

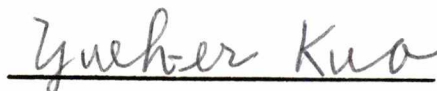
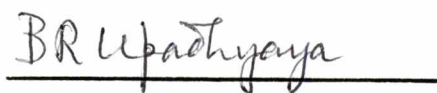
To the Graduate Council:

I am submitting herewith a thesis written by José A. Llamas-Arjona entitled "Research on Closed Loop Control Theory." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

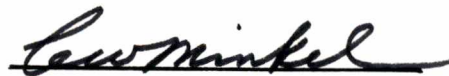


Rafael B. Pérez, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



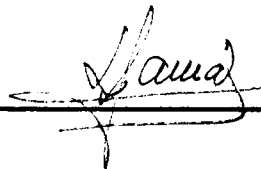
Vice Provost

and Dean of Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature 

Date Nov 27, 89

**RESEARCH ON CLOSED LOOP
CONTROL THEORY**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

José A. Llamas-Arjona

December 1989

To Susana , a very
special person in my life.

ACKNOWLEDGMENTS

The work performed in this thesis was sponsored by the U.S. Department of Energy Advanced Control Program (ACTO) at the Oak Ridge National Laboratory (ORNL).

The author wishes to express his gratitude to his major professor Rafael B. Pérez, Pedro J. Otaduy, Supervisory Control Task Leader and to J.D. White, ACTO program manager, for their continuous support and help.

The author also wishes to thank Luis Rovere, José and Carlos March-Leuba. Without their encouragement, comments and suggestions this work could hardly have been accomplished.

The author is indebted to all the members of the committee, who so generously contributed their time and talent for the completion of this work: Dr. T.W. Kerlin, Dr. B.R. Upadhyaya and Dr. Y. Kuo.

Last but not least, the author thanks all the members of his family for their support and understanding, and thanks Mr. Israel Alguindigue who kept the author in good physical shape during the elaboration of this thesis.

ABSTRACT

A general control algorithm has been developed within the framework of Lagrangian mechanics. In the present algorithm the cost function plays the role of the potential energy in the Lagrangian formalism and the model equations are introduced as constraints weighted by a set of Lagrange multipliers. Since the Lagrangian and Hamiltonian descriptions of classical mechanics are equivalent, the new methodology is also, in principle, equivalent to Pontryagin's Maximum Principle (PMP) formulation of optimal control theory. Previous work by March-Leuba and Pérez at the University of Tennessee, Knoxville, had shown that for demand-following applications, the usual two-point boundary problem, which arises in the PMP methodology could be converted into an initial value problem. However, in this PMP-based algorithm the sign of the time derivatives in the adjoint equations had to be inverted to avoid numerical instabilities and the formulation could not function in the case of sampled input signals. Nevertheless, for the case in which the plant model and control equations were solved together, the algorithm proved to be flexible and capable of tracking uncertain terms and parameters, as well as working with noise-contaminated signals.

The present work was initiated with the purpose of developing, from first principles, a control algorithm which had to be robust against numerical integration instabilities and capable of working with sampled input signals. To this end the Lagrangian formalism was shown to be more appropriate than its

equivalent Hamiltonian formalism.

Firstly the formalism is illustrated by using a single nonlinear equation to achieve a clear understanding of the workings of the control algorithm. Secondly the formalism is applied to a realistic simulation of a simplified model of a LMR (Liquid Metal) reactor. In this case two scenarios are considered. The first one deals with signals delayed by the dynamics of the various sensors, being the equations of the plant model and the control solved together. In the second, sampled input signals are introduced with a PID controller for the control of the plant, and the Lagrangian formalism is used to follow the uncertain terms (i.e. the unknown part of the plant dynamics).

TABLE OF CONTENTS

CHAPTER	PAGE
1. INTRODUCTION	1
1.1 Introduction	1
1.2 Motivation and Objectives	1
1.3 Review of Previous Work	4
1.4 Organization of the Text	5
1.5 Original Contributions	7
2. GENERAL THEORY	9
2.1 Introduction	9
2.2 General Theory	10
2.3 Solving the Differential Equations	15
3. APPLICATIONS	16
3.1 Introduction	16
3.2 Single Equation Model	18
3.2.1 Lagrangian Mechanics-Based Algorithm	19
3.2.2 Proportional Integral Differential Controller	30
3.2.3 Conclusions	30
3.3 Simplified Model of a Liquid Metal Reactor	34

