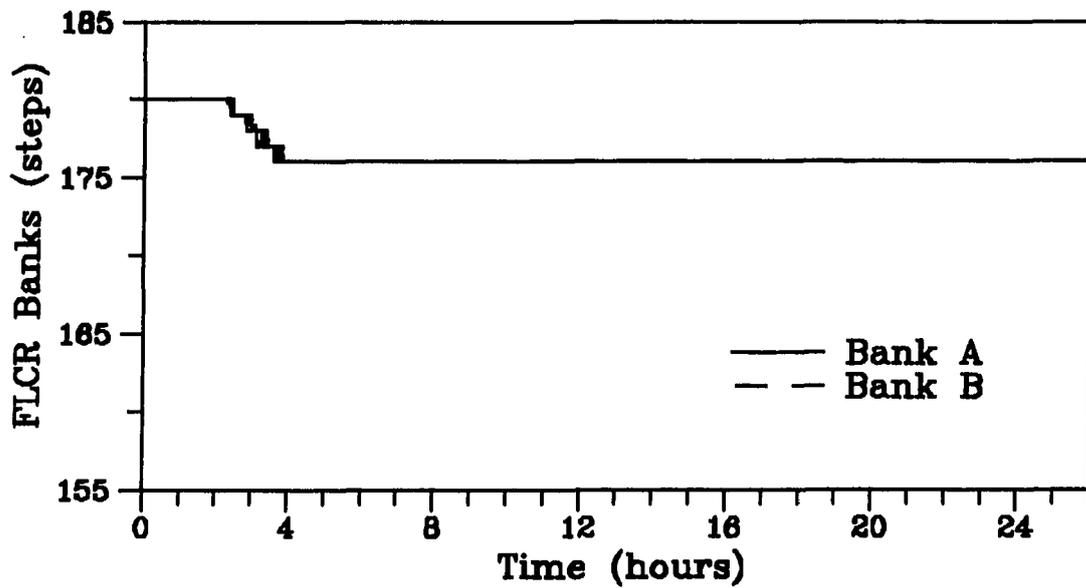


Figure 6.4: Case A: Full length control rod positions-1

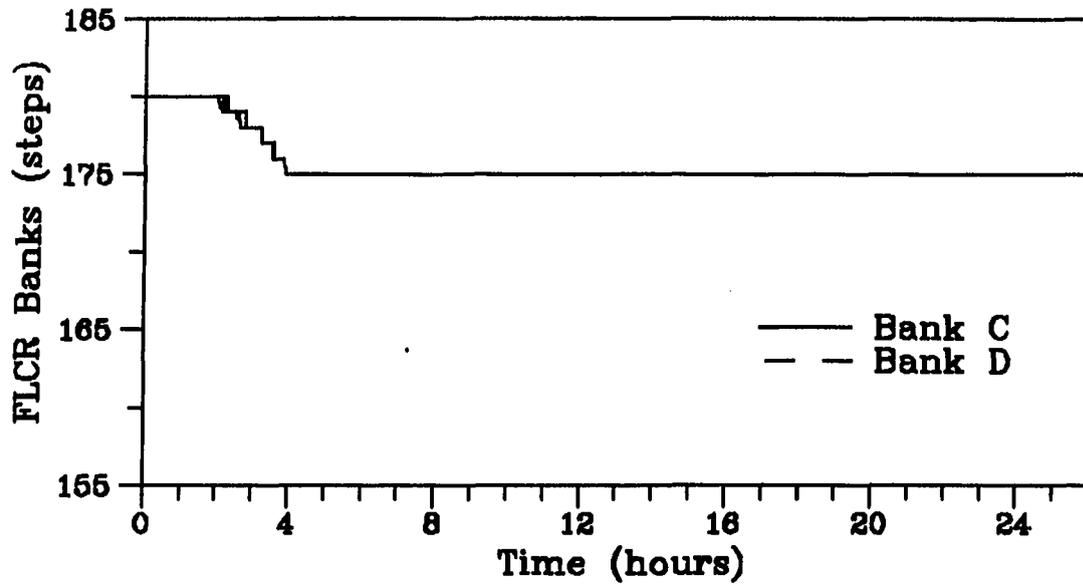


Figure 6.5: Case A: Full length control rod positions-2

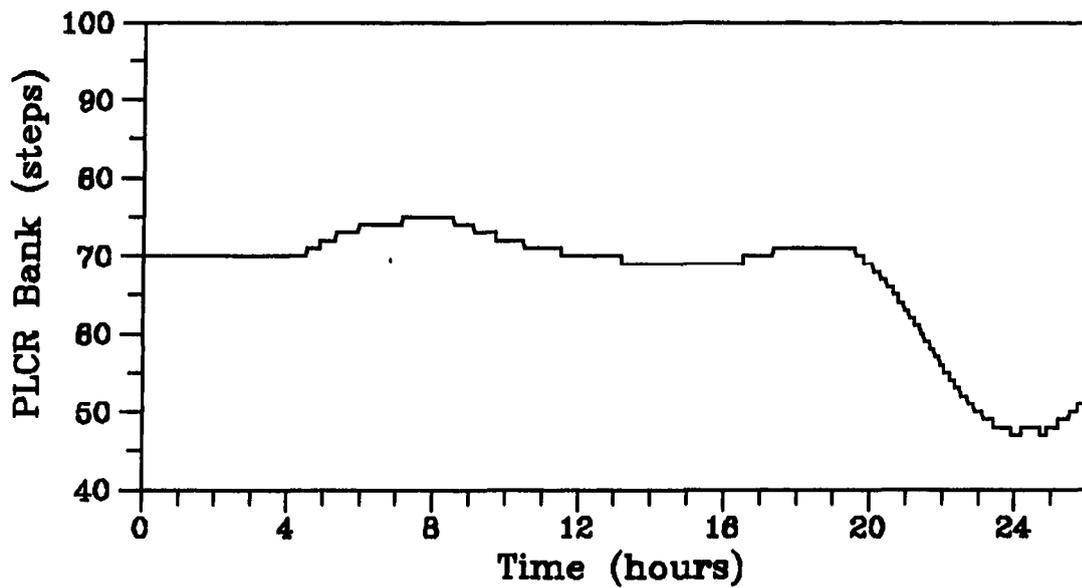


Figure 6.6: Case A: Part length control rod position

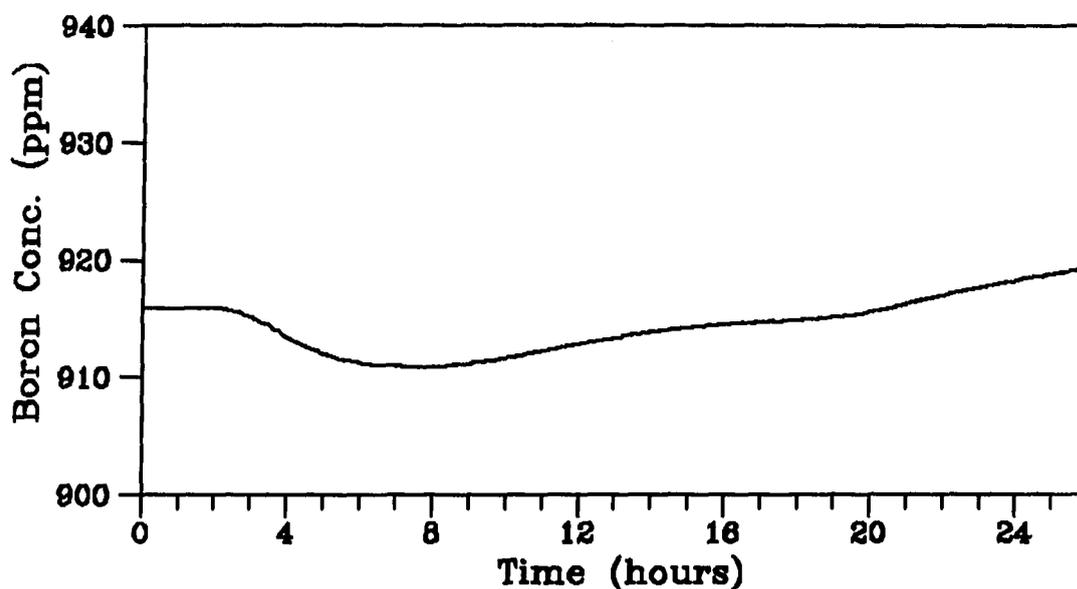


Figure 6.7: Case A: Boron concentration change

In order to explain both xenon transients in detail, the change in xenon distribution is given at every four hours in Figures 6.8 and 6.9. The global increase in xenon concentration right after the power transient can easily be seen in these figures. At the end of the power transient, the xenon concentration is just increased in magnitude, preserving the distribution the same as seen from the first and the second curves in Figure 6.8. It peaks four hours after the transient, but continues to change. The first effects of a xenon oscillation can be seen in the xenon distribution given at the 8th hour. The xenon concentration peak in the bottom-half was higher than the peak at the top-half of the core, and therefore, the axial offset was shifting to the top. This shift is sensed by ACES and kept under control by pulling the PLCRs to the center of the core as can be seen in Figure 6.6. The flux distribution during this

time clearly shows the shift to the top of the core(Figure 6.10).

The flux distribution change also shows the tendency of a phenomenon known as the *power pinching* effect. As xenon peaks at both ends of the core(Figure 6.8), it depresses the flux in these regions, and forces the flux to peak at the center. However, as the xenon transient continues, the flux distribution resumes a distribution close to its steady state shape (Figure 6.11) around the 18th hour.

Later during the transient, the xenon oscillation becomes more effective. The xenon concentration changes to shift the AO down. The magnitude of the oscillation is much stronger than the xenon feedback as can be seen from the corresponding figures of the flux distribution, the xenon distribution, the boron concentration, and finally the PLCR position. The oscillation reaches its peak around the 26th hour. The response of ACES to the oscillation changes its direction, i.e., it pulls the PLCRs to the top of the core (Figure 6.6) after they are inserted to keep the AO under control successfully as seen in Figure 6.3.

Case A is unique in several ways. It is a unique test case that was extended to 26 hours to observe the xenon oscillation. It is also a unique test case in which the power was kept constant for the same reason. The flux and xenon distributions are given only for this case since they are not measurable parameters in a real case study. Therefore, the response of ACES will be explained based on the observable variables in the other test cases.

Case B is designed to test two expected behaviors . First, the steady state behavior of the model should be tested for its time-dependent response. It has been previously shown that the steady state solution of the model gave reasonable results in terms of the flux distribution and the axial offset of the core. If the system is