CHAPTER	PAGE
3.3.1 Power Generation Dynamics	38
3.3.2 Core Thermohydraulics	41
3.3.3 Pipes	43
3.3.4 Steam Generator	44
3.3.5 Steam Drum	45
3.4 Control for a Simplified Model of a Liquid Metal Reactor	48
3.4.1 Lagrangian Mechanics-Based Control	48
3.4.2 Proportional Integral Differential Controller with Un- known Terms	52
4. CONCLUSIONS AND RECOMMENDATIONS	64
4.1 Conclusions	64
4.2 Recommendations	66
REFERENCES	67
APPENDICES	70
A. General Optimal Control Theory	72
B. Mathematical Background	85
C. ACSL Source Code	88
VITA	104

LIST OF FIGURES

FIGURE	PAGE
3.1 Demand following as a function of time	23
3.2 Unknown term g_1	24
3.3 Control and conjugate momentum (1 equation)	25
3.4 Unpredicted single perturbation in state variable	26
3.5 Control for a single perturbation	27
3.6 Repetitive perturbation in the plant	28
3.7 Control for repetitive perturbation	29
3.8 Proportional Integral Differential cotroller with single perturbation	31
3.9 Proportional Integral Differential controller with repetitive per- turbation	32
3.10 Proportional Integral Differential controller behavior in the de- mand following problem	33
3.11 Lagrangian mechanics control without the proportional terms, in the cases of a stable and an unstable system	35
3.12 $\dot{x}_1$ as a function of x_1 in the case of a Lagrangian mechanics control with no proportional terms	36
3.13 Original PRISM model formed by three modules	39
3.14 Liquid Metal Reactor simplified model	40

FIGURE	PAGE
3.15 Lagrangian Mechanics control in the demand-following problem	
with delayed signals	53
3.16 Mass and pressure with delayed signals	54
3.17 Unknown term g_1 in Liquid Metal Reactor	55
3.18 Unknown terms g_4 and g_8 for the Liquid Metal Reactor	56
3.19 Shape of the reactivity control and the primary pump controls in	
the Liquid Metal Reactor	57
3.20 Shape of the variation in the control valves u_3 and u_4	58
3.21 Demand following problem with a Proportional Integral Differen-	
tial Controller for the neutronics	61
3.22 Proportional Integral Differential control for the neutronics . . .	62
3.23 Unknown using a Proportional Integral Differential controller . .	63

LIST OF ABBREVIATIONS

ACSL : Advanced Continuous Simulation Language

FTT : Free Terminal Time

LM : Lagrangian Mechanics

LMR : Liquid Metal Reactor

PID : Proportional-Integral-Differential

PMP : Pontryagin's Maximum Principle

TPBVP: Two Point Boundary Value Problem

CHAPTER 1

INTRODUCTION

1.1 Introduction

Historically, the control philosophies followed in the nuclear industry have been based on Proportional-Integral-Differential (PID) methodologies or, on the application of Pontryagin's Maximum Principle (PMP) [1], [2] to linearized plant models. With the exception of Rosztoczy et al. [4] early work, very little use of the PMP principle has been made for the development of non-linear control algorithms. In the author's opinion the reason for this situation arises from two facts: firstly the PMP formulation requires the solution of a two-point boundary value problem, whose iterative nature prevents on-line operation and secondly, since realistic models of most systems are seldom completely known, the PMP formalism could not be made robust against unanticipated plant transients and/or lack of knowledge of plant dynamics.

1.2 Motivation and Objectives

With the availability of new systems and computers, industry is demanding new reliable techniques that ease the human task in a control room. Since

computer-based systems operate with digital signals, there is a renewed interest in the development of new algorithms that can be implemented in a digital form. Advanced control algorithms are model based, but the internal plant model does not usually describe accurately the real world system. In addition, the advanced control algorithms must perform well with incomplete plant information. Thus, the control engineer is faced with both lack of signal information and lack of system's knowledge.

The algorithm proposed in this thesis is an attempt to alleviate these problems. Its main advantage is its ability to estimate some unavailable measurements from the plant with great success. This allows us to work with small and not very accurate models of the systems. Our lack of precision using small models is compensated by the so called "unknown terms".

A first approach to the problem was started by March-Leuba [5],[6] and Pérez in the Nuclear Engineering Department, University of Tennessee, Knoxville. They developed a control theory based on the PMP principle. In order to integrate the set of ordinary differential equations some approximations had to be made. These will be briefly discussed in section (1.3) in this chapter. Eliminating the approximations makes the problem formally different, and allows us to build a more robust theory from the mathematical point of view.

Two main objectives are pursued in this work:

1. The first one is to build a consistent general control theory that can work with any kind of system. That is why the theoretical point of view is the

most important issue in this thesis. A special effort was made to show clearly how the algorithm can be built. Because of the complexity of real systems, a large set of equations is needed to describe them. Uncertainties in large models are a large source of errors which makes the task of developing a new technique very difficult. In order to accomplish our first objective, the author illustrates his studies with a single equation system. The nonlinearity in the control in this equation has some resemblance to that describing the neutronics in a nuclear reactor. This equation is highly nonlinear, and just changing the value of the constant a in Eq.(1.1) the system can go from stable to unstable, and vice versa.

$$\dot{x} = ax^2 + \frac{ux}{\Lambda} \quad (1.1)$$

$$x(t = 0) = x_0 \quad (1.2)$$

The algorithm has been implemented in ACSL (Advanced Continuous Simulation Language), which can be easily incorporated to personal computers. This language was chosen because the user can interact continuously with the program. As in the developing process many new things are constantly tried, the interaction with the algorithm makes finding the solution to the problem easier.

2. Our second objective is to apply the algorithm to a realistic engineering system: an LMR (Liquid Metal) reactor. The model initially chosen was developed and supplied by Luis Rovere [7]. It was consisted of three mod-

ules, each of them described by 32 equations. This model was subsequently simplified in collaboration with Rovere to just one module described by 9 equations.

1.3 Review of Previous Work

A PMP was developed for nonlinear systems (see ref. [5]). In this work the Free Time Optimal Control Problem (FTT) was reformulated, so that the solution of the classical two point boundary value problem (TPBVP) could be avoided.

Let

$$\frac{dy}{dt} = F(y, u) \quad (1.3)$$

be a nonlinear system, where y represents the state variables of the system and u the controls. The free terminal time problem is formulated as the minimization of the *extended cost function*

$$J = \int_{t_0}^{t_F} \left[H(y, w, u) - w^T \frac{dy}{dt} \right] dt \quad (1.4)$$

where w is a set of Lagrange multipliers (adjoints or momenta), and H represents the Hamiltonian defined as

$$H = w^T \frac{dy}{dt} + V(y, u) \quad (1.5)$$

where V is a potential function. We assume that H is time-invariant, $y(t_0)$ and $y(t_F)$ are given and the system is at steady-state at $t = t_0$. Because of

the symmetry of the problem the minimization of the cost function can be done backwards. After a time inversion of the expression for J , the following Hamiltonian equations are obtained:

$$\frac{dy_i}{dt} = \frac{dH}{dw_i} \quad (1.6)$$

$$\frac{dw_i}{dt} = -\frac{dH}{dy_i} \quad (1.7)$$

and the conditions

$$\frac{dH}{du_i} = 0 \quad (1.8)$$

$$(H)_{t_0} = 0 \quad (1.9)$$

If steady-state conditions are assumed at $t = t_0$, and $\frac{dV}{du_i}$ are different from zero, then, the initial conditions for w can be found. Moreover, because the state variables are time-reversal invariant, and the controls are only function of y and w , the first Hamilton equation can be solved forward instead of backwards as the result of the variations implied.

1.4 Organization of the Text

After introducing a previous approach to the control of nonlinear systems [5], a general control theory based on Lagrangian Mechanics will be developed and discussed in chapter 2. All the applications will be discussed in chapter 3. Chapter 1 will be divided into three main sections:

Section 1 : We shall apply the general technique to a single equation problem, showing with as much detail as possible how to build the algorithm from the beginning. We shall integrate the model describing the plant and the control algorithm independently. This induces stability problems to the algorithm. The same problem will be controlled with a PID controller showing that it is outperformed by the LM based algorithm.

Section 2 : In order to apply our algorithm to the LMR reactor we had to simplify the original set of equations as a first step to show the algorithm works in a more complex case. The simplifications and the equations will be discussed in this section.

Section 3 : In this section we shall apply the general algorithm to the set of equations described in section 2. In this case we shall consider continuous signals to illustrate the use of only one Lagrangian. The signals will be delayed with a characteristic transfer function of the form

$$\frac{dy_d}{dt} = \frac{1}{\tau}(y - y_d) \quad (1.10)$$

where y is the signal, y_d the delayed signal and τ a constant that gives the delay time.

In this same section it will be shown how we can take advantage of the algorithm, using a conventional PID as a control in the case of discrete signals, but getting an estimation of the model given by the formalism.

In chapter 4 some conclusions and suggestions will be made. In Appendix B some details of the analytical integration of some ordinary differential equations that arise in the problem will be given, while in Appendix C there is a print-out of the ACSL source code, where the plant simulation was developed by Luis Rovere, and to which, the control algorithm has been added. All the values of the constants and the actual expressions of the controls can be found there.