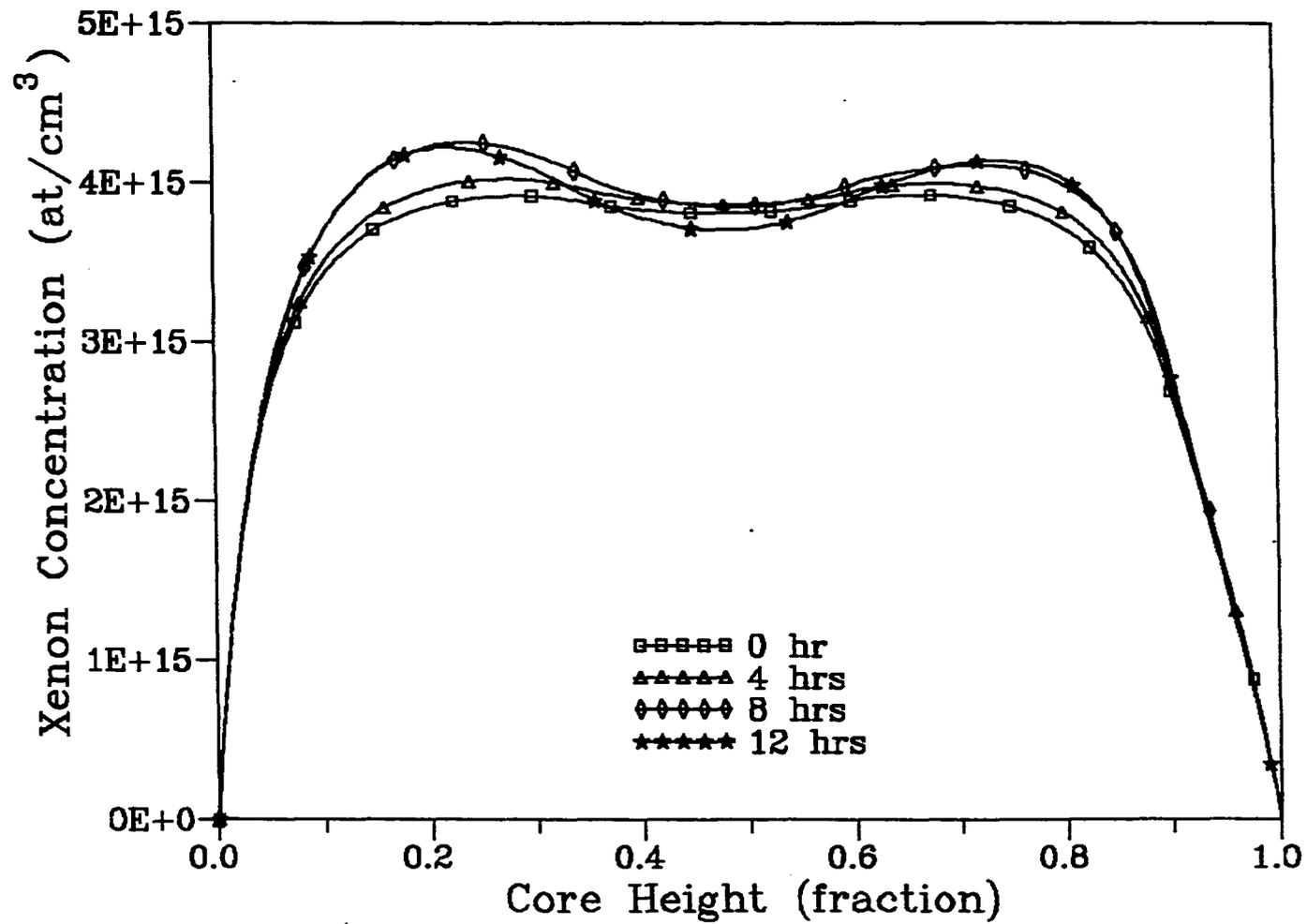


Figure 6.8: Case A: The xenon distribution change-1

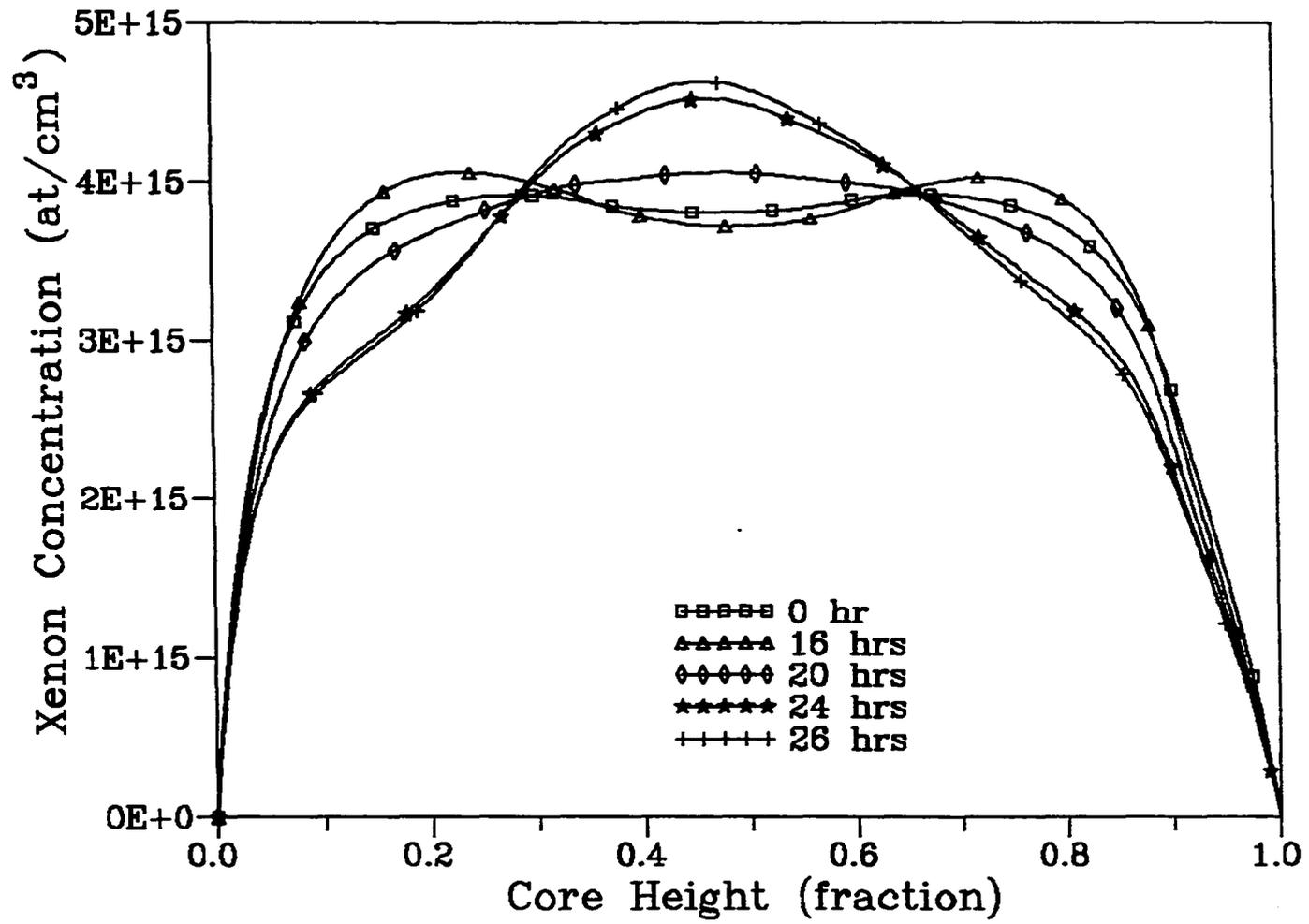


Figure 6.9: Case A: The xenon distribution change-2

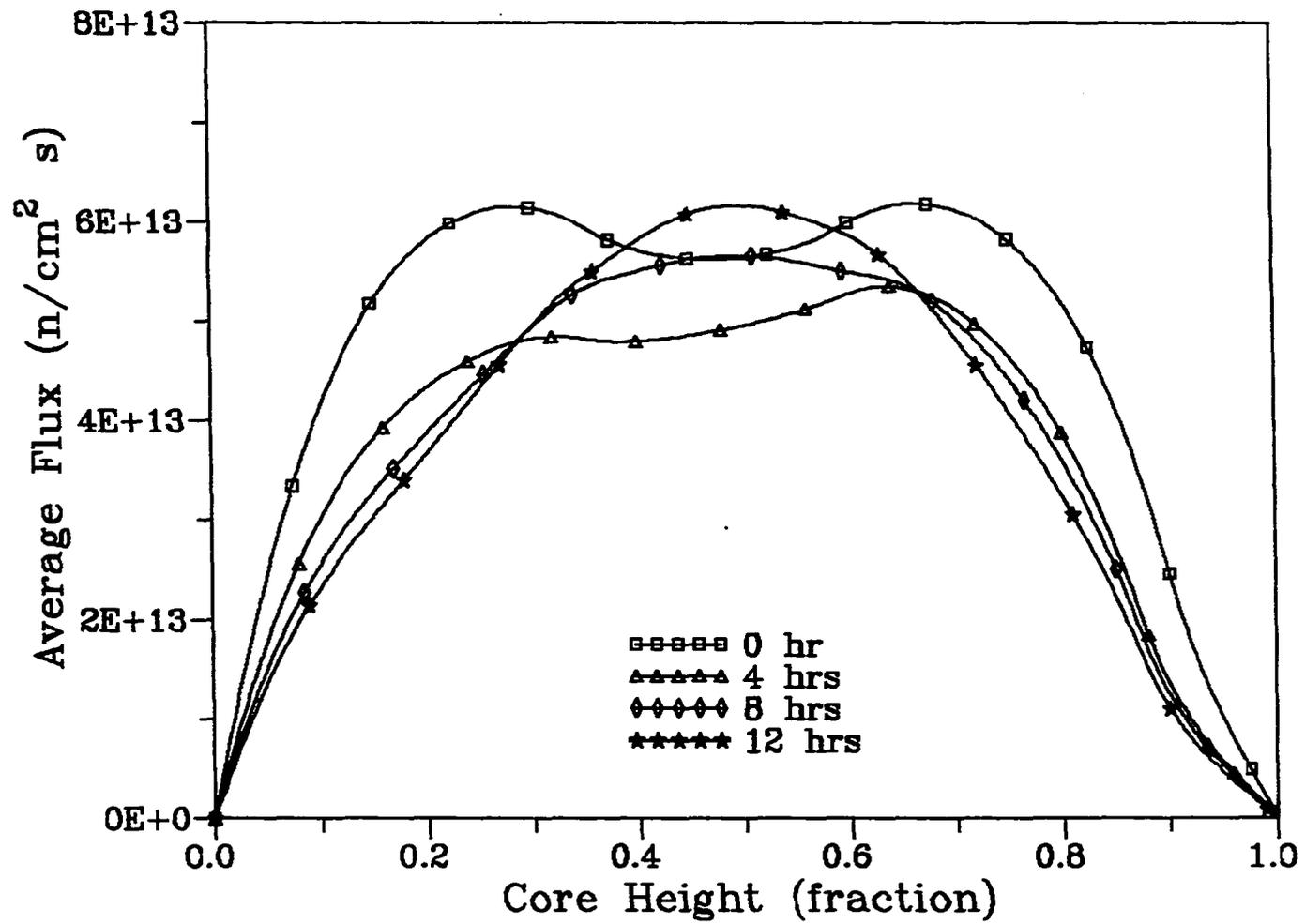


Figure 6.10: Case A: The flux distribution change-1

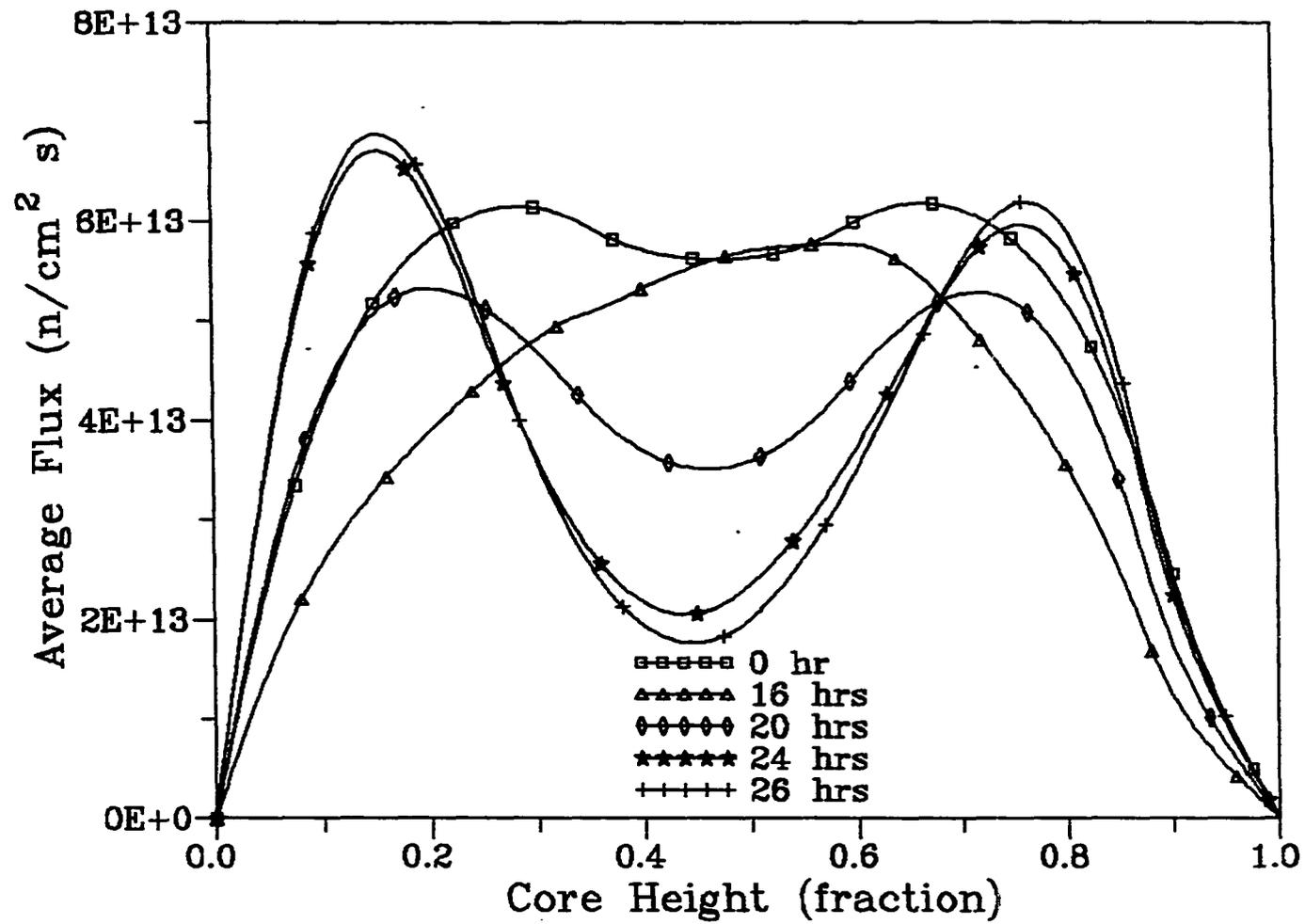


Figure 6.11: Case A: The flux distribution change-2

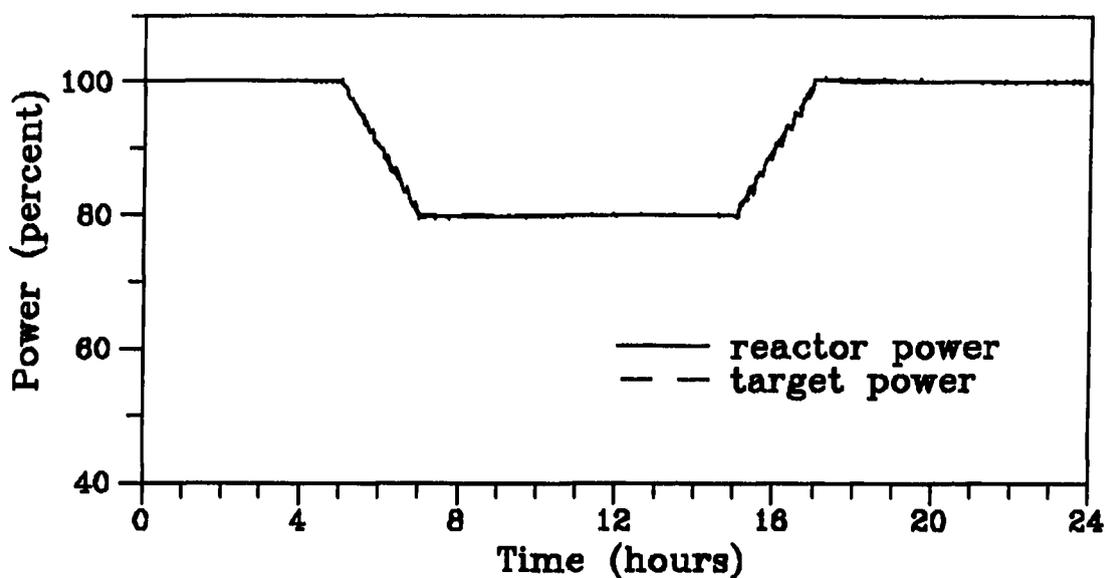


Figure 6.12: Case B: Reactor power and load demand

really at steady state, it should stay there for a long time period without any control action. Second, after observing that the ACES was able to handle a 10 % per hour power decrease rate in case A, it should be able to restore the power back to 100 %.

This case required ACES to keep power at steady state for 5 hours. The model stayed at steady state without any need for control action for more than 4 hours. The power oscillations around the steady state were under 0.1 % of full power during this period, which can hardly be seen in Figure 6.12. Therefore, it is reasonable to say that the core model is accurate enough, and the reference core is at steady state for our purposes.

ACES successfully followed the 10 % per hour rate of power decrease for two hours, kept the power at 80 % for eight hours. ACES was already tested up to this

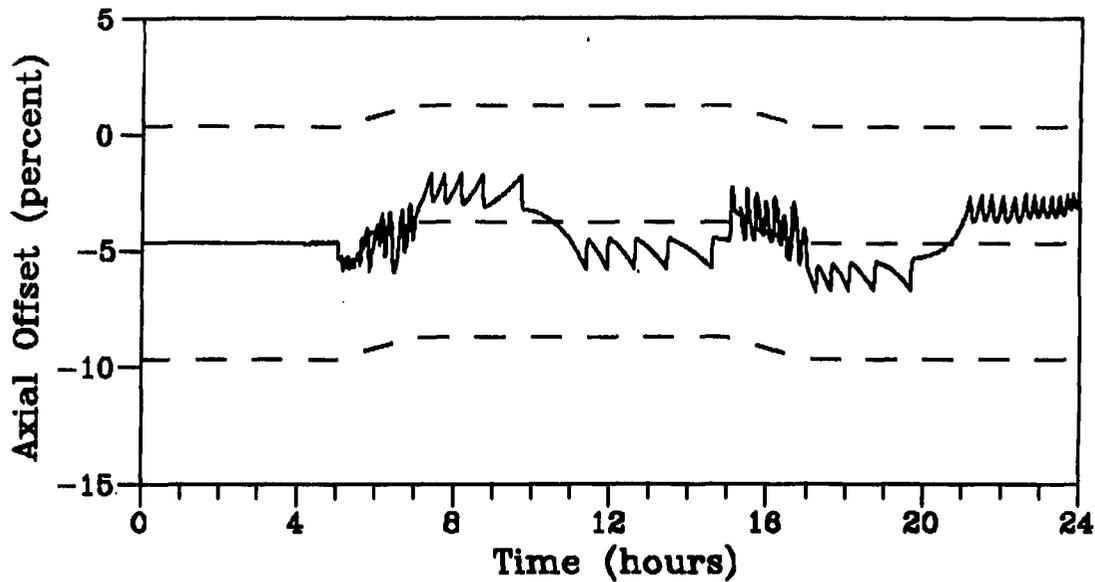


Figure 6.13: Case B: Axial offset change

point in case A, and it was successful in keeping the power and the AO under control. However, there is one characteristic of ACES that clearly reveals itself here. Even though cases A and B are numerically identical up to this point, the FLCR positions are not the same at 80 % power as can be seen in Figures 6.4, 6.5, 6.14, and 6.15. It shows that the response of ACES is not limited by some magic numbers related to control gains, and it is not limited by fixed rules based on some pattern recognition procedures. In either case, ACES would respond the same, and control rods would be at the same position. Because of the generic characteristic of rule set # 4, ACES was able to respond differently even for very similar cases.

After eight hours of 80 % power operation, the reactor power was increased to 100 % with the same rate as used for the power decrease. It can be seen from

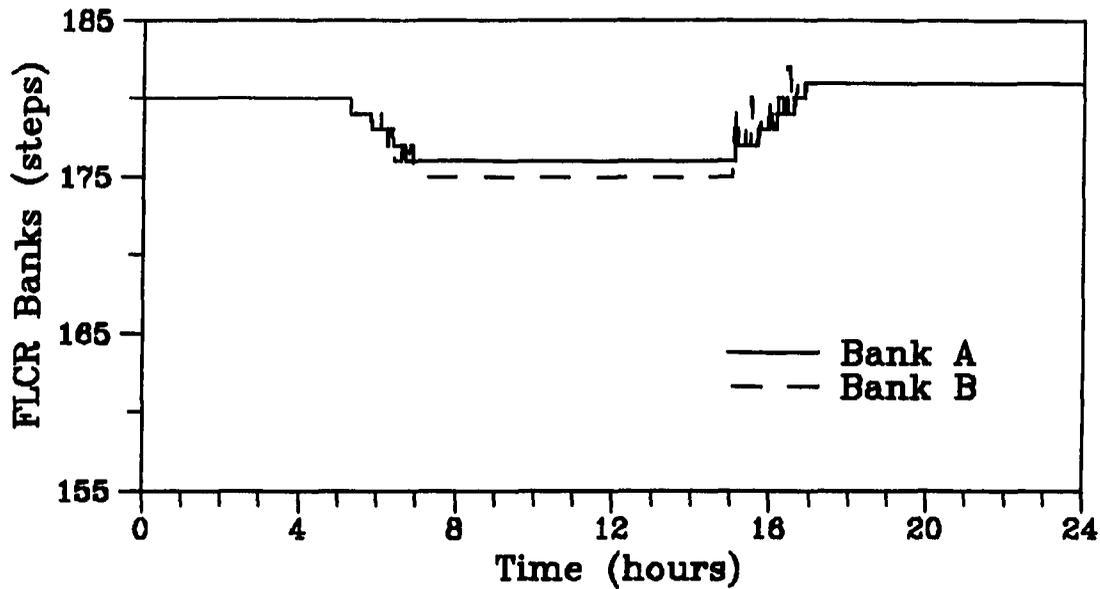


Figure 6.14: Case B: Full length control rod positions-1

Figure 6.12 that the deviations from the target load were somewhat larger during the power increase than during the power decrease. However, the reason for these deviations was not related to the direction of the power transient, but related to the unmeasurable late reactivity effects of the xenon transients because of the previous power level change. Although this effect is compensated by the boron concentration change, it is impossible to predict the xenon feedback exactly. On the other hand, the power curve in Figure 6.12 and the AO curve given in Figure 6.13 show that the heuristic rules explained in the previous chapters worked adequately to keep the variations within one percent of full power.

The rest of the test consisted in keeping the power constant at full power until the 24 hour period was completed. Any xenon reactivity effect, including the oscillation,

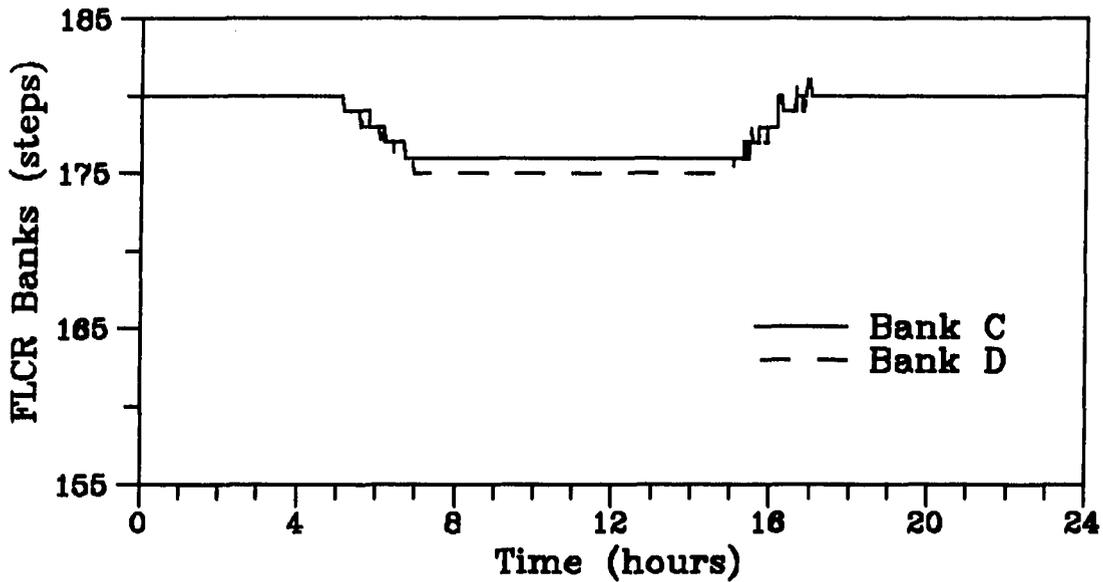


Figure 6.15: Case B: Full length control rod positions-2

was controlled by changing the boron concentration and PLCR position during this stage. A simple comparison of the magnitude of xenon oscillations in case A and B from PLCR positions (see Figures 6.7 and 6.17) will show that the new power transient which takes the reactor back to full power lessens the effect of the xenon oscillation. To control the AO, ACES proposed a much smaller change in the PLCR position in case B than in case A. In order to facilitate these comparisons, the scales are kept constant for each similar graph in the different cases.

The characteristics of case C are a short initial steady state power, longer power transient with the same rate as used in previous cases, and therefore a lower final power level. The initial steady state power was kept short in order to have a longer steady state at full power through the end of a 24 hour period. After it is shown that

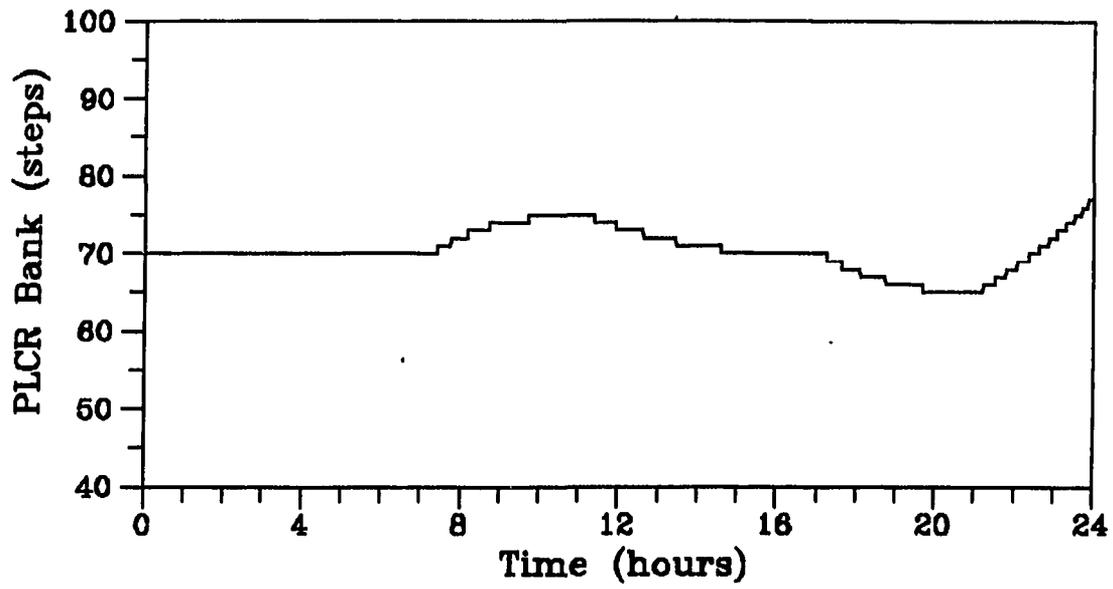


Figure 6.16: Case B: Part length control rod position

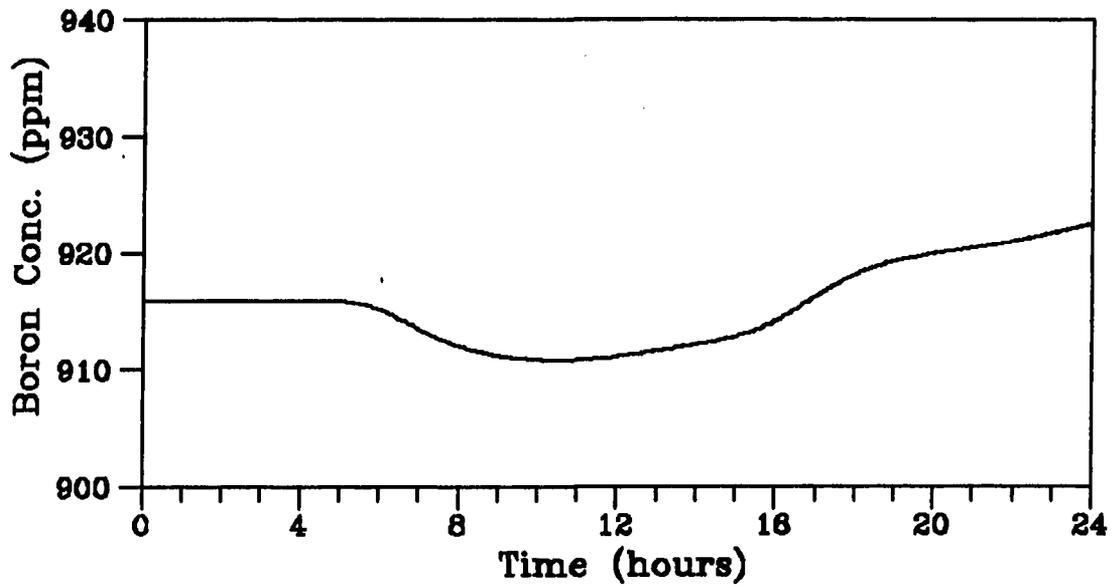


Figure 6.17: Case B: Boron concentration change

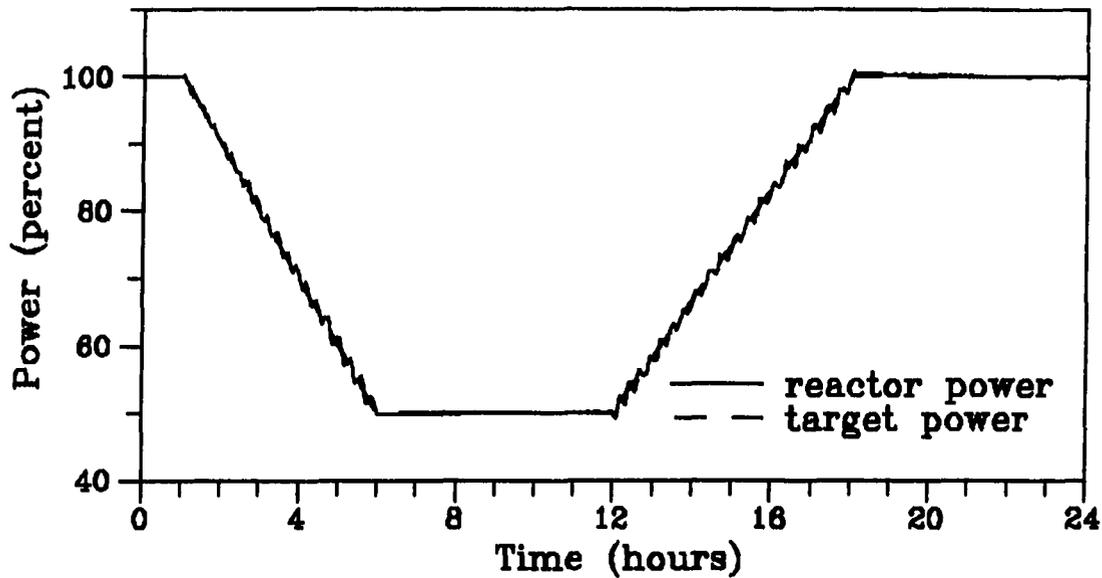


Figure 6.18: Case C: Reactor power and load demand

the reactor can stay at a steady power for a long time period, there is no need to repeat the same test. On the other hand, a long steady state at the end of a period will reveal any difficulties in controlling an oscillation.

The deviations from the desired load increased as the first transient continues in case C, showing that the unpredictable xenon feedback may appear within 6 hours. However, the deviations did not diverge during the power transient, and did not exceed one percent of full power. When Figures 6.18 and 6.22 are compared in the time domain, it can be seen that the major deviations occur especially after the PLCRs are moved. Since the net reactivity effect of a PLCR motion strongly depends upon the flux distribution, it is almost impossible to estimate the effect of the suggested PLCR control changes on the power. Moving the PLCR the same

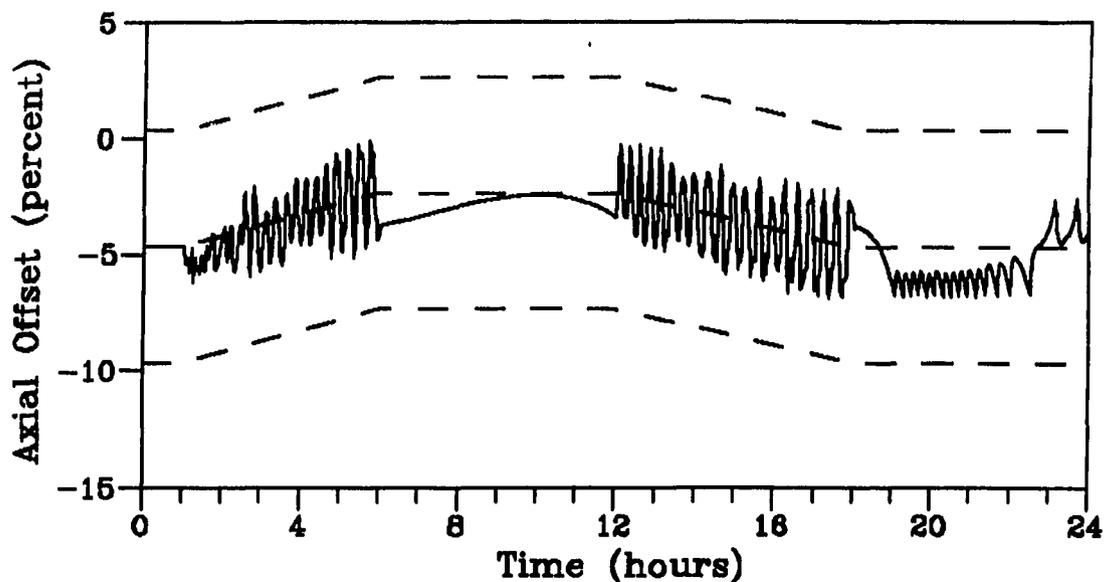


Figure 6.19: Case C: Axial offset change

amount from the same position may have a completely opposite reactivity effects for two different flux distribution. Therefore, ACES does not correct its suggested control action based on the use of PLCR, but waits to observe its effect on the power or the reactivity of the system. The rules that control the power will take care of this effect at the next cycle of control.

The Figures 6.20 and 6.21 show that the control rods may not return to the same position when the reactor power is restored to full power at the end of the daily load cycle. In case B, it happened to be that the CR positions were the same for full power. ACES does not have any rule to force the control parameters to their initial positions as was the case in previous studies [36, 47]. Depending on the transient, FLCRs may end up at any location after a power transient. However, FLCRs ended

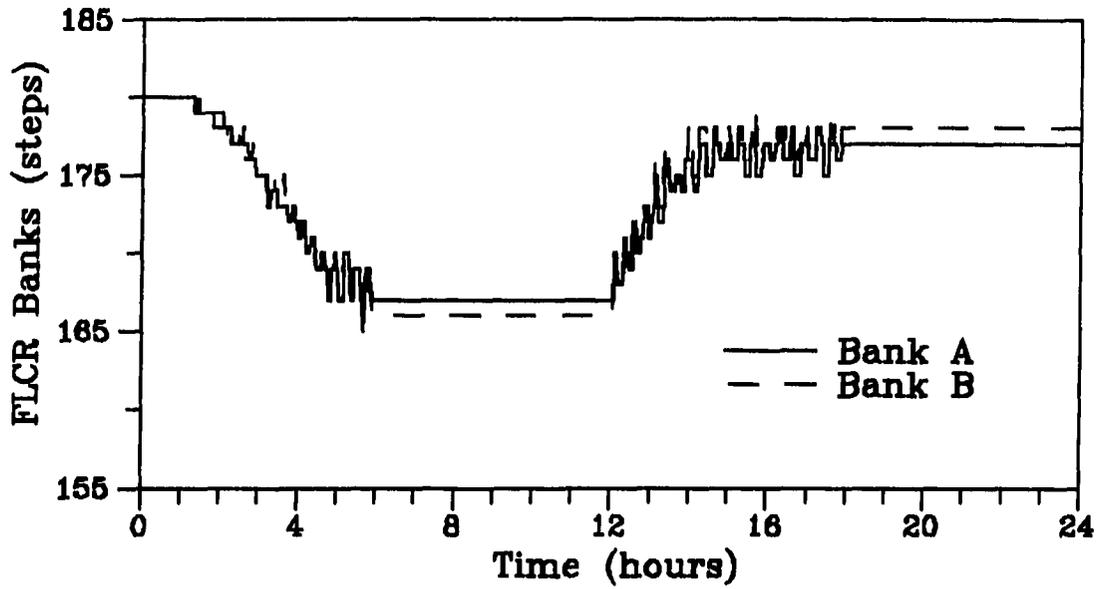


Figure 6.20: Case C: Full length control rod positions-1

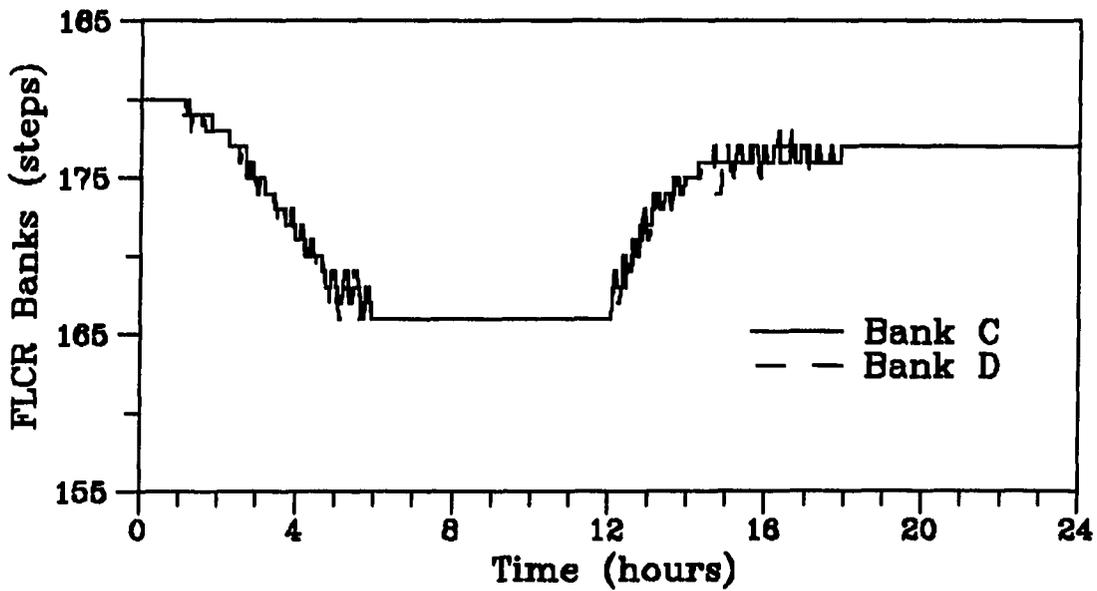


Figure 6.21: Case C: Full length control rod positions-2

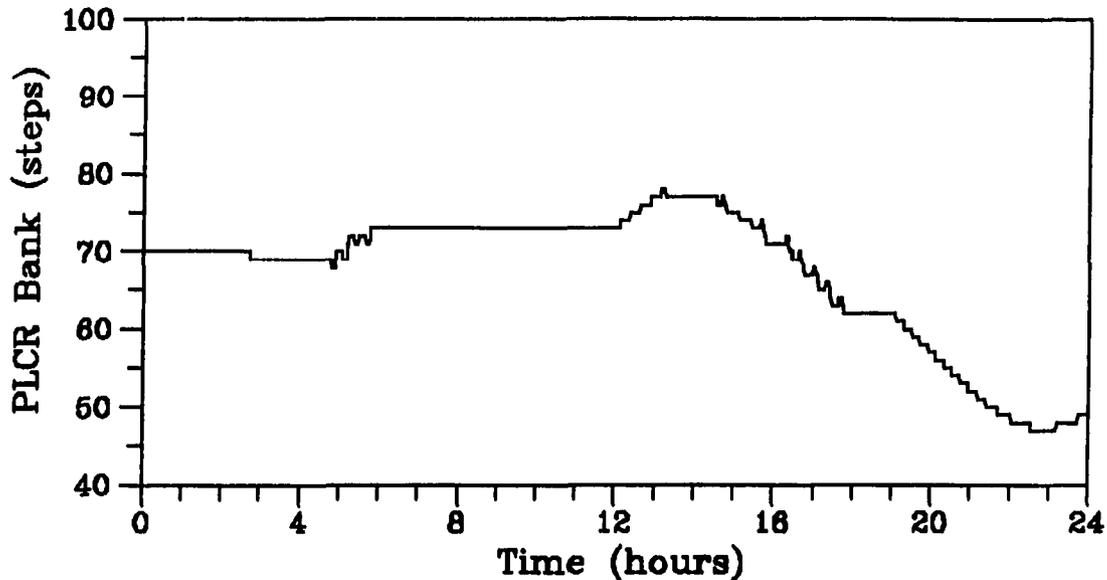


Figure 6.22: Case C: Part length control rod position

up a position in close vicinity of their initial value, 180 steps or 90 % out of the core. This shows the heuristic strength of the rule set # 4 used in ACES for power control. Although there is not an explicit rule that force the control variables to their original positions, ACES does take FLCRs back which means that the next daily load cycle would start with a similar initial state.

The lower power level was chosen to test the response of ACES to a stronger xenon oscillation because of a stronger initiating event. As can be seen from Figures 6.20 and 6.21, FLCRs were inserted in this case more than case B, which was expected to create a stronger AO shift. The strength of the xenon transient can be understood from the change of the boron concentration shown in Figure 6.23. On the other hand this change was not an oscillation but the xenon feedback following

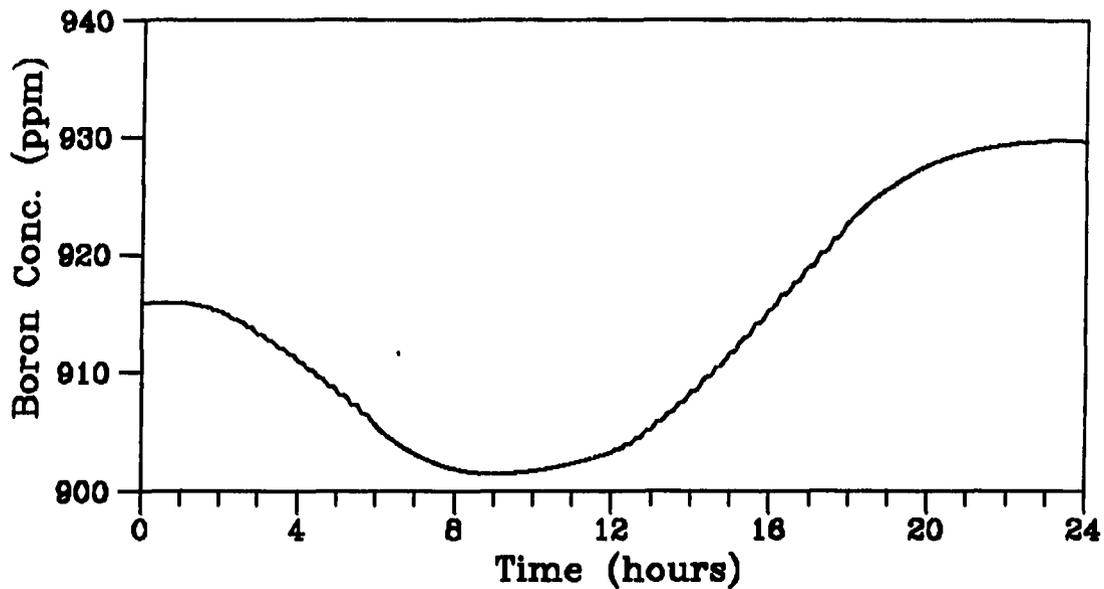


Figure 6.23: Case C: Boron concentration change

a power level change. The effect of taking the power back to 100 % again softens the oscillations, and by creating a reverse oscillation, it dampens the magnitude of oscillation. However, the oscillation was still stronger than the previous case, and ACES responded to the AO shift with PLCRs. As can be seen in Figure 6.22, ACES controlled the oscillation and began to pull PLCRs back toward the center of the core through the end of the daily cycle.

Case D was designed to test ACES for a faster rate of power change, and its handling ability for a stepwise increase of power. Instead of 10 % per hour rate, the power decreased to 50 % with a rate of 25 % per hour. Surprisingly, ACES was able to follow the load demand closer than the previous cases even though PLCRs were moved which creates unpredictable reactivity changes. One of the reasons for the

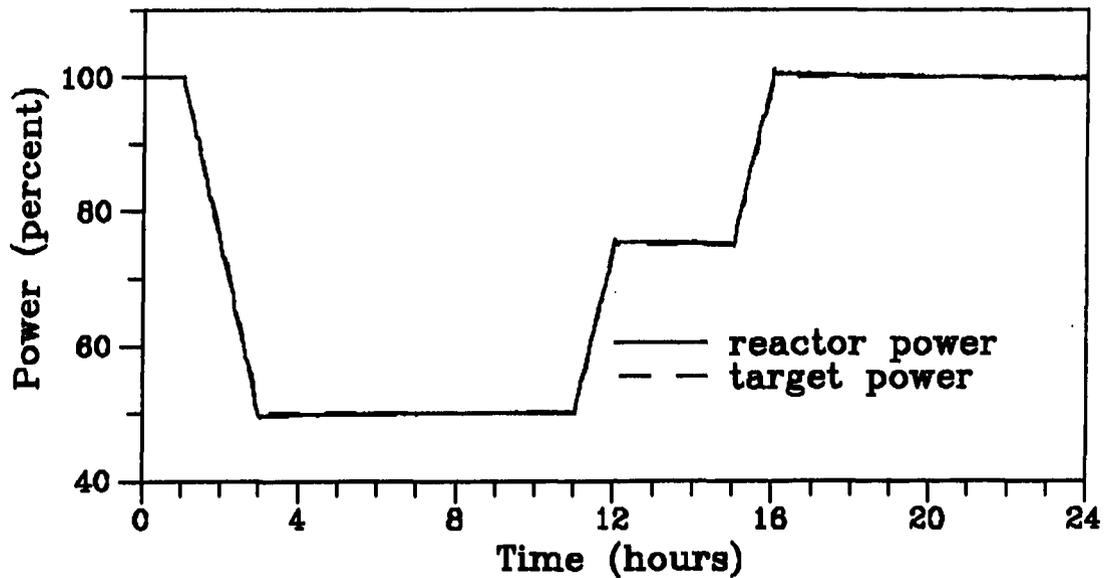


Figure 6.24: Case D: Reactor power and load demand

closer power control is the delay of the xenon feedback. Within two hours, xenon feedback does not have enough time to change the flux concentration significantly. Second, as can be seen in Figure 6.24, the unpredictable reactivity effect of a PLCR move may help the power control as illustrated in this case, or it may work against it as shown previously. These completely opposite effects of a PLCR move originate from the current flux distribution of the core, which is an unmeasurable quantity. As seen in Figure 6.29, there has been a strong AO shift to the top of the core, which was compensated by PLCRs. Therefore, it is reasonable to conclude that the PLCRs are moved to the high flux region, and hence, negative reactivity is introduced into the system, which helped the power control.

Although ACES responded to the strong AO transient through the rule set #

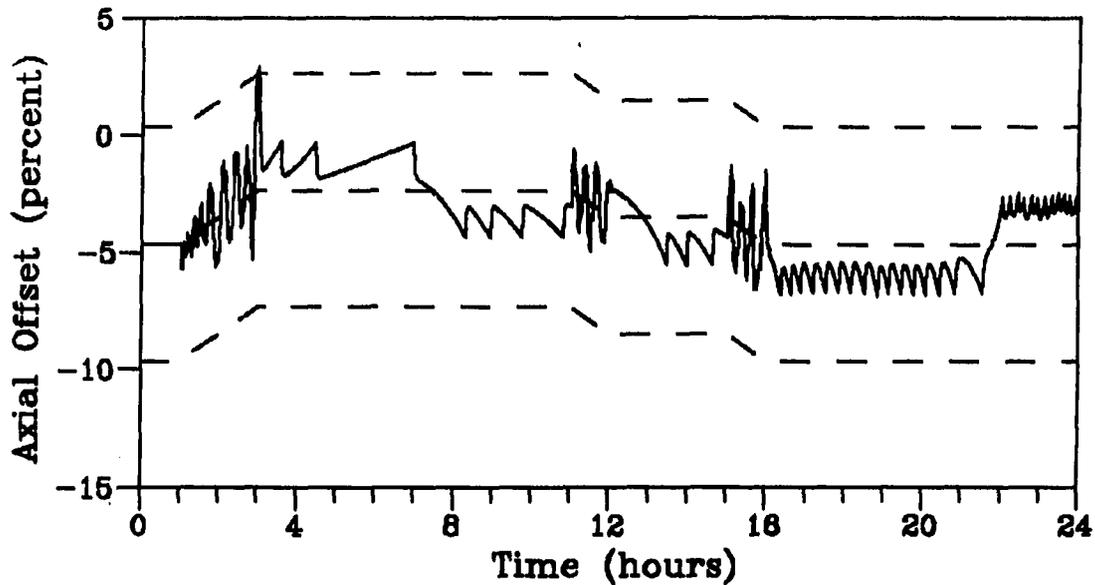


Figure 6.25: Case D: Axial offset change

2, and pulled the PLCR to the top of the core, AO had shifted out of the target band (Figure 6.25) as a unique example among the test cases. This was a result of the fast transient. Even though AO is forced back into the target band immediately, it shows an important characteristic of ACES. Because of the computational limitations of the reactor model, ACES is designed to sample the reactor every minute. However, the decision process of ACES takes much less time than a minute. We believe that if ACES samples the reactor more frequently, it will less likely allow such an excessive AO shift. Other than that one point out of the target band, the AO is kept within limits throughout the daily cycle as is shown in Figure 6.25.

Another interesting response of ACES in this case was the FLCR positions when the power reached 50% (see Figures 6.26 and 6.27). They were inserted less than the

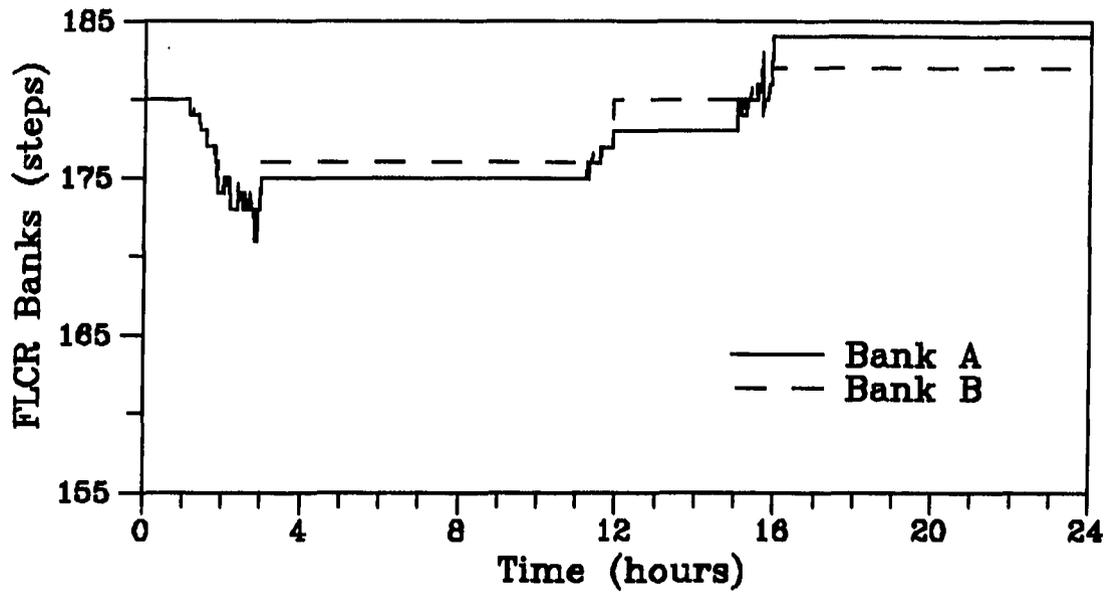


Figure 6.26: Case D: Full length control rod positions-1

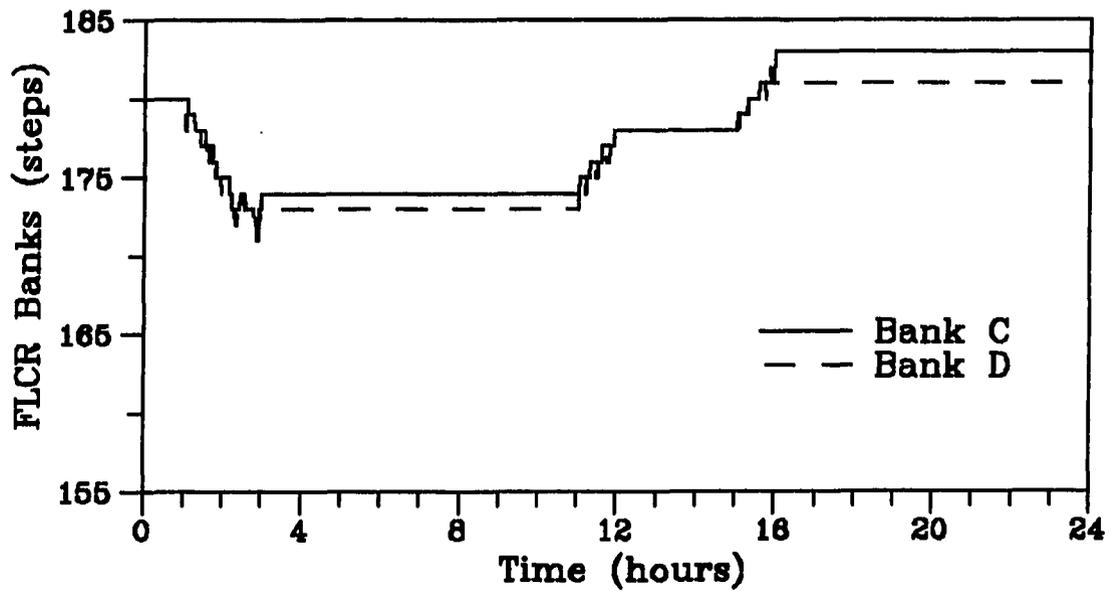


Figure 6.27: Case D: Full length control rod positions-2

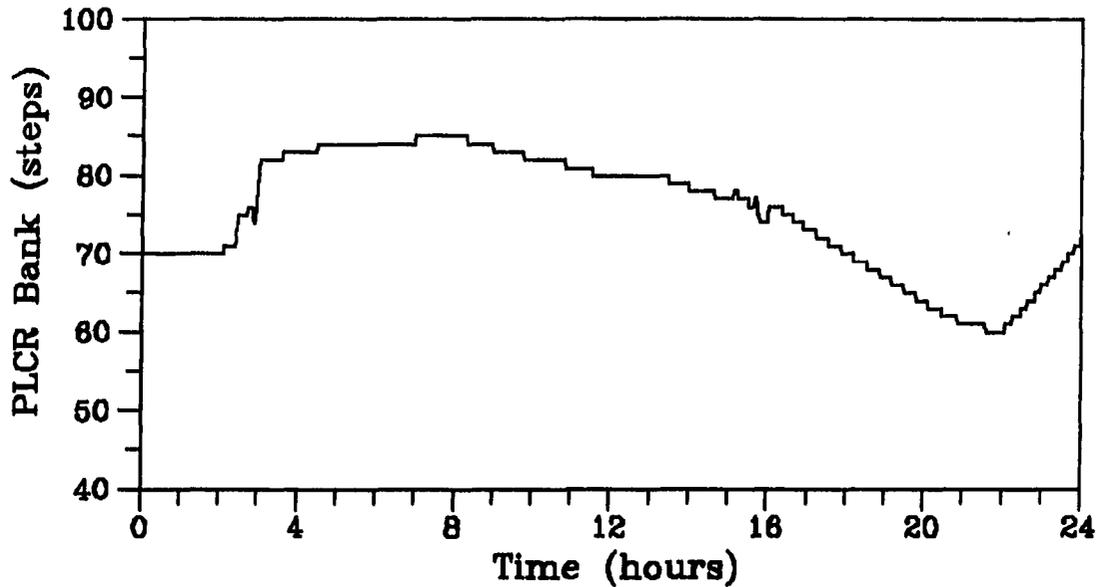


Figure 6.28: Case D: Part length control rod position

previous case in which the power was also at the same level after the transient. This should be due to the negative reactivity insertion as a result of PLCR motion during the transient. The sudden FLCR withdrawals through the very end of the power transient, following almost a total 15 steps of successive PLCR motion, can not be a simple coincidence. This example shows how strong a PLCR motion may affect the reactivity of the system. However, ACES was able to keep up with its goals, and was able to handle an unexpected reactivity change of the core.

Following the 50 % steady state operation, ACES successfully controlled the power by keeping up with the load demand with the given rate of power increase. It kept the power at 75 % as an intermediate power level, and then, took the reactor to full power as load demanded. The xenon oscillation following the transients was also

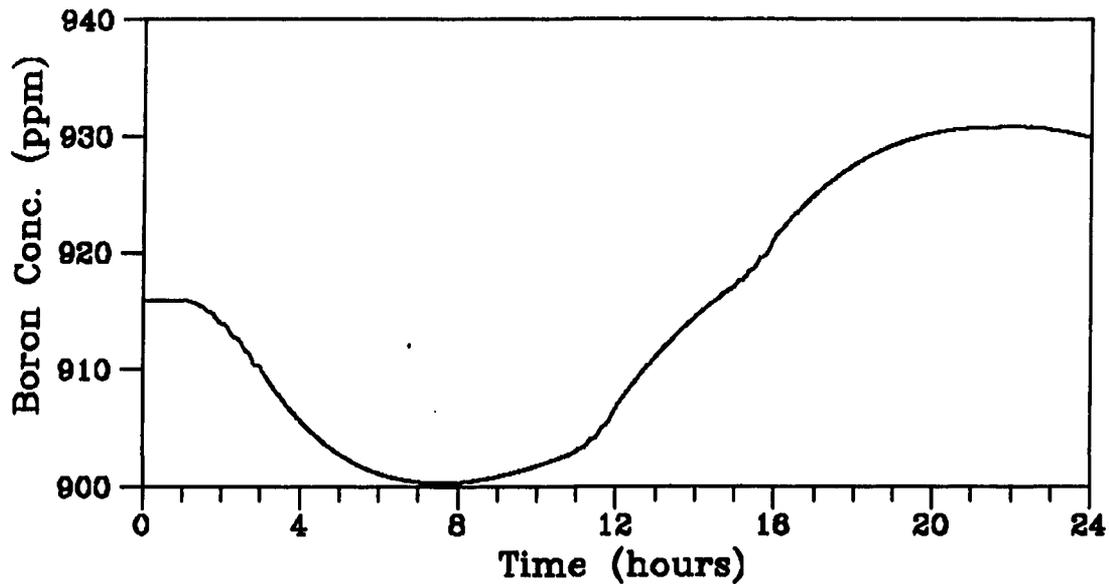


Figure 6.29: Case D: Boron concentration change

kept under control as shown in Figures 6.28 and 6.29.

The final case was not our design. After testing ACES for various cases, it was crucial to test it for an arbitrary load demand. ACES was tested against a typical power load of a PWR given by Lamarsh [4]. The given transient had some unique characteristics, such as having no steady state operation at low power. Instead, power load decreased gradually, with a slower rate than with which we had previously tested ACES. The load demand then increased back to full power, and stayed at this level for 10 hours.

The deviations of power from the load demand were highest for this case, owing to the duration of the transient, which was long enough to experience the xenon feedback. When the unpredictable reactivity effects of the xenon feedback and the

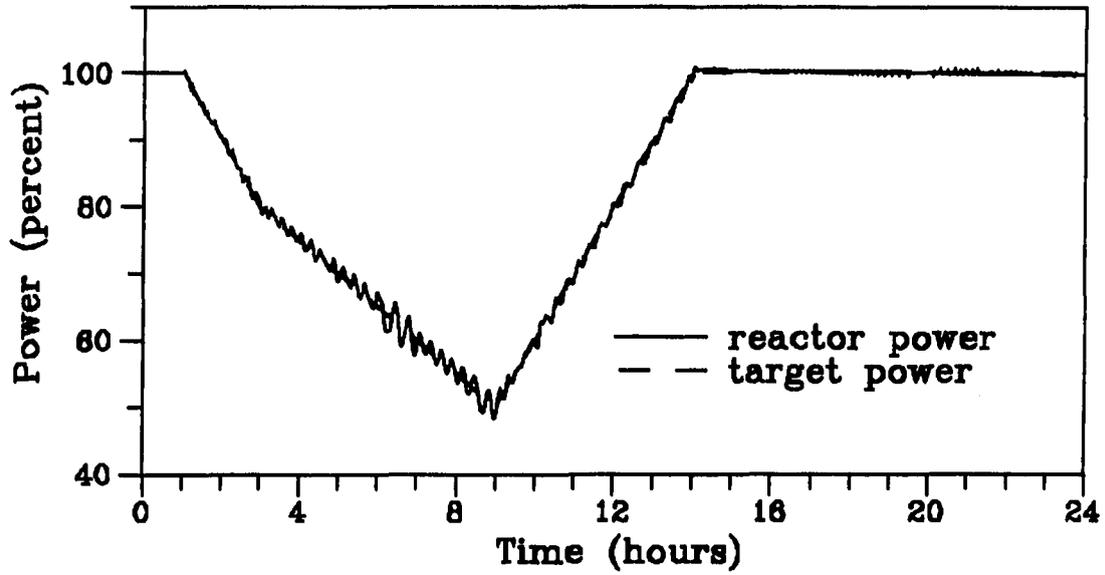


Figure 6.30: Case E: Reactor power and load demand

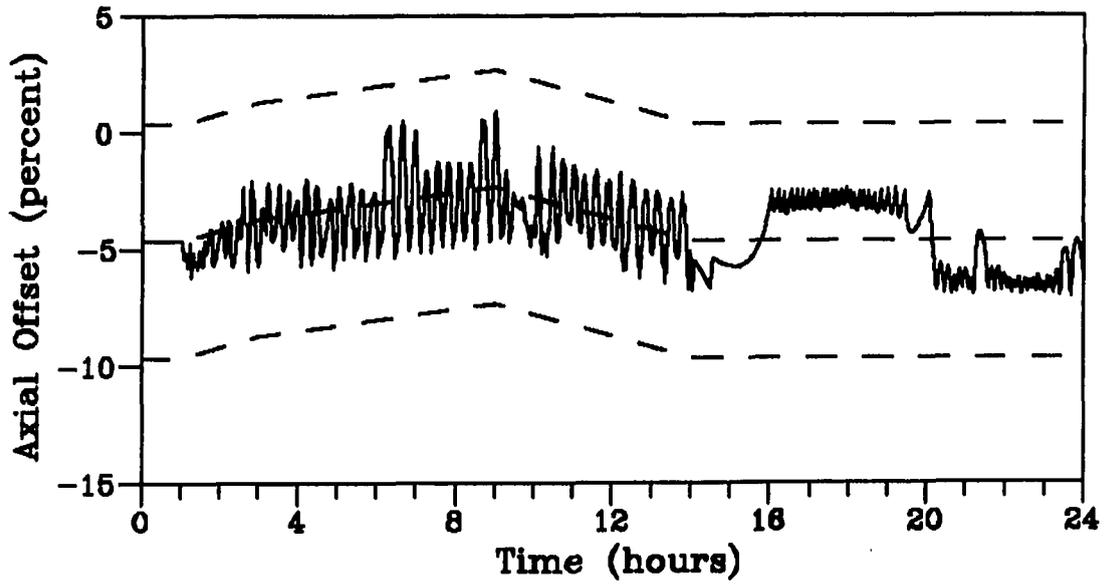


Figure 6.31: Case E: Axial offset change

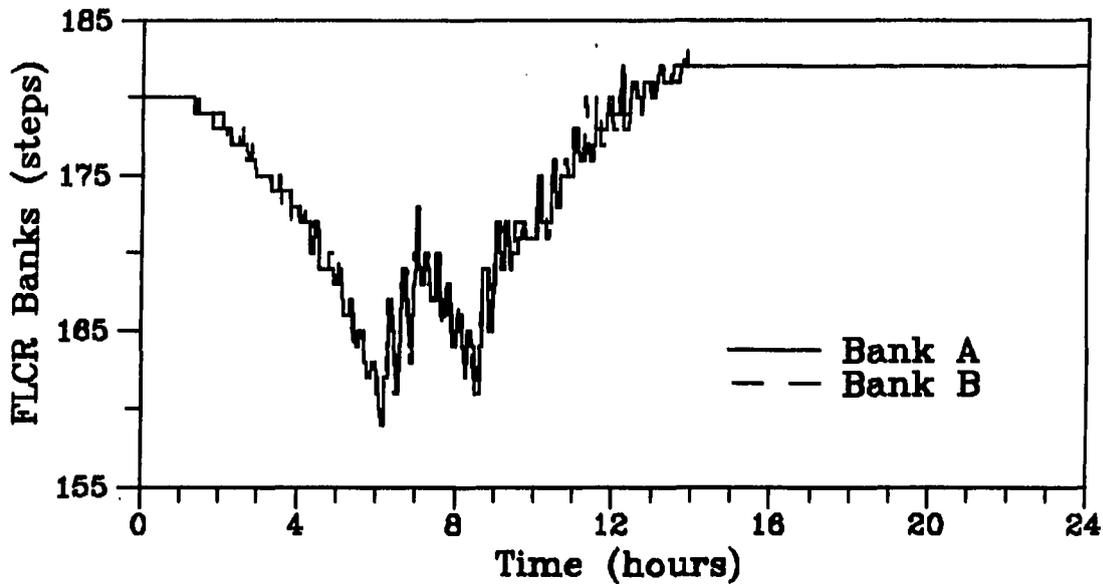


Figure 6.32: Case E: Full length control rod positions-1

PLCR motions are combined, the power control was able to keep the deviation within $\pm 3-4\%$ of the desired power (Figure 6.30). However, this should not be considered as a big flaw for the expert system, since similar or bigger power errors were experienced in an experiment of a load-follow mode demonstrations employing CAOC control procedures performed by Sipush et al. at Indian Point, Unit 2 plant in New York [10]. On the other hand, increasing the sampling frequency of the reactor is a possible solution for improving the performance of the system. By frequent sampling, ACES would detect the unpredictable reactivity changes before they cause considerable changes in power level.

Other than the given problem, ACES did what it is intended to do, and kept the reactor power, and the axial offset under control during a 24 hour period. Addition-

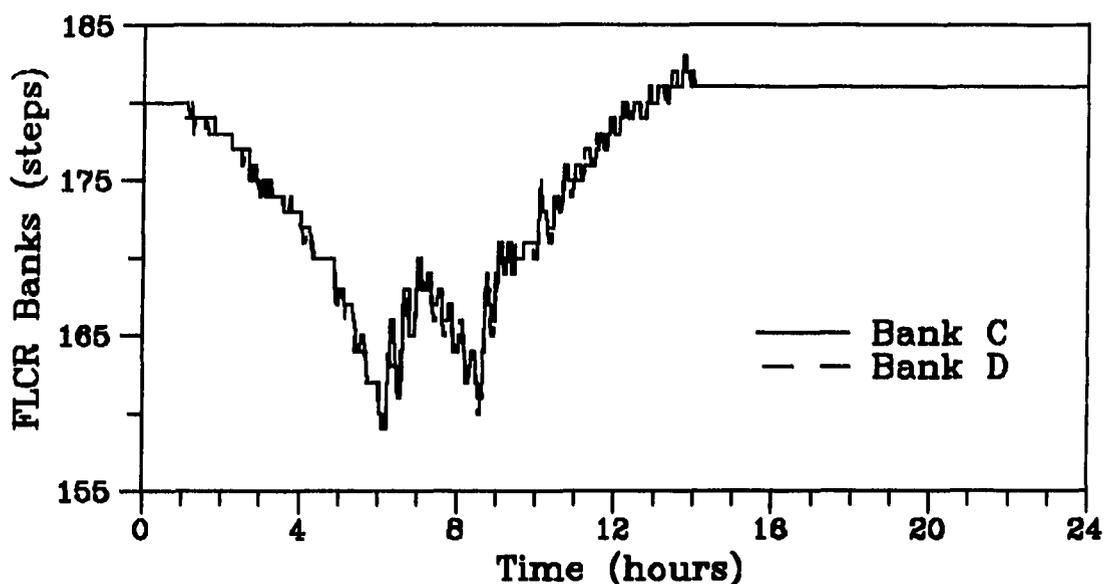


Figure 6.33: Case E: Full length control rod positions-2

ally, the xenon oscillation occurring as a result of the power transients was controlled as can be seen from PLCR position and boron concentration curves, Figs. 6.34 and 6.35, respectively. ACES successfully switched from boration to deboration and back to boration during the power transient by using the rules # 3.1, 3.2, 3.4, and 3.7 as seen in Figure 6.35.

The final evaluation point for an expert system should be the speed of the decision process. Additionally, it has been noted in this study that the sampling frequency of the reactor in ACES is set to 1 minute by the limitations of the model, and that more frequent sampling would solve the AO drift out of the target band experienced in case D. In this respect, OPS5 provides an effective tool, the Performance Measurement and Evaluation package, which provides CPU time reports when necessary [50].

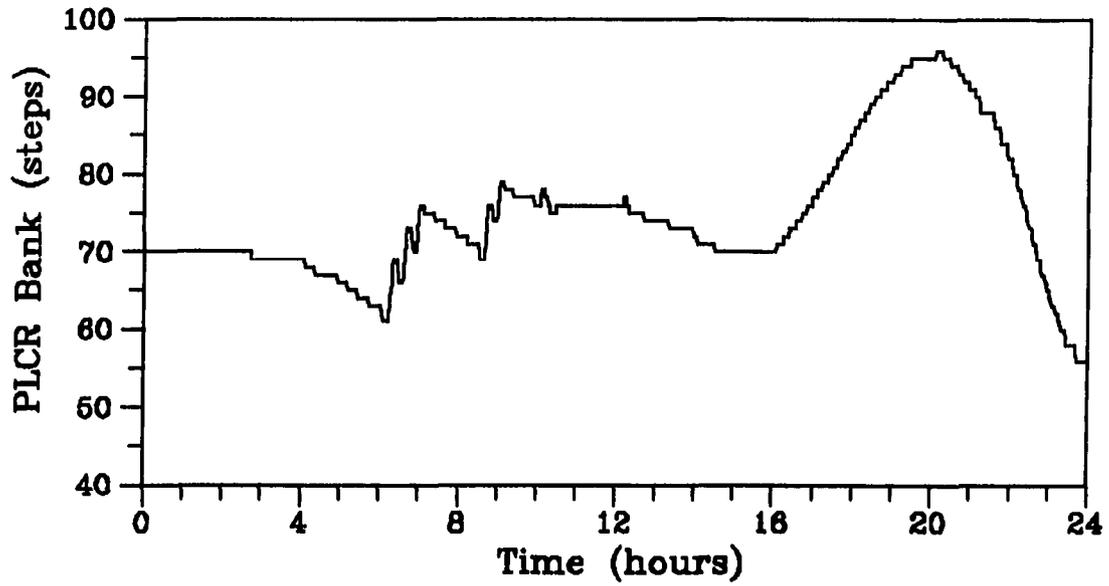


Figure 6.34: Case E: Part length control rod position

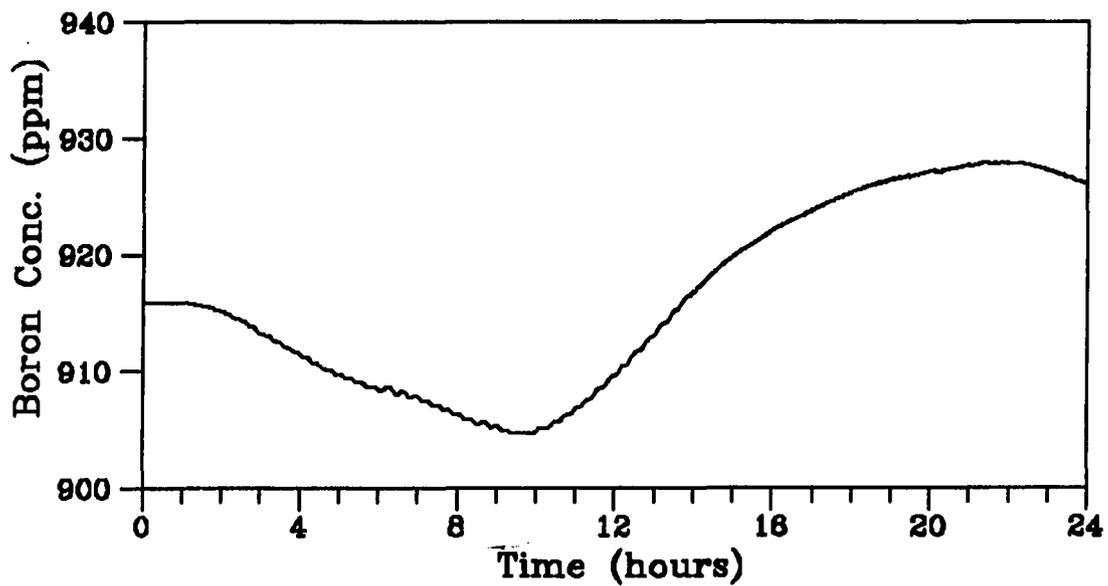


Figure 6.35: Case E: Boron concentration change

Table 6.4: CPU time for initialization of ACES

Timing CPU report on 24-JUL-1992 15:41:11.13			
PRODUCTION NAME	# FIRINGS	LHS TIME	RHS TIME
clean_old_crod_wmes	0	1	0
initialization	1	0	168
moving_selected_control_rod	0	47	0
preference	0	2	0
stop_moving_cr	0	35	0
initialize_control_variables	0	603	0
read_differential_worths	801	0	1633
end_of_data	1	88	3
insertion_limit_on_moving_cr	0	1	0
set_crod_reactivity_worths_and_limits	0	103	0

This package is used for generating some timing reports of ACES, and the reports are presented in its original form in Tables 6.4 through 6.6.

The CPU time consumed by ACES during the initialization process took a relatively long time because of creating the active working memory for the first time. The numbers given under the title of LHS and RHS times are the 10-millisecond ticks of CPU time in VAX computer system, used for executing the given side of a production. Therefore, the initialization of working memory costs a CPU time of less than 27 seconds in VAX, including the conflict resolution process and accessing the user interface for gathering the power load information. As it can be seen in Table 6.4, the conflict resolution amounts to almost 33 % of the total time, and 90 % of it is spent for the productions that are not even fired. Since the initialization process is performed only once in this scale, design criteria were to set the working memory, not to save time.

On the other hand, unnecessary matches that increase the time spent for the

conflict resolution process were avoided as much as possible for an ordinary control cycle of ACES. Table 6.5 gives CPU time consumed during the steady state power operation, which is possibly the fastest cycle in VAX. The total CPU time spent in this cycle was 1.21 seconds. Out of 1.21 seconds, 0.4 seconds were spent for the conflict resolution, and 0.1 seconds were spent for productions that didn't fire.

Table 6.5: CPU time for control cycle of ACES at steady state power

Timing CPU report on 24-JUL-1992 22:14:19.40			
PRODUCTION NAME	# FIRINGS	LHS TIME	RHS TIME
clean_old_crod_wmes	4	1	2
modify_response_in_transient_1	0	2	0
no_change	1	0	3
physical_limit_on_moving_cr	0	1	0
preference	1	15	19
stop_moving_cr	0	2	0
initialize_control_variables	1	4	28
reset_direction	0	1	0
calling_reactor_for	0	1	0
end_of_cr_moves	1	1	1
insertion_limit_on_moving_cr	0	3	0
set_crod_reactivity_worths_and_limits	4	9	21
no_boron_control	1	0	5
no_axial_offset_control_needed	1	0	2

A control cycle during a power transient will definitely take more time since we expect more rules to be fired. A sample cycle CPU time report is given in Table 6.6 for transient cases. The total CPU time was 1.76 seconds for this case in VAX system. Although this case may not be the longest cycle, it gives an idea about the time domain for transient cases. There might be more CR movements which amounts to 1 tick for each step as is listed for the production named *moving selected control rod* in Table 6.6. Therefore, the sampling frequency of ACES can easily be reduced to 5

Table 6.6: CPU time for control cycle of ACES during a power transient

Timing CPU report on 24-JUL-1992 22:17:16.56			
PRODUCTION NAME	# FIRINGS	LHS TIME	RHS TIME
clean_old_crod_wmes	4	4	4
modify_response_in_transient_1	1	2	3
moving_selected_control_rod	1	0	1
physical_limit_on_moving_cr	0	3	0
preference	2	25	24
stop_moving_cr	1	1	13
initialize_control_variables	1	7	22
calling_reactor_for	0	1	0
end_of_cr_moves	1	0	1
insertion_limit_on_moving_cr	0	1	0
set_crod_reactivity_worths_and_limits	5	15	30
providing_negative_reactivity	1	0	4
boron_control_in_transient	1	1	11
providing_positive_reactivity	0	1	0
no_axial_offset_control_needed	1	0	2

seconds with almost 3 seconds of safety margin, and a closer watch can be provided for the AO and power control.

The CPU time evaluation also shows the importance of the generic characteristic of ACES. In a design based on pattern recognition, we would have more rules to handle every specific case, including some very similar ones. The conflict resolution process would consider each one by one and would cost more CPU time. The generic rules cut down the number of rules and the CPU time.

CHAPTER 7. CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The purpose of this study was to explore the application of expert system technology to the spatial xenon oscillation control problem in a typical PWR. An expert system, ACES has been developed to implement the Heuristic Constant Axial Offset Control strategy for the control of spatial xenon oscillations in PWRs. ACES is designed to be as realistic as possible with a set of generic rules to increase its area of applicability.

ACES is designed to use only measurable reactor parameters during the decision process, and to evaluate the reactor status for determination of necessary control action without using any non-measurable "heuristic" constants. It samples the reactor status in terms of available parameters, and evaluates these data to determine the necessity of a control action. It uses the differential rod worth curves, and reactivity worth of boron control as a knowledge base, and requires the user to supply this information.

The knowledge base is built into the active working memory of the expert system instead of the productions. This enabled ACES to adapt its control parameters to the reactor status during runtime. In addition, it increased the applicability of ACES to any reactor since the knowledge base of ACES is kept as a set of measur-

able quantities, such as the control rod worth curves of a PWR. Provided that the knowledge base is loaded in ACES, it is capable of controlling an arbitrary PWR for an arbitrary load demand.

As a result of the nature of the xenon oscillation problem, a forward chaining algorithm was found effective for this particular application. ACES used forward chaining to search for the solution of the given core status. A given core status is not recognized by a pattern recognition procedure in ACES, but it is evaluated to solve for the necessary corrective action which may include no correction at all. This unique feature of the design also enhanced the generic characteristic of ACES. No quantification of error terms, or no ranging for parameters is used for creating patterns unless they are well known and common for PWRs. The $\pm 5\%$ target band of AO around the steady state value is the only ranging used in ACES, and it is a very common parameter [10].

Although there is no other previous study directly comparable with ACES, it has been tested against some cases used by Cho [36] and Chung [47] in their studies. The test cases are extended for different power levels, and different transient rates in ACES. ACES successfully controlled the PWR core model for the given test cases even though the model overpredict the xenon feedback. Therefore, it is expected to perform better in real applications. Finally, ACES is tested against an arbitrary load demand given in the literature [4], and its performance is evaluated based on CPU time consumption. It is shown that ACES can follow any load demand and can keep the AO within the target band.

The following conclusions have been drawn from this study:

1. It is possible to use expert system technology in spatial xenon oscillation

control in PWRs.

2. The forward chaining is a useful technique for this particular application.
3. It is possible to implement an expert system controller based on only measurable parameters of a reactor.
4. The expert system ACES implemented in this study is capable of controlling a load-follow PWR for any load demand provided that the control rod worths are supplied as the knowledge base.
5. ACES is able to sample the reactor status much more frequently -twenty times more in the VAX computer system- than the interval used in this study, which ensures much finer power control than the ones shown in the sample cases.
6. Designed as a generic code, ACES is free of "heuristic" constants, it adapts its control parameters in runtime, and therefore, can be used in an arbitrary PWR.

As a final reminder, we would like to add that, beside the advantages of expert systems, this technology helps to preserve the expertise in case of an interruption in the inheritance of the knowledge. It will ensure an ongoing expert control in times of personnel changes at the plant. On the other hand, even though expert system technology is not now or may never be developed enough to simulate the human creativity, it has the ability to serve useful applications. Lacking of creativity, however, only limits the application areas of expert systems, and being a slow transient, xenon oscillations are not one of these limited areas.

Recommendations for Future Work

The ultimate goal of this study is to design a controller to replace the reactor operator in the control of xenon oscillations. Unfortunately, the cost of a complete verification of computer software and hardware, and the environmentally opinionated insecure feeling of the general public limit the expert systems to be an advisory controller for the time being. Therefore, ACES should be furnished with user friendly graphic displays for presenting the results to the reactor operator. Although, ACES currently displays the current status of the core, and suggests control actions, no specific attention is paid to the form of the output.

Although ACES has been tested against several load schedules using a PWR core model, it should be tested for more cases using a more detailed model. Also, the sampling frequency of ACES should be increased to achieve better control, and to reduce errors in load-follow. Today, most of the reactors have their own simulators which are able to simulate any transient with enough accuracy. ACES should be tested using these simulators prior to any real time applications.

Additionally, ACES has been tested only against a core configuration that simulates the beginning of life conditions. However, the limitations on the boron concentration changes at the end of life of a fuel cycle require special attention. Therefore, ACES should be tested and modified -if necessary- for the end of life conditions of a fuel cycle.

And finally, ACES is designed using a VAX cluster which may not be available everywhere. A microcomputer version can be generated provided that the microcomputer version of the OPS5 compiler supports an interface with a conventional language.

BIBLIOGRAPHY

- [1] J. J. Duderstadt, L. J. Hamilton, Nuclear Reactor Analysis, (John Wiley and Sons, New York, NY, 1976).
- [2] J. R. Lamarsh, Introduction to Nuclear Reactor Theory, (Addison-Wesley, Reading, MA, 1966).
- [3] D. Randall, D. S. St. John, "Xenon Spatial Oscillations," *Nucleonics*, Vol. 16, No. 3, 82 (1958).
- [4] J. R. Lamarsh, Introduction to Nuclear Engineering, (Addison-Wesley, Reading, MA, 1977).
- [5] W. M. Stacey Jr., "Control of Xenon Spatial Oscillations," *Nuclear Science and Engineering*, Vol. 38, No. 3, 229 (1969).
- [6] P. Jackson, Introduction to Expert Systems, (Addison-Wesley, Reading, MA, 1990).
- [7] A. A. Berk, LISP: The Language of Artificial Intelligence, (Van Nostrand Reinhold, New York, NY, 1985).
- [8] H. Schildt, Artificial Intelligence... Using C, (Osborne McGraw Hill, Berkeley, CA, 1978).
- [9] P. H. Winston, Artificial Intelligence, (Addison-Wesley, Reading, MA, 1979).
- [10] P. J. Sipush, R. A. Kerr, A. P. Ginsberg, T. Morita, L. R. Scherpereel, "Load-Follow Demonstrations Employing Constant Axial Offset Power-Distribution Control Procedures," *Nuclear Technology*, Vol. 31, No. 1, 12 (1976).
- [11] D. C. Bauer, C. G. Poncelet, "Practical Xenon Spatial Control," *Nuclear Technology*, Vol. 21, No. 3, 165 (1974).

- [12] J. March-Leuba, E. D. Blakeman, "A Study of Out-Of-Phase Power Instabilities in Boiling Water Reactors," 1988 (unpublished).
- [13] R. L. Crowther, "Nonlinear Xenon Effects in Boiling Water Reactors," *Trans. of Am. Nucl. Society*, Vol. 11, 570 (1968).
- [14] O. A. Trojan, "Analysis of Spatial Flux Control in Heavy Water Reactors," *Trans. of Am. Nucl. Society*, Vol. 11, 571 (1968).
- [15] R. Nabbi, "Analysis of the Xenon Feedback on the Core Dynamics of High-Temperature Reactors During Heat Removal Transients without Reactor Shutdown," *Nuclear Technology*, Vol. 64, No. 1, 5 (1984).
- [16] A. M. Afanasev, B. Z. Torlin, "Stability of the High-Level Field of Reactor Neutrons with respect to Xenon Oscillations," *Soviet Atomic Energy*, Vol. 44, No. 6, 566 (1978).
- [17] S. V. Topp, R. P. Germann, "Xenon Oscillation Experiments in a Production Reactor," *Trans. of Am. Nucl. Society*, Vol. 12, No. 2, 765 (1969).
- [18] W. E. Graves, "Experience with Xenon Oscillations," *Trans. of Am. Nucl. Society*, Vol. 12, No. 2, 644 (1969).
- [19] D. Rawle, V. Rajagopal, C. G. Poncelet, R. J. Johnson, J. R. Himmelwright, "Spatial Xenon Stability Measurements on the Connecticut Yankee Reactor," *Trans. of Am. Nucl. Society*, Vol. 12, No. 2, 766 (1969).
- [20] J. C. Lee, K. A. Jones, W. L. McCoy, "Axial Stability Measurements at the Rochester Gas and Electric Reactor," *Trans. of Am. Nucl. Society*, Vol. 14, Suppl. 2, 15 (1971).
- [21] D. D. Ebert, J. R. Humphries, R. M. Versluis, W. J. Lippold, "Maneuvering Experience at Calvert Cliffs, Unit 1," *Trans. of Am. Nucl. Society*, Vol. 26, Suppl. 1, 62 (1977).
- [22] W. R. Corcoran, J. R. Humphries, H. J. Litke, J. D. LeBlanc, "Dumping of Xenon Oscillations in the Maine Yankee Reactor," *Nuclear Technology*, Vol. 22, No. 2, 252 (1974).
- [23] W. R. Casto, "Safety Related Occurrences Reported in October and November 1972 (Axial Xenon Oscillations Controlled at a PWR)," *Nuclear Safety*, Vol. 14, No. 2, 121 (1973).

- [24] S. Schaefer, "Procedure for Damping of Xenon Oscillations," Northern States Power Company, 1984 (unpublished).
- [25] B. J. Delano, "Operational Flexibility of B&W PWR Control Systems for Load Follow," *Trans. of Am. Nucl. Society*, Vol. 26, Suppl. 1, 64 (1977).
- [26] J. Karppinen, "Spatial Reactor Control Methods," *Nucl. Sci. and Engineering*, Vol. 64, No. 2, 657 (1977).
- [27] A. M. Christie, C. G. Poncelet, "On the Control of Spatial Xenon Oscillations," *Nucl. Sci. and Engineering*, Vol. 51, No. 1, 10 (1973).
- [28] S. Tzafestas, "Distributed Parameter Nuclear Reactor Optimal Control," *Proc. IRIA Symp. New Trends in Systems Analysis*, Versailles, France, (1976).
- [29] K. Surendran, "Quasistatic Control of Xenon Spatial Oscillations," *Int. J. Systems Sci.*, Vol. 6, No. 3, 269 (1975).
- [30] W. M. Stacey Jr., "A Numerical Study of Xenon-Power Spatial Oscillations," *Trans. of Am. Nucl. Society*, Vol. 11, 226, (1968).
- [31] W. Hanke, "A Method for Solving the Xenon Oscillation Control Problem," *Nucl. Sci. and Engineering*, Vol. 72, No. 2, 265 (1979).
- [32] W. M. Stacy Jr, "Optimal Control of Xenon-Power Spatial Transients," *Nucl. Sci. and Engineering*, Vol. 33, No. 2, 162 (1968).
- [33] H. D. Purandare, S. V. Lawande, "A Dynamic Programming Approach to Spatial Control of Large CANDU Type Reactors," Bhabha Atomic Research Centre (unpublished).
- [34] E. J. Schulz, J. C. Lee, "Time-Optimal Control of Spatial Xenon Oscillations to a Generalized Target," *Nucl. Sci. and Engineering*, Vol. 73, No. 2, 140 (1980).
- [35] A. T. Chiang, P. E. Bennecke, R. A. Rydin, "Optimal Control of Xenon Spatial Oscillations in Reactors," *Trans. of Am. Nucl. Society*, Vol. 24, 427 (1976).
- [36] N. Z. Cho, L. M. Grossman, "Optimal Control for Xenon Spatial Oscillations in Load Follow of a Nuclear Reactor," *Nucl. Sci. and Engineering*, Vol. 83, No. 1, 136 (1983).
- [37] R. J. Onega, R. A. Kisner, "Parameter Identification in the Xenon Oscillation Problem Using the Method of Maximum Likelihood," *Trans. of Am. Nucl. Society*, Vol. 30, 745 (1978).

- [38] R. W. Pack, P. L. Chambré, "Observer for Spatial Xenon Oscillation Control," *Trans. of Am. Nucl. Society*, Vol. 19, 177 (1974).
- [39] G. T. Park, G. H. Miley, "Application of Adaptive Control to a Nuclear Power Plant," *Nucl. Sci. and Engineering*, Vol. 94, No. 2, 145 (1986).
- [40] R. C. Berkan, B. R. Upadhyaya, L. H. Tsoukalas, R. A. Kisner, "Reconstructive Inverse Dynamics Control and Application to Xenon-Induced Power Oscillations in Pressurized Water Reactors," *Nucl. Sci. and Engineering*, Vol. 109, No. 2, 171 (1991).
- [41] N. Noito, A. Sakuma, K. Shigeno, N. Mori, "A Real-Time Expert System for Nuclear Power Plant Failure Diagnosis and Operational Guide," *Nuclear Technology*, Vol. 79, No. 3, 284 (1987).
- [42] R. C. Erdmann, B. K-H. Sun, "An Expert System Approach for Safety Diagnosis," *Nuclear Technology*, Vol. 82, No. 2, 162 (1988).
- [43] N. Barbet, M. Dumas, G. Mihelich, Y. Souchet, J. B. Thomas, "Expert Systems for the Analysis of Transients on Nuclear Reactors: SEXTANT, A General-Purpose Physical Analyzer," *Nucl. Sci. and Engineering*, Vol. 100, No. 4, 435 (1988).
- [44] J. O. Yang, S. H. Chang, "A Diagnostic Expert System for the Nuclear Power Plant Based on the Hybrid Knowledge Approach," *IEEE Trans. on Nucl. Science*, Vol. 36, No. 6, 2450 (1989).
- [45] R. Bhatnagar, D. W. Miller, B. K. Hajek, J. E. Stasenko, "An Integrated Operator Advisor System for Plant Monitoring, Procedure Management, and Diagnosis," *Nuclear Technology*, Vol. 89, No. 3, 281 (1990).
- [46] R. E. Uhrig, "Opportunities for Automation and Control of the Next Generation of Nuclear Power Plants," *Nuclear Technology*, Vol. 88, No. 2, 157 (1989).
- [47] S. K. Chung, "Implementation of an Expert System for Xenon Spatial Control in Pressurized Water Reactors," Ph.D. Dissertation, Iowa State Univ., (1988).
- [48] L. Brownston, E. Kant, R. Farrell, N. Martin, Programming Expert Systems in OPS5 : An Introduction to Rule Based Programming, (Addison-Wesley, Reading, MA, 1985).
- [49] Vax OPS5 User's Manual, (Digital Equipment Corp., Maynard, MA, Feb. 1988).

- [50] Vax OPS5 Reference Manual, (Digital Equipment Corp., Maynard, MA, Feb. 1988).
- [51] R. J. Onega, R. A. Kisner, "A Two-Point Xenon Oscillation Model Using a Variational Principal," *Trans. of Am. Nucl. Society*, Vol. 24, 431 (1976).
- [52] J. Josephson, "Servo-Analysis and Stabilization of Xenon Oscillations in Large Thermal Nuclear Reactor Power Plants," *IEEE Trans. on Nucl. Sci.*, Vol. NS-25, No. 1, 875 (1978).
- [53] J. D. Teachman, R. J. Onega, "The Influence of Energy Group Structure and Nonlinearities on the Calculation of Xenon-Induced Flux Oscillations," *Nucl. Sci. and Engineering*, Vol. 83, No. 1, 149 (1983).
- [54] W. F. Ames, Numerical methods for partial differential equations, (2nd Edition, Academic Press, New York, NY, 1977).
- [55] L. C. Schmid, Critical Assemblies and Reactor Research, (John Wiley and Sons, New York, NY, 1971).
- [56] Babcock-241 Standard Safety Analysis Report (B-SAR-241), (Babcock & Wilcox Co., Vols. 1-4, 1974).

APPENDIX A: SOURCE CODE OF ACES

ACES is written in OPS5 and furnished with comments for each rule to increase the understandability of the program. Rules are enumerated for referencing purposes.

```

;
; Declarations
;
(VECTOR-ATTRIBUTE
  POSTN ; This vector will contain the current control rod
        ; positions at any time
  GOAL) ; This vector will contain the information for the
        ; flow of control
;
(EXTERNAL
;
; Functions
;
  (RREAC FLOAT-ATOM (FLOAT-ATOM BY REFERENCE)
(FLOAT-ATOM BY REFERENCE) (FLOAT-ATOM BY REFERENCE))
;
  (TREND INTEGER-ATOM (FLOAT-ATOM BY REFERENCE)
(FLOAT-ATOM BY REFERENCE) )
;
  (ABSOLUTE FLOAT-ATOM (FLOAT-ATOM BY REFERENCE))
;
  (AOTREND INTEGER-ATOM (FLOAT-ATOM BY REFERENCE)
(FLOAT-ATOM BY REFERENCE) (FLOAT-ATOM BY REFERENCE) )
;
  (BORON INTEGER-ATOM (FLOAT-ATOM BY REFERENCE)

```

```

      (FLOAT-ATOM BY REFERENCE) )
;
(TINLET FLOAT-ATOM (FLOAT-ATOM BY REFERENCE)
 (FLOAT-ATOM BY REFERENCE) )
;
; Subroutines
  (INIT)
  (LOAD)
  (REACTOR)
  (OUTPUT))
;
; Literalizations of Working Memory Elements
;
(LITERALIZE DWORTHS ; Knowledge base for differential worth curves
  BANK ; Name of the FLCR bank
  STEP ; Position of the bank
  WORTH) ; Differential worth at the given position
;
(LITERALIZE BORON ; Knowledge base for boron reactivity worth
  UNIT ; Minimum unit of boron in ppm
  WORTH ; Reactivity worth of a unit change
  STEPS ; Amount of control in multiples of UNIT
  CHANGE ; Amount of control in ppm
  DIRECTION) ; Flag, 1 for boration, -1 for deboration
;
(LITERALIZE POWER ; Knowledge base WME for the power schedule
  FROM ; Initial time that function  $P=at+b$  is valid
  TO ; Final time for the same function
  BY ; Sampling intervals, set to 1 minutes
  RATE ; Rate 'a' of the given function
  CONST) ; Constant 'b' of the given function
;
(LITERALIZE STATE ; State of the core which will be sampled
  POWER ; Current reactor power
  AXIAL_OFFSET ; Current axial offset of the core
  PERIOD ; Current reactor period
  TEMP_IN ; Current inlet temperature of the core
  TEMP_OUT ; Current outlet temperature of the core
  B_CONC ; Current boron concentration in ppm
  TIME) ; Current time

```

```

;
(LITERALIZE GOALS      ; Set of goals to satisfy
  GOAL)                ; Current goal of ACES (vector)
;
(LITERALIZE AOCNTRL   ; Axial-offset control parameters
  AO_RATE             ; Constant 'a' of target(AO) = a * P
  AO_SIGN             ; Flag, = TREND(current(AO),target(AO))
  AOE )              ; Flag, = AOTREND(current(AO),target(AO),2)
;
(LITERALIZE PWRCTRL   ; Power control parameters
  POWERI              ; Target power at current time
  POWERJ              ; Target power at next time step
  PSIGN               ; Flag, = TREND(current power, POWERI)
  TPSIGN              ; Flag, = TREND(POWERJ,POWERI)
  REACTIVITY          ; Absolute value of target reactivity
  RSIGN)              ; Sign of the target reactivity
;
(LITERALIZE TEMPERATURE ; WME for temperature control
  TEMP_REF            ; Average core temperature at full power
  TEMP_IN )           ; Proposed inlet temperature for the next step
;
(LITERALIZE CRODS     ; Temporary WMEs for control rod banks
  BANK                ; Name of the control rod bank
  POSITION              ; Current position in steps
  NUMBER              ; 1 for FLCR-A, 2 for FLCR-B... and 5 for PLCR
  PWORDTH              ; Differential reactivity if withdrawn one step
  NWORDTH              ; Differential reactivity if inserted one step
  MAX_POS              ; Uppermost possible position ( 200 )
  MIN_POS              ; Lowermost possible position ( 0 )
  PREF_DIR )          ; Preferred direction of motion
;
(LITERALIZE MOVEROD   ; Temporary variable for iteration in CR moves
  BANK                ; Name of the control rod bank to be moved
  DIRECTION            ; Direction of move, 1 upward, -1 downward
  REACTIVITY           ; Target reactivity to satisfy
  STEP                ; Current position of control rod bank
  LIMIT                ; Maximum steps that a given bank can be moved
  COUNT)              ; The number of steps that the bank moved
;
(LITERALIZE SENDCR    ; Temporary WME for data communication

```

```

POSTN)      ; Current positions of all control rod banks
;
; The start-up production, sets the runtime options of OPS5
;
(STARTUP
  (ENABLE HALT)
  (STRATEGY MEA)
  (MAKE START)
  (RUN))
;
;
; Rule set # 1: Initialization of WMEs at various phases of ACES
;
; Rule 1.1 : This rule performs the initialization of the reactor
;           model, queries the data file name of differential rod worth
;           knowledge base, and interface with user for power schedule.
;           It fires only once.
(p initialization
  { <go> (START)}
-->
  (CALL INIT)
  (WRITE | Please enter the file name for CR worths.. | (CRLF))
  (BIND <file> (ACCEPT))
  (OPENFILE DW <file> IN )
  (MAKE SENDCR )
  (CALL LOAD 100.0 0.0 )
  (MAKE GOALS ^goal read_data ))
;
; Rule 1.2 : This rule reads control rod worth data from input file
;           in the order of bank name, position, and worth, reads nil
;           for an empty line to mark the end of data, fires as much as
;           necessary for initialization of ACES active working memory.
(p read_differential_worths
  { <goal> (GOALS
    ^goal read_data )}
-->
  (MODIFY <goal> ^goal read_data ).
  (MAKE DWORTHS ^bank (ACCEPTLINE DW nil nil nil )))
;
; Rule 1.3 : Initializes boron worth and reference temperature

```

```

;      from the same data file, fires only once.
(p end_of_data
  { <goal> (GOALS
    ^goal read_data )}
  { <dwr> (DWORTHS
    ^bank nil )}
-->
(REMOVE <goal> <dwr> )
(MAKE BORON ^unit      (ACCEPT DW )
          ^worth      (ACCEPT DW )
          ^steps      0
          ^change     0
          ^direction 1 )
(MAKE TEMPERATURE ^temp_ref (ACCEPT DW ) )
;
; Rule 1.4 : Sets current reactivity worths of control rods,
;           and limits. Fires once for each FLCR bank when the FLCR
;           positions are sampled and whenever an FLCR is moved.
;           For withdrawals, positive reactivity is set to differential
;           worth at the current position of the bank, and
;           for insertions, negative reactivity is set to differential
;           worth at one step down of the current position.
(p set_crod_reactivity_worths_and_limits
  { <cr> (CRODS
    ^bank { <crbank> <> plcr }
    ^position <z>
    ^number <num>
    ^pworth nil )}
  (DWORTHS
    ^bank <crbank>
    ^step <z>
    ^worth <upw>)
  (DWORTHS
    ^bank <crbank>
    ^step ( COMPUTE <z> - 1 )
    ^worth <downw> )
  { <scr> (SENDCR )}
-->
(BIND <pos> (LITVAL POSTN ))
(BIND <post> (COMPUTE <pos> + <num> - 1 ))

```

```

(MODIFY <scr> ^<post> <z> )
(MODIFY <cr> ^pworth <upw>
          ^nworth <downw>
          ^max_pos 200
          ^min_pos 0 ))
;
; Rule 1.5 : After the sampling of the reactor, this rule
;           reinitializes all control parameters and flags used in
;           evaluation of the core status, and produce output
;           for current core status. It fires only once for each
;           control cycle.
;
(p initialize_control_variables
  (STATE
    ^power      <pr>
    ^period     <per>
    ^axial_offset <fao>
    ^temp_in    <tin>
    ^temp_out   <tout>
    ^b_conc     <ppm>
    ^time       <t> )
  (POWER
    ^from       <= <t>
    ^to         > <t>
    ^by         <step>
    ^rate       <a>
    ^const      <b> )
  { <aocntrl> (AOCNTRL
    ^ao_rate    <faor> )}
  -(PWRCNTRL)
  -(GOALS)
  { <plcr> (CRODS
    ^bank      plcr
    ^position   <zplcr> )}
  { <scr> (SENDCR )}
  { <temp> (TEMPERATURE
    ^temp_ref   <tav> )}
  -->
  (BIND <tnext> (COMPUTE <t> + <step> ))
  (BIND <pi>    (COMPUTE <b> + <a> * <t> ))
  (BIND <pj>    (COMPUTE <b> + <a> * <tnext> ))

```

```

(BIND <rs>      (COMPUTE 0.0848 // <per> ))
(BIND <re>      (RREAC <pj> <pr> <step> ))
(BIND <tr>      (COMPUTE <re> - <rs> ))
(BIND <dte>      (COMPUTE <tout> - <tin> ))
(BIND <tfao>     (COMPUTE <faor> * <pi> ))
(BIND <aob1>     (COMPUTE <tfao> + 5.0 ))
(BIND <aob2>     (COMPUTE <tfao> - 5.0 ))
(WRITE (CRLF) | Time (mins) : | <t> (TABTO 25)
        | Period (s) : | <per>
        (CRLF) | Power (%) : | <pr> (TABTO 25)
        | AO (%) : | <fao>
        (CRLF) | Targets (%) : | <pi> (TABTO 41) <tfao>
(CRLF) | Control Parameters : | (CRLF) | ----- |
(CRLF) | FLCR pos'ns | (TABTO 21) | PLCR pos'n | (TABTO 35)
        | Boron conc. (ppm) | (TABTO 57) | Inlet Temp. (C) |
(CRLF) (SUBSTR <scr> postn inf ) (TABTO 21) <zplcr>
(TABTO 35) <ppm> (TABTO 57) <tin> (CRLF) (CRLF) )
(CALL OUTPUT <t> <pr> <fao> <ppm> (SUBSTR <scr> postn inf )
        <zplcr> <pi> <tfao> <tin> <tout> <aob1> <aob2> )
(MODIFY <aocntrl> ^aoe (AOTREND <fao> <tfao> 2.0 )
        ^ao_sign (TREND <fao> <tfao> ))
(MODIFY <temp> ^temp_in (TINLET <tav> <dte> ))
(MAKE PWRCTRL ^poweri <pi>
        ^powerj <pj>
        ^reactivity (ABSOLUTE <tr>)
        ^psign (TREND <pr> <pi> )
        ^tpsign (TREND <pj> <pi> )
        ^rsign (TREND <tr> 0.0 ))
(MAKE GOALS ^goal ao_control ))
;
;
; Rule set # 2 : Axial-Offset control
;
; Rule 2.1 : Transfers the control to next step since the AO
; is within the control band and no control is necessary.
(p no_axial_offset_control_needed
  { <goal> (GOALS
    ^goal ao_control )}
  (AOCNTRL
    ^aoe 0 )

```

```

-->
  (MODIFY <goal>    ^goal    correct ))
;
; Rule 2.2 : If the core AO is out of the control band, this
;           rule moves the PLCR one step in the direction of AO error.
(p axial_offset_control
  { <goal> (GOALS
    ^goal    ao_control )}
  { <aoc> (AOCNTRL
    ^aoe    { <fl> <> 0 } )}
  { <cr> (CRODS
    ^bank    plcr
    ^position <z> )}
-->
  (MODIFY <cr>    ^position (COMPUTE <z> + <fl> ))
  (MODIFY <goal> ^goal    correct ))
;
;
; Rule set # 3 : Boron Control
;
; Rule 3.1 : This rule increases the amount of boron control
;           during the power transient by one unit before checking for
;           using boron at this step, and transfers the control to
;           boron control, may fire only once for each cycle during
;           a power transient.
(p modify_response_in_transient_1
  { <goal> (GOALS
    ^goal    correct )}
  { <br> (BORON
    ^steps    <s>
    ^direction <d> )}
  (AOCNTRL
    ^ao_sign  <err> )
  (PWRCNTRL
    ^tpsign   (COMPUTE <err> * <d> * -1 ))
-->
  (MODIFY <br>    ^steps (COMPUTE <s> + 1 ))
  (MODIFY <goal> ^goal    b_control ))
;
; Rule 3.2 : This rule decreases the amount of boron control

```

```

;      during the power transient by one unit before checking for
;      using boron at this step, and transfers the control to
;      boron control, may fire only once for each cycle during
;      a power transient.
(p modify_response_in_transient_2
  { <goal> (GOALS
    ^goal      correct )}
  { <br> (BORON
    ^steps     { <s> > 1 }
    ^direction <d> )}
  (AOCNTRL
    ^ao_sign   <err> )
  (PWRCNTRL
    ^tpsign    (COMPUTE <err> * <d> ))
-->
  (MODIFY <br>   ^steps (COMPUTE <s> - 1 ))
  (MODIFY <goal> ^goal  b_control ))
;
; Rule 3.3 : This rule sets the amount of boron control to zero
;      during the power transient by one unit before checking for
;      using boron at this step, and transfers the control to
;      boron control, may fire only once for each cycle during
;      a power transient.
(p set_response_to_zero
  { <goal> (GOALS
    ^goal      correct )}
  { <br> (BORON
    ^steps     > 0
    ^direction <d> )}
  (AOCNTRL
    ^ao_sign   <err>
    ^aoe       <err> )
  (PWRCNTRL
    ^tpsign    (COMPUTE <err> * <d> ))
-->
  (MODIFY <br>   ^steps 0)
  (MODIFY <goal> ^goal  b_control ))
;
; Rule 3.4 : This rule switches the direction of boron control
;      when its necessary, may fire only once for each cycle.

```

```

(p switch_the_direction
  { <goal> (GOALS
    ^goal      correct)}}
  { <br> (BORON
    ^steps     0
    ^direction <d> )}
  (AOCNTRL
    ^ao_sign   <err>
    ^aoe       <err> )
  (PWRCNTRL
    ^tpsign    (COMPUTE <err> * <d> ))
-->
  (MODIFY <br> ^direction (COMPUTE <d> * -1 ))
  (MODIFY <goal> ^goal b_control ))
;
; Rule 3.5 : This rule resets the direction of boron response
;           after the reactor reaches to steady state following a
;           transient, may fire only once in a cycle.
(p reset_direction
  { <goal> (GOALS
    ^goal      correct )}
  (PWRCNTRL
    ^tpsign    0 )
  { <br> (BORON
    ^direction <> 1 )}
-->
  (MODIFY <br> ^direction 1 )
  (MODIFY <goal> ^goal b_control ))
; Rule 3.6 : Transfers the control to boron control if no
;           change is necessary in the amount of boron control
(p no_change
  { <goal> (GOALS
    ^goal      correct )}
-->
  (MODIFY <goal> ^goal b_control ))
;
; Rule 3.7 : This rule changes the boron concentration
;           as much as 's' steps in the direction 'd' during
;           a power transient, modifies the target reactivity
;           and transfers the control to power control rules.

```

```

(p boron_control_in_transient
  { <goal> (GOALS
    ^goal      b_control )}
  { <pc> (PWRCTRL
    ^reactivity <r>
    ^tpsign     { <hao> <> 0 }
    ^rsign      <rs> )}
  { <br> (BORON
    ^worth      <wr>
    ^steps      <s>
    ^direction  <d> )}
-->
  (BIND <steps> (COMPUTE <s> * <hao> * <d> ))
  (BIND <rnew> (COMPUTE ( <r> * <rs> ) + <steps> * <wr> * -1 ))
  (MODIFY <br> ^change <steps> )
  (MODIFY <pc> ^reactivity (ABSOLUTE <rnew> )
    ^rsign (TREND <rnew> 0.0 ))
  (MODIFY <goal> ^goal set_preference move_crods send_info))
;
; Rule 3.8 : This rule determines the amount of boron control during
; the steady state operation, and transfers the control
; directly to the communication rules.
(p boron_control_in_steady_state
  { <goal> (GOALS
    ^goal b_control )}
  { <pr> (PWRCTRL
    ^tpsign 0
    ^reactivity <r>
    ^rsign <rs> )}
  { <br> (BORON
    ^worth { <wr> < <r> } )}
-->
  (BIND <st> (BORON <wr> <r> ))
  (MODIFY <br> ^change ( COMPUTE <st> * <rs> * -1 ))
  (MODIFY <goal> ^goal send_info ))
;
; Rule 3.9 : This rule transfers the control to power control
; during the steady state, since no control is required
(p no_boron_control
  { <goal> (GOALS

```

```

    ^goal  b_control)}}
-->
  (MODIFY <goal> ^goal set_preference move_crods send_info ))
;
;
; Rule set # 4 : Power control using flcrs
;
; Rule 4.1 : This rule finds the control rod bank which is at the
;             downmost position to introduce positive reactivity to the
;             system by withdrawal, and creates the necessary WMEs to
;             move it
(p providing_positive_reactivity
  { <goal> (GOALS
    ^goal      move_crods )}
  (PWRCNTRL
    ^reactivity <tr>
    ^rsign      1 )
  (CRODS
    ^bank       { <crbank> <> plcr }
    ^position   <z>
    ^pref_dir   1
    ^pworth     < <tr>
    ^max_pos    { <zmax> > <z> })
-->
  (BIND  <pos>      (LITVAL GOAL))
  (MODIFY <goal>   ^goal      move_cr (SUBSTR <goal> <pos> inf))
  (MAKE  MOVEROD   ^bank      <crbank>
    ^direction  1
    ^reactivity <tr>
    ^step       <z>
    ^limit      (COMPUTE <zmax> - <z> )
    ^count      0 ))
;
; Rule 4.2 : This rule finds the control rod bank which is at the
;             upmost position to provide negative reactivity by inserting
;             the rod, and creates the necessary WMEs to move it.
(p providing_negative_reactivity
  { <goal> (GOALS
    ^goal      move_crods )}
  ( PWRCNTRL

```

```

    ^reactivity <tr>
    ^rsign      -1 )
(CRODS
    ^bank      { <crbank> <> plcr }
    ^position  <z>
    ^pref_dir  -1
    ^nworth    < <tr>
    ^min_pos   { <zmin> < <z> }
-->
(BIND <pos> (LITVAL GOAL))
(MODIFY <goal> ^goal move_cr (SUBSTR <goal> <pos> inf))
(MAKE MOVEROD ^bank <crbank>
             ^direction -1
             ^reactivity <tr>
             ^step (COMPUTE <z> - 1 )
             ^limit (COMPUTE <z> - <zmin> )
             ^count 0 ))
;
; Managing the FLCR movements
;
; Rule 4.3 : This rule sets FLCR directional preference by
; comparing their relative positions.
(p preference
  { <goal> (GOALS
    ^goal set_preference )}
  { <cr1> (CRODS
    ^bank { <b1> <> plcr }
    ^position <z1> )}
  { <cr2> (CRODS
    ^bank { <b2> <> plcr <> <b1> }
    ^position { <z2> <= <z1> } )}
  { <cr3> (CRODS
    ^bank { <b3> <> plcr <> <b1> <> <b2> }
    ^position { <z3> <= <z2> } )}
  { <cr4> (CRODS
    ^bank { <b4> <> plcr <> <b1> <> <b2> <> <b3> }
    ^position <= <z3> )}
-->
(BIND <pos> (LITVAL GOAL))
(BIND <pos2> (COMPUTE <pos> + 1 ))

```

```

(MODIFY <goal> ^goal (SUBSTR <goal> <pos2> inf) nil )
(MODIFY <cr1> ^pref_dir -1 )
(MODIFY <cr2> ^pref_dir 0 )
(MODIFY <cr3> ^pref_dir 0 )
(MODIFY <cr4> ^pref_dir 1 ))
;
; Rule 4.4 : This rule moves the selected rod one step at a
; time up to total of 5 steps, as long as limitation
; are not exceeded.
(p moving_selected_control_rod
(GOALS
  ^goal      move_cr)
{ <movecr> ( MOVEROD
  ^bank      <crbank>
  ^direction <dir>
  ^reactivity <tr>
  ^step      <z>
  ^limit     { <lim> <> 0 }
  ^count     { <count> < 5 } )})
(DWORTHS
  ^bank      <crbank>
  ^step      <z>
  ^worth     { <dw> < <tr> } )
-->
(MODIFY <movecr> ^reactivity (COMPUTE <tr> - <dw>)
          ^step      (COMPUTE <z> + <dir>)
          ^limit     (COMPUTE <lim> - 1 )
          ^count     (COMPUTE <count> + 1 )))
;
; Rule 4.5 : This rule stops moving the selected control rod bank
; when the target reactivity is provided.
(p stop_moving_cr
{ <goal> (GOALS
  ^goal      move_cr )})
{ <pwrctrl> (PWRCTRL )})
{ <movecr> (MOVEROD
  ^bank      <crbank>
  ^direction <dir>
  ^reactivity <tr>
  ^step      <z>

```

```

    ^count      <count> )}
(DWORTHS
  ^bank        <crbank>
  ^step        <z>
  ^worth       { <w> > <tr> } )
{ <cr>        (CRODS
  ^bank        <crbank>
  ^position    <old_z> )}
-->
(MODIFY <goal>      ^goal set_preference )
(MODIFY <pwrctrl>  ^reactivity <tr> )
(MODIFY <cr>      ^position (COMPUTE <old_z> + <dir> * <count>)
                ^pworth  nil )
(REMOVE <movecr> )
;
; Rule 4.6 : This rule stops moving the selected bank since
;           the 5 steps limit is met. It returns the control to
;           power control.
;
(p insertion_limit_on_moving_cr
  { <goal> (GOALS
    ^goal      move_cr )}
  { <pwrctrl> (PWRCTRL )}
  { <movecr> (MOVEROD
    ^bank      <crbank>
    ^direction <dir>
    ^reactivity <tr>
    ^count     5 )}
  { <cr>      (CRODS
    ^bank      <crbank>
    ^position  <old_z> )}
-->
(MODIFY <goal>      ^goal      set_preference )
(MODIFY <pwrctrl>  ^reactivity <tr>)
(MODIFY <cr>      ^position  (COMPUTE <old_z> + <dir> * 5 )
                ^pworth    nil
                ^nworth    nil )
(REMOVE <movecr> )
;
; Rule 4.7 : This rule stops moving the selected control rod bank
;           when the upper or lower limit is reached.

```

```

(p physical_limit_on_moving_cr
  { <goal>      (GOALS
    ^goal      move_cr )}
  { <pwrctrl> (PWRCTRL )}
  { <movecr>   (MOVEROD
    ^bank      <crbank>
    ^direction <dir>
    ^reactivity <tr>
    ^count     <count>
    ^limit     0 )}
  { <cr>       (CRODS
    ^bank      <crbank>
    ^position  <old_z> )}
-->
  (MODIFY <goal>      ^goal      set_preference )
  (MODIFY <pwrctrl> ^reactivity <tr>)
  (MODIFY <cr>       ^position (COMPUTE <old_z> + <dir> * <count>)
    ^pworth      nil
    ^nworth      nil )
  (REMOVE <movecr> ))
;
; Rule 5.8 : This rule transfers the control to communication
;           rules when no more FLCr moves are necessary.
(p end_of_cr_moves
  { <goal> (GOALS
    ^goal  move_crods )}
  -(MOVEROD)
-->
  (BIND <pos>      (LITVAL GOAL))
  (BIND <pos2>     (COMPUTE <pos> + 1 ))
  (MODIFY <goal>   ^goal (SUBSTR <goal> <pos2> inf) nil ))
;
; Rule set # 5: Send the proposed reactivity changes to the
;               simulator, or communicate with the Reactor
;
; Rule 5.1 : This rule removes the control rod WMEs in order
;           to be ready for the next cycle.
(p clean_old_crod_wmes
  (GOALS
    ^goal  send_info )

```

```

{ <cr> (CRODS
  ^bank { <crbank> <> plcr } )}
-->
(REMOVE <cr>))
;
; Rule 5.2 : This rule wraps up the current state, sends data
; to REACTOR.FOR and creates new state for next cycle of
; the control.
(p calling_reactor_for
  { <goal> (GOALS
    ^goal send_info )}
  { <pwrctrl> (PWRCTRL )}
  { <send> (SENDCR )}
  { <bor> (BORON
    ^change <dppm> )}
  { <plcr> (CRODS
    ^bank plcr
    ^position <zpl> )}
  -(CRODS
    ^bank <> plcr )
  { <st> (STATE )}
  (TEMPERATURE
    ^temp_in <tin> )
-->
(CALL REACTOR (SUBSTR <send> postn inf ) <zpl> <dppm> <tin> )
(MODIFY <bor> ^change 0 )
(REMOVE <pwrctrl> <st> <goal> <plcr> ))
;
; Rule 5.3 : This rule is a garbage collection rule. Removes
; WME that defines the old power schedule.
(p remove_power_history_data
  (STATE
    ^time <t> )
  { <pow> (POWER
    ^to <=<t> )}
-->
(REMOVE <pow> ))

```

APPENDIX B: SUPPORT ROUTINES OF ACES

ACES is furnished with a set of FORTRAN77 routine to perform numerical tasks, and user interface. Therefore, these routines are a part of ACES. We listed the source codes of these external routines of ACES in this section.

```

C
C The function RREAC will calculate the necessary reactivity to
C satisfy the next target power, takes the current power, target
C power, and the time step size as arguments, and
C returns the reactivity.
C
      INTEGER FUNCTION RREAC (ATOM1,ATOM2,ATOM3)
      INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
      REAL ATOM1,ATOM2,ATOM3,TPR2,PR,DT,TP
C
      DT=OPS$CVAF (%VAL (ATOM3))
      DT=DT*60.0
      TPR2=OPS$CVAF (%VAL (ATOM1))
      PR=OPS$CVAF (%VAL (ATOM2))
      IF (PR.EQ.TPR2) THEN
        TP=0.0
      ELSE
        TP=DT/LOG(TPR2/PR)
        TP=0.0848/TP
      ENDIF
      RREAC=OPS$CVFA(%VAL (TP))
      RETURN
      END
C

```

C The function TREND compares its arguments and returns
 C 1 if first argument is greater, 0 if they are equal, and
 C -1 if the second argument is greater.

C

```

    INTEGER FUNCTION TREND (ATOM1,ATOM2)
    INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
    REAL*4 ATOM1,ATOM2,A1,A2
    INTEGER I
  
```

C

```

    A1=OPS$CVAF (%VAL (ATOM1))
    A2=OPS$CVAF (%VAL (ATOM2))
    IF (A1.GT.A2) THEN
      I=1
    ELSEIF (A1.EQ.A2) THEN
      I=0
    ELSE
      I=-1
    ENDIF
    TREND=OPS$CVNA (%VAL (I))
    RETURN
  END
  
```

C

C The function AOTREND returns 1 if the first argument is higher
 C than the upper limit of control band defined by the second and
 C third argument, 0 if it is within the band, and -1 otherwise

C

```

    INTEGER FUNCTION AOTREND (ATOM1,ATOM2,ATOM3)
    INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
    REAL*4 ATOM1,ATOM2,ATOM3,A1,A2,A3
    INTEGER I
  
```

C

```

    A1=OPS$CVAF (%VAL (ATOM1))
    A2=OPS$CVAF (%VAL (ATOM2))
    A3=OPS$CVAF (%VAL (ATOM3))
    IF (A1.LT.(A2-A3)) THEN
      I=-1
    ELSEIF (A1.LE.(A2+A3)) THEN
      I=0
    ELSE
      I=1
  
```

```

ENDIF
AOTREND=OPS$CVNA (%VAL (I))
RETURN
END

```

```

C
C The function ABSOLUTE simply returns the absolute value of
C its argument
C

```

```

INTEGER FUNCTION ABSOLUTE(ATOM1)
INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
REAL*4 ATOM1,T

```

```

C
T=OPS$CVAF(%VAL (ATOM1))
T=ABS(T)
ABSOLUTE=OPS$CVFA(%VAL (T))
RETURN
END

```

```

C
C BORON takes the target reactivity and the reactivity worth of boron
C as its arguments and returns the necessary boron control in steps
C

```

```

INTEGER FUNCTION BORON (ATOM1,ATOM2)
INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
REAL*4 ATOM1,ATOM2,A1,A2,A3
INTEGER I

```

```

C
A1=OPS$CVAF (%VAL (ATOM1 ))
A2=OPS$CVAF (%VAL (ATOM2 ))
I=A2/A1
BORON=OPS$CVNA (%VAL (I))
RETURN
END

```

```

C
C TINLET calculates the proposed inlet temperature to keep the
C core average temperature constant
C

```

```

INTEGER FUNCTION TINLET(ATOM1,ATOM2)
INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
REAL*4 ATOM1,ATOM2,A1,A2,T

```

```

C

```

```

A1=OPS$CVAF (%VAL (ATOM1 ))
A2=OPS$CVAF (%VAL (ATOM2 ))
T=A1-A2/2.0
TINLET=OPS$CVFA (%VAL (T ))
RETURN
END

```

```

C
C Subroutine LOAD is the user interface routine of ACES. It asks
C the user for the power schedule for an upcoming period daily cycle
C

```

```

SUBROUTINE LOAD
IMPLICIT REAL*4 (A-H,O-Z)
INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'

```

```

C
C Setting constants for creating a WME later

```

```

INP1=OPS$PARAMETER (%VAL (1 ))
INP2=OPS$PARAMETER (%VAL (2 ))
POWER=OPS$CVAF (%VAL (INP1 ))
T1=OPS$CVAF (%VAL (INP2 ))
IP=OPS$INTERN(%REF ('POWER' ), %VAL (5 ))
IA1=OPS$INTERN (%REF ('FROM' ), %VAL (4 ))
IA2=OPS$INTERN (%REF ('TO' ), %VAL (2 ))
IA5=OPS$INTERN (%REF ('BY' ), %VAL (2 ))
IA3=OPS$INTERN (%REF ('RATE' ), %VAL (4 ))
IA4=OPS$INTERN (%REF ('CONST' ), %VAL (5 ))

```

```

C
HT=1.d0
IS=OPS$CVFA (%VAL (HT ))
WRITE(*,5) POWER,T1

```

```

C
C Start asking the user for information
PRINT*, ' Begin entering load cycle per day '
1 PRINT*, ' Choose one of the options:'
PRINT*, ' 1. Steady State'
PRINT*, ' 2. Decrease Power'
PRINT*, ' 3. Increase Power'
PRINT*, ' 4. End of Data'
READ*, IOPTION
IF (IOPTION.EQ.4) GO TO 2
CALL OPS$RESET()

```

```

CALL OPS$VALUE (%VAL (IP ))
CALL OPS$TAB (%VAL (IA1 ))
IT1=OPS$CVFA (%VAL (T1 ))
CALL OPS$VALUE (%VAL (IT1 ))
IF (IOPTION.EQ.2) PRINT*, ' Decrease to ( % power ): '
IF (IOPTION.EQ.3) PRINT*, ' Increase to ( % power ): '
IF (IOPTION.NE.1) READ*,POWERJ
PRINT*, ' Enter time interval in minutes '
READ*,IT
DT=FLOAT(IT)
T2=T1+DT
CALL OPS$TAB (%VAL (IA2 ))
IT2=OPS$CVFA (%VAL (T2 ))
CALL OPS$VALUE (%VAL (IT2 ))
CALL OPS$TAB (%VAL (IA5 ))
CALL OPS$VALUE (%VAL (IS ))

```

C

C For given time interval, create a function for power as a
C function of time, $P(t)=aP(0)+b$

```

IF (IOPTION.NE.1) THEN
    RATE=(POWERJ-POWER)/DT
    CONST=POWER-RATE*T1
    POWER=POWERJ
ELSE
    RATE=0.0
    CONST=POWER
ENDIF
T1=T2

```

C

C Upload the information by creating a WME in the active working
C memory of ACES

```

CALL OPS$TAB (%VAL (IA3 ))
IR=OPS$CVFA (%VAL (RATE ))
CALL OPS$VALUE (%VAL (IR ))
CALL OPS$TAB (%VAL (IA4 ))
IC=OPS$CVFA (%VAL (CONST ))
CALL OPS$VALUE (%VAL (IC ))
CALL OPS$ASSERT()
GO TO 1

```

5 FORMAT(2X, 'Current power is ',F8.4,' % at time ',

```
2      +      F8.2, ' mins')  
      RETURN  
      END
```

APPENDIX C: SUPPORT ROUTINES OF REACTOR MODEL

The routines introduced in this section are related to the reactor model and has no direct effects on the decision process of ACES. However, the subroutine REACTOR receives information from ACES related to proposed control, and sends the current status after a minute long time step back to ACES by creating a WME. This process is supposed to be replaced with a monitoring rule in actual applications.

```

C
C One-group one-dimensional PWR core model
  SUBROUTINE REACTOR
C
  IMPLICIT REAL*8 (A-H,O-Z)
  INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
C
  PARAMETER (MX=200)
  DOUBLE PRECISION A(MX),B(MX),XXE(MX),XI(MX),T(MX),TR(MX)
  DOUBLE PRECISION SIGMAC(MX),FLUX(MX),FLUXO(MX),C(MX),AF(MX)
  DOUBLE PRECISION CRBANK(5),OUTS(10),KOLD,NU,NW
  REAL*4 AOS,PPOW,PER,BORON,TIMES,TT,TO
  INTEGER IPOS(5),IPOSC(5)
C
C Restore the parameters sent from ACES related to control
  INP1=OPS$PARAMETER (%VAL (1 ))
  INP2=OPS$PARAMETER (%VAL (2 ))
  INP3=OPS$PARAMETER (%VAL (3 ))
  INP4=OPS$PARAMETER (%VAL (4 ))
  INP5=OPS$PARAMETER (%VAL (5 ))

```

```

INP6=OPS$PARAMETER (%VAL (6 ))
INP7=OPS$PARAMETER (%VAL (7 ))
IPOSC(1)=OPS$CVAN (%VAL (INP1 ))
IPOSC(2)=OPS$CVAN (%VAL (INP2 ))
IPOSC(3)=OPS$CVAN (%VAL (INP3 ))
IPOSC(4)=OPS$CVAN (%VAL (INP4 ))
IPOSC(5)=OPS$CVAN (%VAL (INP5 ))
IBOR=OPS$CVAN (%VAL (INP6 ))
TIN=DBLE(OPS$CVAF (%VAL (INP7 )))

```

C

C Restore the state from the data file if the variables are not
C initialized yet

```

IF(TIME.EQ.0.0) THEN
  OPEN(1,NAME='STE',STATUS='UNKNOWN')
  OPEN(2,NAME='TRS',STATUS='UNKNOWN')
  READ(1,*) MS,KOLD,BR2
  READ(1,*) THPOW,HEIGHT,HZ
  READ(1,*) D,NU,SIGMAF
  READ(1,*) SIGMAA,TAV,HT
  READ(1,*) NW,CB,SIGMIB
  READ(1,*) YIELDX,DECAYX,SIGMIX
  READ(1,*) YIELDI,DECAYI,DECAYC
  READ(1,*) CRBANK,SFNU,AMNUSF,DB,OPF
  READ(1,*) PCOEF,TCOEF,CON
  READ(1,*) BETA,DT
  READ(1,*) TR
  READ(2,*) POWER,BOR,TIME,K
  READ(2,*) FLUXO,XXE,XI,C
  READ(2,*) IPOS,SIGMAC
  CLOSE(1)
ENDIF

```

C

```

DO 9 I=1,MS
9   AF(I)=FLUXO(I)
  POLD=POWER
  IFLAG=0
  TIMEOUT=TIME+HT

```

C

C Perform the control actions proposed by ACES
DO 6 I=1,5

```

6      IPOSC(I)=IPOSC(I)-IPOS(I)
      IBS=SIGN(1,IBOR)
C
C Boron concentration change is limited by 1 unit per second
1      IF (IBOR.NE.0) THEN
          BOR=BOR+DBLE(IBS)*DB
          IBOR=IBOR-IBS*1
      ENDIF
C
C FLCRs will be moved one step per second to simulate the
C reactivity insertion rates. Therefore, there will be a limit of
C 60 steps per minute.
      IF (IFLAG.EQ.0) CALL MOVECR(SIGMAC,CRBANK,IFLAG,IPOS,IPOSC)
C
      COEF1=2*D/HZ**2+SIGMIB*CB*BOR*NW+D*BR2+SIGMAA
C
C Generation of tri-diagonal finite difference matrix for
C diffusion equation. Off-diagonal elements are all constant and
C equal to parameter CON. Therefore, The matrix is stored in
C a vector A as diagonal elements.
      TIME=TIME+DT
      TSUM=0.d0
      DO 10 I=1,MS-1
          T(I)=TCOEF*(HZ*TSUM+FLUXO(I)*HZ/2.d0)+TIN
          A(I)=COEF1+SIGMAC(I)+AMNUSF*(T(I)-TR(I))
          +
            +SIGMIX*XKE(I)-(1.d0-BETA)*SFNU
          C(I)=C1*C(I)+C2*FLUXO(I)
          B(I)=DECAYC*C(I)
10      TSUM=TSUM+FLUXO(I)
      T(MS)=TCOEF*HZ*TSUM+TIN
C
C Solution of the system at next time step by forward Gaussian
C elimination, and backward substitution
      DO 15 I=2,MS-1
          COEF=CON/A(I-1)
          A(I)=A(I)-COEF*CON
15      B(I)=B(I)-COEF*B(I-1)
      FLUX(MS-1)=B(MS-1)/A(MS-1)
      SUM1=FLUX(MS-1)
      DO 20 I=MS-2,1,-1

```

```

                FLUX(I)=(B(I)-CON*FLUX(I+1))/A(I)
20             SUM1=SUM1+FLUX(I)
C
C Calculate new power
                POWER=PCOEF*SIGMAF*HZ*SUM1
                DPDT=(POWER-POLD)/DT
                POLD=POWER
                IF (TIME.LT.TIMEOUT) THEN
                    DO 35 I=1,MS
35                 FLUXO(I)=FLUX(I)
                    GO TO 1
                ENDIF
C
                DO 2 I=1,MS-1
                    XXE(I)=(YIELDX*SIGMAF*FLUXO(I)+DECAYI*XI(I)-DECAYX*XXE(I)
+                     -SIGMIX*XXE(I)*FLUXO(I))*HT+XXE(I)
2                   XI(I)=(YIELDI*SIGMAF*FLUXO(I)-DECAYI*XI(I))*HT+XI(I)
C
C Calculating the reactor parameters for ACES
                P1=FLUX(100)*HZ/2.d0
                P2=P1
                DO 110 I=1,99
                    P1=P1+HZ*FLUX(I)
110                 P2=P2+HZ*FLUX(I+100)
                AO=(P2-P1)/(P1+P2)*100d0
                AOS=SNGL(AO)
                PPOW=SNGL(POWER/THPOW*100.0)
                PER=SNGL(POWER/DPDT)
                TIMES=SNGL(TIMEOUT/60.d0)
                BORON=SNGL(BOR)
                TI=SNGL(TIN)
                TO=SNGL(T(MS))
C
C Sending the reactor parameters to ACES by creating WMEs
                CALL CREATE(PPOW,AOS,PER,BORON,TIMES,TI,TO,IPOS)
C
                TOUT=TIMEOUT/60.DO
                OPT=OPF*DBLE(K)
C
C Output generation for graphics

```

```

IF (TOUT.GE.OPT) THEN
  CALL OUTFL(FLUX,XXE)
  K=K+1
ENDIF
RETURN
END

```

C

```

C This routine moves FLCRs by one step if it is proposed by ACES
SUBROUTINE MOVECR (SIGMAC,CRBANK,IFLAG,IPOS,IPOSC)
DOUBLE PRECISION SIGMAC(200),CRBANK(5)
INTEGER IPOS(5),IPOSC(5)

```

C

```

KOUNT=0
DO 1 J=1,5
  IF (IPOSC(J).NE.0) THEN
    IS=SIGN(1,IPOSC(J))
    IL=IPOS(J)+IS
    IF (IS.GT.0) IL=IL-1
    IF (J.EQ.5) THEN
      I1=IPOS(J)-IS
      I2=I1+50
      SIGMAC(IL)=SIGMAC(I1)
      SIGMAC(IL+50)=SIGMAC(I2)
    ELSE
      SIGMAC(IL)=SIGMAC(IL)-IS*CRBANK(J)
    ENDIF
    IPOSC(J)=IPOSC(J)-IS
    IPOS(J)=IPOS(J)+IS
  ELSE
    KOUNT=KOUNT+1
  ENDIF
1 CONTINUE
IF(KOUNT.EQ.5) IFLAG=1
RETURN
END

```

C

```

C This routine creates WMEs of STATUS and CRODS for ACES
C It translates the data into OPS5 atoms and assigns their
C values to attributes of related WMEs.
SUBROUTINE CREATE (POWER,AO,PERIOD,BOR,TIME,TI,TO,IPOS)

```

```

IMPLICIT REAL*4 (A-H,O-Z)
INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'

```

C

```

INTEGER IPOS(5)
CHARACTER*5 BANK(4)

```

C

```

DATA BANK/'FLCRA','FLCRB','FLCRC','FLCRD'/

```

C

```

CALL OPS$RESET ( )
IST=OPS$INTERN (%REF ('STATE' ), %VAL (5 ))
CALL OPS$VALUE (%VAL (IST ))
IA1=OPS$INTERN (%REF ('POWER' ), %VAL (5 ))
IA2=OPS$INTERN (%REF ('AXIAL_OFFSET' ), %VAL (12 ))
IA3=OPS$INTERN (%REF ('PERIOD' ), %VAL (6 ))
IA4=OPS$INTERN (%REF ('TIME' ), %VAL (4 ))
IA5=OPS$INTERN (%REF ('TEMP_IN' ), %VAL (7 ))
IA6=OPS$INTERN (%REF ('TEMP_OUT' ), %VAL (8 ))
IA8=OPS$INTERN (%REF ('B_CONC' ), %VAL (6 ))

```

C

C Creating the STATUS

```

CALL OPS$TAB (%VAL (IA1 ))
IP=OPS$CVFA (%VAL (POWER ))
CALL OPS$VALUE (%VAL (IP ))

```

C

```

CALL OPS$TAB (%VAL (IA2 ))
IA0=OPS$CVFA (%VAL (AO ))
CALL OPS$VALUE (%VAL (IA0 ))

```

C

```

CALL OPS$TAB (%VAL (IA3 ))
IPR=OPS$CVFA (%VAL (PERIOD ))
CALL OPS$VALUE (%VAL (IPR ))

```

C

```

CALL OPS$TAB (%VAL (IA4 ))
IT=OPS$CVFA (%VAL (TIME ))
CALL OPS$VALUE (%VAL (IT ))

```

C

```

CALL OPS$TAB (%VAL (IA5 ))
IT=OPS$CVFA (%VAL (TI ))
CALL OPS$VALUE (%VAL (IT ))

```

C

```

CALL OPS$TAB (%VAL (IA6 ))
IT=OPS$CVFA (%VAL (TO ))
CALL OPS$VALUE (%VAL (IT ))
C
CALL OPS$TAB (%VAL (IA8 ))
IB=OPS$CVFA (%VAL (BOR ))
CALL OPS$VALUE (%VAL (IB ))
CALL OPS$ASSERT()
C
IST=OPS$INTERN (%REF ('CRODS' ), %VAL (5 ))
IA1=OPS$INTERN (%REF ('BANK' ), %VAL (4 ))
IA2=OPS$INTERN (%REF ('POSITION' ), %VAL (8 ))
IA3=OPS$INTERN (%REF ('NUMBER' ), %VAL (6 ))
C
C Creating CRODS for each FLCR banks
DO 10 I=1,4
CALL OPS$RESET ()
CALL OPS$VALUE (%VAL (IST ))
CALL OPS$TAB (%VAL (IA1 ))
IP=OPS$INTERN (%REF (BANK(I) ), %VAL (5 ))
CALL OPS$VALUE (%VAL (IP ))
C
CALL OPS$TAB (%VAL (IA2 ))
IPO=OPS$CVNA (%VAL (IPOS(I) ))
CALL OPS$VALUE (%VAL (IPO ))
C
CALL OPS$TAB (%VAL (IA3 ))
INUM=OPS$CVNA (%VAL (I ))
CALL OPS$VALUE (%VAL (INUM ))
C
CALL OPS$ASSERT()
10 CONTINUE
C
C Creating CRODS for PLCR bank
CALL OPS$RESET ()
CALL OPS$VALUE (%VAL (IST ))
CALL OPS$TAB (%VAL (IA1 ))
IP=OPS$INTERN (%REF ('PLCR' ), %VAL (4 ))
CALL OPS$VALUE (%VAL (IP ))
C

```

```

CALL OPS$TAB (%VAL (IA2 ))
IPO=OPS$CVNA (%VAL (IPOS(5) ))
CALL OPS$VALUE (%VAL (IPO ))

```

C

```

CALL OPS$TAB (%VAL (IA3 ))
INUM=OPS$CVNA (%VAL (5 ))
CALL OPS$VALUE (%VAL (INUM ))

```

C

```
CALL OPS$ASSERT()
```

C

```

RETURN
END

```

C

```

C This routine is called by ACES directly, and initialize the
C subroutine REACTOR. It also initializes the STATUS and CRODS
C

```

C

```

SUBROUTINE INIT
IMPLICIT REAL*8 (A-H,O-Z)
INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'

```

C

```

PARAMETER (MX=200)
DOUBLE PRECISION XXE(MX),XI(MX),TR(MX)
DOUBLE PRECISION SIGMAC(MX),FLUXO(MX),C(MX)
DOUBLE PRECISION CRBANK(5),KOLD,NU,NW
REAL*4 POW,AOF,PER,BORON,TIME,TI,TO,AOD
INTEGER IPOS(5)
CHARACTER*6 FILE1

```

C

```

PRINT*,' Enter the steady state file name '
READ*,FILE1
OPEN(1,NAME=FILE1,STATUS='UNKNOWN')
OPEN(2,NAME='STE',STATUS='UNKNOWN')
OPEN(3,NAME='TRS',STATUS='UNKNOWN')

```

C

```

READ(1,*) MS,KOLD,BR2
READ(1,*) POWER,HEIGHT,BOR
READ(1,*) D,NU,SIGMAF
READ(1,*) SIGMAA,TIN,SIGMAC
READ(1,*) NW,CB,SIGMIB
READ(1,*) YIELDX,DECAYX,SIGMIX

```

```

READ(1,*) YIELDI,DECAYI,IPOS
READ(1,*) CRBANK,SFNU,AMNUSF,BETA,DECAYC,TAV
READ(1,*) PCOEF,TCOEF,CON,AO,AO2
READ(1,*) FLUXO,XXE,XI,C
READ(1,*) HZ,HT,TIM
READ(1,*) TR
READ(1,*) DB
CLOSE(1)

```

C

```

DT=1d0
OPF=60.d0
POW=SNGL(POWER/POWER*100.0)
AOF=SNGL(AO)
PER=SNGL(0.0848*KOLD/(KOLD-1.d0))
TIME=0.0
BORON=SNGL(BOR)
TI=SNGL(TIN)
TO=SNGL(2.d0*TAV-TIN)
CALL CREATE(POW,AOF,PER,BORON,TIME,TI,TO,IPOS)

```

C

```

CALL OPS$RESET ( )
IST=OPS$INTERN (%REF ('AOCNTRL' ), %VAL (7 ))
CALL OPS$VALUE (%VAL (IST ))
IA2=OPS$INTERN (%REF ('AO_RATE' ), %VAL (7 ))
AOF=AOF/100.0
CALL OPS$TAB (%VAL (IA2 ))
IAO=OPS$CVFA (%VAL (AOF ))
CALL OPS$VALUE (%VAL (IAO ))
CALL OPS$ASSERT ( )
K=1

```

C

```

WRITE(2,*) MS,KOLD,BR2
WRITE(2,*) POWER,HEIGHT,HZ
WRITE(2,*) D,NU,SIGMAF
WRITE(2,*) SIGMAA,TAV,HT
WRITE(2,*) NW,CB,SIGMIB
WRITE(2,*) YIELDX,DECAYX,SIGMIX
WRITE(2,*) YIELDI,DECAYI,DECAYC
WRITE(2,*) CRBANK,SFNU,AMNUSF,DB,OPF
WRITE(2,*) PCOEF,TCOEF,CON

```

```

WRITE(2,*) BETA,DT
WRITE(2,*) TR
WRITE(3,*) POWER,BOR,TIM,K
WRITE(3,*) FLUXO,XXE,XI,C
WRITE(3,*) IPOS,SIGMAC
CLOSE(2)
CLOSE(3)
RETURN
END

```

```

C
C Generates outputs of the flux and xenon distribution
SUBROUTINE OUTFL(FLUX,XE)

```

```

C
IMPLICIT REAL*8 (A-H,O-Z)

```

```

C
PARAMETER (MX=200)
DOUBLE PRECISION XE(MX),FLUX(MX)
CHARACTER*10 FILE

```

```

C
PRINT*,' Do you want output (1=yes/0=no) '
READ*,IRESP
IF (IRESP.NE.1) RETURN
PRINT*,' Enter the file name '
READ*,FILE
Z=0.DO
OPEN(3,NAME=FILE,STATUS='UNKNOWN')
WRITE(3,10) Z,Z,Z
DO 35 I=1,MX
  Z=Z+.005
35  WRITE(3,10) Z,FLUX(I),XE(I)
10  FORMAT(1X,F7.4,2(2X,E15.7))
CLOSE(3)
RETURN
END

```

```

C
C Subroutine OUTPUT is called from ACES to create
C a set of output for graphics purposes,
C and saves the current information in a file. It has been used
C for practical purposes only, and it does not contain any on-line
C graphics routine.

```

SUBROUTINE OUTPUT

C

```
IMPLICIT REAL*4 (A-H,O-Z)
INCLUDE 'OPS$LIBRARY:OPSDEF.FOR'
```

C

C Restoring the data sent from ACES

```
INP1=OPS$PARAMETER (%VAL (1 ))
INP2=OPS$PARAMETER (%VAL (2 ))
INP3=OPS$PARAMETER (%VAL (3 ))
INP4=OPS$PARAMETER (%VAL (4 ))
INP5=OPS$PARAMETER (%VAL (5 ))
INP6=OPS$PARAMETER (%VAL (6 ))
INP7=OPS$PARAMETER (%VAL (7 ))
INP8=OPS$PARAMETER (%VAL (8 ))
INP9=OPS$PARAMETER (%VAL (9 ))
INP10=OPS$PARAMETER (%VAL (10 ))
INP11=OPS$PARAMETER (%VAL (11 ))
INP12=OPS$PARAMETER (%VAL (12 ))
INP13=OPS$PARAMETER (%VAL (13 ))
INP14=OPS$PARAMETER (%VAL (14 ))
INP15=OPS$PARAMETER (%VAL (15 ))
TIME=OPS$CVAF (%VAL (INP1 ))
POW=OPS$CVAF (%VAL (INP2 ))
AO=OPS$CVAF (%VAL (INP3 ))
B=OPS$CVAF (%VAL (INP4 ))
I1=OPS$CVAN (%VAL (INP5 ))
I2=OPS$CVAN (%VAL (INP6 ))
I3=OPS$CVAN (%VAL (INP7 ))
I4=OPS$CVAN (%VAL (INP8 ))
I5=OPS$CVAN (%VAL (INP9 ))
TPOW=OPS$CVAF (%VAL (INP10 ))
TAO=OPS$CVAF (%VAL (INP11 ))
TIN=OPS$CVAF (%VAL (INP12 ))
TOUT=OPS$CVAF (%VAL (INP13 ))
AOU=OPS$CVAF (%VAL (INP14 ))
AOL=OPS$CVAF (%VAL (INP15 ))
TIME=TIME/60.0
OPEN(UNIT=1,NAME='STATE',STATUS='UNKNOWN',ACCESS='APPEND')
OPEN(UNIT=2,NAME='SAUX',STATUS='UNKNOWN',ACCESS='APPEND')
WRITE(1,1) TIME,POW,AO,B,I1,I2,I3,I4,I5
```

```
WRITE(2,2) TIME,TPOW,TAO,AOU,AOL,TIN  
CLOSE(1)  
CLOSE(2)  
1  FORMAT(1X,4(F10.5,1X),5I4)  
2  FORMAT(1X,6(F10.5,1X))  
RETURN  
END
```

APPENDIX D: STEADY.FOR

The initial conditions of the REACTOR.FOR will be determined by the steady state solution of the system. STEADY.FOR solves the nonlinear eigenvalue problem with power method, and iterates on the boron concentration to achieve the criticality. It writes the initial conditions into a user defined file.

```

IMPLICIT REAL*8 (A-H,O-Z)
PARAMETER (MX=200)
DOUBLE PRECISION A(MX),SOURCE(MX),XXE(MX),XI(MX),T(MX)
DOUBLE PRECISION SIGMC(4),FLUX(MX),FLUXO(MX),C(MX),BES(5)
DOUBLE PRECISION SIGMAC(MX),CRBANK(5),KOLD,KNEW,NU,NWO
INTEGER IPOS(5)
CHARACTER*15 FOUT

```

C

C Core composition, and basic data

```

DATA THPOW,GC,HEIGHT,RADIUS/3.4d9,3.2d-11,3.7d2,1.7d2/
DATA D,NU,SIGMAF,SIGMAA/1.2d0,2.418d0,.06617d0,.1285D0/
DATA NWO,CB,SIGMIB/2.41d22,.33161d-6,3.838d-21/
DATA YIELDX,DECAYX/.228d-2,.20917d-4/
DATA YELDI,DECAYI/.06386d0,.2875d-4/
DATA XMDOT,CP/1.569444d4,6.06d3/
DATA BETA,DECAYC/.65d-2,.767d-1/
DATA IPOS/70,120,180,201,0/
DATA RATIO,BESJ1/.63182d0,.5395077d0/

```

C

```

EXTRAD=176.34d0
MS=200

```

```

H=HEIGHT/DBLE(MS)
BR=2.405d0/EXTRAD
BR2=BR**2
PI=4.d0*DATAN(1.d0)
BES(1)=8.d0*.40356d0
BES(2)=4.d0*.69078d0
BES(3)=4.d0*(.92834d0+.21534d0)
BES(4)=1.d0+8.d0*.3574d0
BES(5)=8.d0*.64014d0
CRC=1.8636d0/(2.d0*PI*RADIUS*BESJ1/BR)
PCOEF=GC*PI*RADIUS**2
ALPHAM=1.d-4
AMNUSF=ALPHAM*NU*SIGMAF
TIN=300.d0

```

C

C Initial guess for flux distribution : SINE function

```

FMAX=THPOW/(PCOEF*SIGMAF*2.d0*HEIGHT/PI)
Z=0.d0
CCOEF=BETA*NU*SIGMAF/DECAYC
TCOEF=SIGMAF*PCOEF/XMDOT/CP
TSUM=0.d0
DO 1 I=1,MS
  Z=Z+H
  FLUXO(I)=FMAX*DSIN(PI*Z/HEIGHT)
  T(I)=TCOEF*(H*TSUM+FLUXO(I)*H/2.d0)+TIN
1  TSUM=TSUM+FLUXO(I)
T(MS)=TCOEF*H*TSUM+TIN

```

C

```

BOR=9.d2
SIGMIX=1.2318943d-18*RATIO
CON=-D/H**2
COEF2=(YIELDI+YIELDX)*SIGMAF
AR=5.d0
RAT=4.DO
CRSUM=0.d0
DO 33 I=1,4
33  CRSUM=CRSUM+BES(I)
SIGMC(4)=AR*CRC*CRSUM
SIGMC(2)=AR*CRC*BES(5)/RAT
EPSK=1.D-8

```

```

      EPSF=1.d-7
      SIGMC(1)=0.d0
      SIGMC(3)=0.d0
      KOLD=1.d0
      ITER=0
C
C Outer iteration on boron concentration
2     COEF1=2*D/H**2+SIGMIB*CB*BOR*NWO+D*BR2+SIGMAA
C
C Inner iteration on effective multiplication factor
5     ITER=ITER+1
      J=1
      DO 10 I=1,MS-1
          IF (I.GE.IPOS(J)) J=J+1
          A(I)=COEF1+SIGMC(J)+SIGMIX*COEF2*FLUXO(I)/
+          (DECAYX+SIGMIX*FLUXO(I))
10     SOURCE(I)=NU*SIGMAF*FLUXO(I)/KOLD
C
C Solution of the system
      DO 15 I=2,MS-1
          COEF=CON/A(I-1)
          A(I)=A(I)-COEF*CON
15     SOURCE(I)=SOURCE(I)-COEF*SOURCE(I-1)
          FLUX(MS-1)=SOURCE(MS-1)/A(MS-1)
      DO 20 I=MS-2,1,-1
20     FLUX(I)=(SOURCE(I)-CON*FLUX(I+1))/A(I)
          TSUM=0.d0
C
C Calculation of Keff
      SUM1=0.d0
      SUM2=0.d0
      DO 25 I=1,MS-1
          SUM1=SUM1+FLUXO(I)
25     SUM2=SUM2+FLUX(I)
          KNEW=SUM2/SUM1*KOLD
C
C Calculation of error terms
      ERR1=DABS((KNEW-KOLD)/KOLD)
      ERR2=0.d0
      DO 30 I=1,MS-1

```

```

ERR=DABS((FLUX(I)-FLUXO(I))/FLUXO(I))
IF (ERR.GT.ERR2) ERR2=ERR
30 CONTINUE
IF (ERR1.GT.EPSF.OR.ERR2.GT.EPSF) THEN
KOLD=KNEW
DO 35 I=1,MS
35 FLUXO(I)=FLUX(I)
GO TO 5
ENDIF
C End of inner iteration
C
IF (DABS(KNEW-1.d0).GT.EPSK) THEN
BOR=BOR*KNEW
ITER=0
PRINT*, 'BORON = ',BOR
GO TO 2
ENDIF
C End of outer iteration
C
C Rescaling the flux by reactor power
PSUM=0.d0
DO 40 I=1,MS-1
40 PSUM=PSUM+FLUX(I)
POWER=PCOEF*SIGMAF*H*PSUM
IF (DABS((POWER-THPOW)/THPOW).GT.EPSF) THEN
DO 45 I=1,MS-1
45 FLUXO(I)=FLUX(I)*THPOW/POWER
ITER=0
GO TO 5
ENDIF
C End of solution
C
OPEN(UNIT=1,NAME='TO',STATUS='UNKNOWN')
Z=0.d0
WRITE(1,50) Z,Z,Z,TIN
50 FORMAT(2X,F6.4,3(2X,E15.7))
TSUM=0.d0
C
C Calculation of xenon, iodine, and precursor concentrations
DO 55 I=1,MS-1

```

```

XXE(I)=COEF2*FLUX(I)/(DECAYX+SIGMIX*FLUX(I))
XI(I)=YIELDI*SIGMAF*FLUX(I)/DECAYI
T(I)=TCOEF*(H*TSUM+FLUX(I)*H/2.d0)+TIN
C(I)=CCOEF/KNEW*FLUX(I)
Z=Z+.005d0
WRITE(1,50) Z,FLUX(I),XXE(I),T(I)
55   TSUM=TSUM+FLUX(I)
Z=1.d0
T(MS)=TCOEF*H*TSUM+TIN
TAV=(T(MS)+TIN)/2.d0
WRITE(1,50) Z,FLUX(MS),XXE(MS),T(MS)

C
P1=FLUX(100)*HZ/2.d0
P2=P1
DO 110 I=1,99
    P1=P1+HZ*FLUX(I)
110   P2=P2+HZ*FLUX(I+100)
A0=(P2-P1)/(P1+P2)*100d0
A02=1.d0-2.d0*FQ1/P1
DO 34 I=1,5
34   CRBANK(I)=BES(I)*AR*CR
CRBANK(5)=CRBANK(5)/RAT
J=1
DO 3 I=1,MX
    IF (I.GE.IPOS(J)) J=J+1
3   SIGMAC(I)=SIGMC(J)
IPOS(5)=IPOS(1)
IPOS(1)=IPOS(3)
IPOS(2)=IPOS(3)
IPOS(4)=IPOS(3)
SFNU=NU*SIGMAF/KNEW
HZ=HEIGHT/DBLE(MS)
TIME=0.d0
HT=6.d1
PRINT*,' Enter output file name '
READ*,FOUT
OPEN(UNIT=2,NAME=FOUT,STATUS='UNKNOWN')
WRITE(2,*) MS,KNEW,BR2
WRITE(2,*) POWER,HEIGHT,BOR
WRITE(2,*) D,NU,SIGMAF

```

```
WRITE(2,*) SIGMAA ,TIN ,SIGMAC  
WRITE(2,*) NWO ,CB ,SIGMIB  
WRITE(2,*) YIELDX ,DECAYX ,SIGMIX  
WRITE(2,*) YIELDI ,DECAYI ,IPOS  
WRITE(2,*) CRBANK ,SFNU ,AMNUSF ,BETA ,DECAYC ,TAV  
WRITE(2,*) PCOEF ,TCOEF ,CON ,AO ,AO2  
WRITE(2,*) FLUXO ,XXE ,XI ,C  
WRITE(2,*) HZ ,HT ,TIME  
WRITE(2,*) T  
STOP  
END
```

APPENDIX E: REACTIVITY WORTH ROUTINES

The calculation of reactivity worth curves for FLCRs and boron are performed by two separate programs for practical purposes, even though the solution methods are the same. The first program retrieves the steady state solution and generates the differential rod worth curves for the FLCRs. The second one solves for the reactivity worth of a user defined unit of boron concentration change, and stores the result in the same file that the rod worth curves are stored.

```

C
C CRWORTH.FOR
C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (MX=200)
      DOUBLE PRECISION XXE(MX),XI(MX),A(MX),SOURCE(MX),T(MX),TR(MX)
      DOUBLE PRECISION SIGMC(4),SIGMAC(MX),FLUX(MX),FLUXO(MX),C(MX)
      DOUBLE PRECISION CRBANK(5),WR(4),KOLD,NU,NW,KIN,KNEW
      DOUBLE PRECISION SIGMAD(MX),XF(MX),CD(MX)
      INTEGER IPOS(5),IPOSD(5)
      CHARACTER*10 FILE1,FILE2
      CHARACTER*5 BANK(5)
      DATA BANK/'flcra','flcrb','flcrc','flcrd','plcr'/
C
      PRINT*,' Enter the name of steady state data file: '
      READ*,FILE1
      OPEN(UNIT=1,NAME=FILE1,STATUS='UNKNOWN')
      PRINT*,' Enter the name of output file: '
      READ*,FILE2

```

```

OPEN(UNIT=2,NAME=FILE2,STATUS='UNKNOWN')
READ(1,*) MS,KIN,BR2
READ(1,*) THPOW,HEIGHT,BORD
READ(1,*) D,NU,SIGMAF
READ(1,*) SIGMAA,TIN,SIGMAD
READ(1,*) NW,CB,SIGMIB
READ(1,*) YIELDX,DECAYX,SIGMIX
READ(1,*) YIELDI,DECAYI,IPOSD
READ(1,*) CRBANK,SFNU,AMNUSF,BETA,DECAYC,TAV
READ(1,*) PCOEF,TCOEF,CON,AO,AO2
READ(1,*) XF,XXE,XI,CD
READ(1,*) D1,D2,D3
READ(1,*) TR

C
PI=4.d0*ATAN(1.d0)
EPS=1.d-5
H=HEIGHT/DBLE(MS)
CCOEF=BETA*NU*SIGMAF/DECAYC
COEF2=(YIELDI+YIELDX)*SIGMAF
K=0
9   K=K+1
    IFL=0
    IDP=20

C
C Outmost loop for each bank
11  DO 21 I=1,5
21  IPOS(I)=IPOSD(I)
    TSUM=0.d0
    DO 22 I=1,MS
      FLUXO(I)=XF(I)
      T(I)=TCOEF*(H*TSUM+FLUXO(I)*H/2.d0)+TIN
      TSUM=TSUM+FLUXO(I)
      C(I)=CD(I)
22  SIGMAC(I)=SIGMAD(I)
C
PRINT*,K,IDP
BOR=BORD
KOLD=KIN
CALL MOVECR (SIGMAC,CRBANK,IPOS,K,IDP)
ITER=0

```

```

C
C Solution of nonlinear eigenvalue problem without
C criticality search
2   COEF1=2*D/H**2+SIGMIB*CB*BOR*NW+D*BR2+SIGMAA
C
5   ITER=ITER+1
   TSUM=0.d0
   DO 10 I=1,MS-1
     A(I)=COEF1+SIGMAC(I)+SIGMIX*COEF2*FLUXO(I)/
+     (DECAYX+SIGMIX*FLUXO(I))+AMNUSF*(T(I)-TR(I))
10  SOURCE(I)=(1.d0-BETA)*NU*SIGMAF*FLUXO(I)/KOLD+DECAYC*C(I)
C
   DO 15 I=2,MS-1
     COEF=CON/A(I-1)
     A(I)=A(I)-COEF*CON
15  SOURCE(I)=SOURCE(I)-COEF*SOURCE(I-1)
     FLUX(MS-1)=SOURCE(MS-1)/A(MS-1)
     C(MS-1)=CCOEF*FLUX(MS-1)
   DO 20 I=MS-2,1,-1
     FLUX(I)=(SOURCE(I)-CON*FLUX(I+1))/A(I)
20  C(I)=CCOEF*FLUX(I)
C
   TSUM=0.d0
   DO 23 I=1,MS-1
     T(I)=TCOEF*(H*TSUM+FLUX(I)*H/2.d0)+TIN
23  TSUM=TSUM+FLUX(I)
   T(MS)=TCOEF*H*TSUM+TIN
C
   SUM1=0.d0
   SUM2=0.d0
   DO 25 I=1,MS-1
     SUM1=SUM1+FLUXO(I)
25  SUM2=SUM2+FLUX(I)
   KNEW=SUM2/SUM1*KOLD
C
   ERR1=DABS((KNEW-KOLD)/KOLD)
   ERR2=0.d0
   DO 30 I=1,MS-1
     ERR=DABS((FLUX(I)-FLUXO(I))/FLUXO(I))
     IF (ERR.GT.ERR2) ERR2=ERR

```

```

30     CONTINUE
      IF (ERR1.GT.EPS.OR.ERR2.GT.EPS) THEN
          KOLD=KNEW
          DO 35 I=1,MS
35         FLUXO(I)=FLUX(I)
          GO TO 5
      ENDIF
      PRINT*, ITER
      PSUM=0.d0
      DO 40 I=1,MS-1
40         PSUM=PSUM+FLUX(I)
          POWER=PCOEF*SIGMAF*H*PSUM
          IF (DABS((POWER-THPOW)/THPOW).GT.EPS) THEN
              DO 45 I=1,MS-1
45         FLUXO(I)=FLUX(I)*THPOW/POWER
              ITER=0
              GO TO 5
          ENDIF
      C
      C Calculation of reactivity at each case
          RHO=(KNEW-KIN)/KNEW
          IF (IFL.EQ.0) THEN
              WX=RHO
              IFL=1
              IDP=-179
              GO TO 11
          ENDIF
      C
      C Calculation of total reactivity worth of each bank
          WR(K)=WX-RHO
          IF(K.LT.4) GO TO 9
          PRINT*, WR
          SUM=0.d0
          K=0
      C
      C Calculation of differential worth curves
          DO 201 I=1,MS
201         SUM=SUM+(SIN(PI*DBLE(I)/DBLE(MS)))**2
202         K=K+1
          DO 203 I=1,MS

```

```

      D=WR(K)*(SIN(PI*DBLE(I)/DBLE(MS)))**2/SUM
203  WRITE(2,100) BANK(K),I,D
      IF(K.LE.3) GO TO 202
100  FORMAT(2X,A5,I4,2X,E15.7)
      STOP
      END

C
C
      SUBROUTINE MOVECR (SIGMAC,CRBANK,IPOS,IOP1,IDP)
      DOUBLE PRECISION SIGMAC(200),CRBANK(5)
      INTEGER IPOS(5)

C
      IS=SIGN(1,IDP)
      IL=IPOS(IOP1)+IS
      IU=IPOS(IOP1)+IDP
      IF (IDP.GT.0) THEN
          IL=IL-1
          IU=IU-1
      ENDIF
      IF (IOP1.EQ.5) THEN
          I1=IPOS(5)-IS
          I2=I1+50
          DO 1 I=IL,IU,IS
              SIGMAC(I)=SIGMAC(I1)
1          SIGMAC(I+50)=SIGMAC(I2)
      ELSE
          DO 2 I=IL,IU,IS
2          SIGMAC(I)=SIGMAC(I)-IS*CRBANK(IOP1)
      ENDIF
      IPOS(IOP1)=IPOS(IOP1)+IDP
      RETURN
      END

```

C

C BRWORTH.FOR

```

    IMPLICIT REAL*8 (A-H,O-Z)
    PARAMETER (MX=200)
    DOUBLE PRECISION A(MX),B(MX),XXE(MX),XI(MX),T(MX),TR(MX)
    DOUBLE PRECISION SIGMAC(MX),FLUX(MX),FLUXO(MX),C(MX),XF(MX)
    DOUBLE PRECISION CRBANK(5),KIN,KOLD,NU,NW,CD(MX),W(3)
    INTEGER IPOS(5)
    CHARACTER*10 FILE1

```

C

```

    PRINT*,' Enter the steady state file name '
    READ*,FILE1
    OPEN(1,NAME=FILE1,STATUS='UNKNOWN')
    PRINT*,' Enter the name of output file: '
    READ*,FILE1
    OPEN(UNIT=2,NAME=FILE1,STATUS='UNKNOWN',ACCESS='APPEND')
    READ(1,*) MS,KIN,BR2
    READ(1,*) THPOW,HEIGHT,BORD
    READ(1,*) D,NU,SIGMAF
    READ(1,*) SIGMAA,TIN,SIGMAC
    READ(1,*) NW,CB,SIGMIB
    READ(1,*) YIELDX,DECAYX,SIGMIX
    READ(1,*) YELDI,DECAYI,IPOS
    READ(1,*) CRBANK,SFNU,AMNUSF,BETA,DECAYC,TAV
    READ(1,*) PCOEF,TCOEF,CON,AO,AO2
    READ(1,*) XF,XXE,XI,CD
    READ(1,*) D1,D2,D3
    READ(1,*) TR

```

C

```

    PI=4.d0*ATAN(1.d0)
    HZ=HEIGHT/DBLE(MS)
    CCOEF=BETA*SFNU/DECAYC
    COEF2=(YELDI+YIELDX)*SIGMAF
    IFL=1
    DT=1d0
    PRINT*,' Enter the minimum amount of boron change '
    READ*,DB
    HT=60.d0
    C1=1.d0-DT*DECAYC
    C2=DT*BETA*SFNU

```

```

      TIMEOUT=HT
C
C  Outermost iteration for positive and negative reactivity insertion
11      DO 21 I=1,200
          FLUXO(I)=XF(I)
21      C(I)=CD(I)
C
          BOR=BORD+DBLE(IFL)*DB
          KOLD=KIN
C
          POWER=THPOW
          POLD=POWER
          TIME=0.d0
C
C  Solution of nonlinear eigenvalue problem without
C  criticality search
1      COEF1=2*D/HZ**2+SIGMIB*CB*BOR*NW+D*BR2+SIGMAA
          TIME=TIME+DT
          TSUM=0.d0
          DO 10 I=1,MS-1
              T(I)=TCOEF*(HZ*TSUM+FLUXO(I)*HZ/2.d0)+TIN
              A(I)=COEF1+SIGMAC(I)+AMNUSF*(T(I)-TR(I))
              +
                +SIGMIX*XXE(I)-(1.d0-BETA)*SFNU
              C(I)=C1*C(I)+C2*FLUXO(I)
              B(I)=DECAYC*C(I)
10      TSUM=TSUM+FLUXO(I)
          T(MS)=TCOEF*HZ*TSUM+TIN
C
          DO 15 I=2,MS-1
              COEF=CON/A(I-1)
              A(I)=A(I)-COEF*CON
15      B(I)=B(I)-COEF*B(I-1)
          FLUX(MS-1)=B(MS-1)/A(MS-1)
          SUM1=FLUX(MS-1)
          DO 20 I=MS-2,1,-1
              FLUX(I)=(B(I)-CON*FLUX(I+1))/A(I)
20      SUM1=SUM1+FLUX(I)
C
          POWER=PCOEF*SIGMAF*HZ*SUM1
          DPDT=(POWER-POLD)/DT

```

```
POLD=POWER
IF (TIME.LT.TIMEOUT) THEN
  DO 35 I=1,MS
35   FLUXO(I)=FLUX(I)
     GO TO 1
ENDIF

C
C Absolute value of reactivity worths are averaged
TP=HT/LOG(POWER/THPOW)
IF (IFL.EQ.1) THEN
  W(1)=-1.d0*0.0848d0/TP
  IFL=-1
  GO TO 11
ENDIF
W(2)=0.0848d0/TP
W(3)=(W(1)+W(2))/2.d0
WRITE(1,*) DB
WRITE(2,101) DB,W(3)
101  FORMAT(/,2(2X,E15.7))
WRITE(2,*) TAV
STOP
END
```