1.5 Original Contributions

Several original contributions were made in this work

- In our algorithm the sign of the adjoint need not be changed in order to integrate the set of differential equations, as it had to be done in [5]. In this way we keep the Optimization Principle as originally formulated.
- The instabilities originated when we integrate the real plant equations and the control algorithm independently have been successfully solved.
- As a difference to the previous algorithms [5] the adjoint equations have been reduced just to the source terms. This adds stability to the algorithm.
- The derivative of the signal has been included, as well as some cross terms (signal and derivative of the signal) in order to handle the case of independent integrations.

- A proportional term has been added to that given by the variational principle in the case of the control and "unknown terms".
- The problem of delayed signals and controls has been solved within the general formulation.
- The relationship between our algorithm and a conventional PID has been shown.
- Even in the case of using a PID as the control, we still are able to get some information from the model (unknown terms) within the general formalism.

The derivative of the signal and the proportional term are really only needed in the case of solving the equations of the real plant separated from those describing the control. These increase the difficulty of the algebra, and in the case in which both are integrated together they are not needed, as it will be shown in the case of the simplified model of a LMR reactor.

CHAPTER 2

GENERAL THEORY

2.1 Introduction

This chapter deals with the Free Terminal Time (FTT) problem and with the problems which arise from integrating the model describing the real plant and the control algorithm independently. In the study the plant and the controls are assumed to be at equilibrium at the beginning ($t = T_0$) and end ($t = T_F$) of a transient (load following period). Hence the state of the system is known at the final time, T_F , which is not specified (FTT problem). These two problems are studied within the framework of Lagrangian Mechanics, to develop a general formulation to implement a control algorithm based on optimal control theory to avoid the solution of a two-point boundary problem.

In the real world, the input to the control algorithm has been obtained from a sampling device, being these data the only information the control algorithm has available. This induces sizable instabilities to the algorithm. These problem are solved introducing a proportional term in both the control and the unknown terms.

2.2 General Theory

Let X_i , $i = 1, 2, \dots, N$ be the set of state variables describing the plant and determined by the set of equations

$$\dot{X}_i = S_i(X_j, U_{ic}, P_r) \quad (2.1)$$

and the initial conditions

$$X_i(0) = X_{i0} \quad (2.2)$$

where U_{ic} , are the control variables and P_r , ($r = 1, 2, \dots, r_0$) a subset of time-varying plant parameters.

In the present case the plant and controls are assumed to be in equilibrium at the beginning ($t = T_0$) and end ($t = T_F$) of a transient (load following period). Hence the state of the system is known at the final time, T_F ; however, the value of this quantity is not available. We are dealing with a free terminal problem [2] where the final time is not specified.

The control algorithm is based on a reduced model where only the subset of state variables which are measured will be included, and where the unknown part of the plant dynamics will be described by uncertain terms. However, there are some cases where certain state variables cannot be measured, but an expression for them is well known, e.g. the delayed neutrons in a nuclear reactor core. In this case we can take them as known parameters.

Thus the subset of measured state variables will be modeled in the general

form

$$\dot{M}_i = G_i + \sum_c U_{ic} F_{ic}(M_j) \quad (2.3)$$

$$M_i(0) = X_{i0} \quad (2.4)$$

where G_i are the unknown terms in the model and F_{ic} known functions of the state variables M_i of the plant dynamics.

The general form of the control and the uncertain terms will respectively

$$U_{ic} = A_{ic}(D_i - X_i) + \rho_{ic} \quad (2.5)$$

$$G_i = B_{ic}(X_i - M_i) + \gamma_{ic} \quad (2.6)$$

where $A_{ic}, B_{ic}, \rho_{ic}, \gamma_{ic}$ are functions to be determined by the variational principle, and D_i are the demands to be followed by the state variables of the plant X_i .

Following well known procedures [1,2] we set up two different cost functions, one for the control of the plant ($k = 1$) and another to determine the uncertain terms G_i ($k = 2$). Each of the cost functions has the general form

$$J_k = \int_{T_0}^{T_F} dt \mathcal{L}_k(M_i, U_{ic}, A_{ic}, B_{ic}, \gamma_{ic}, \rho_{ic}, G_i) \quad (2.7)$$

where $\mathcal{L}_k$ are the Lagrangian functions

$$\mathcal{L}_k = -V_k +$$

$$\sum_i W_{ki} \left\{ \dot{M}_i - B_{ic}(X_i - M_i) - \gamma_{ic} - \sum_c [\rho_{ic} + A_{ic}(D_i - X_i)] F_{ic} \right\} \quad (2.8)$$

and T_0, T_F are the initial and final times of the plant transient respectively.

The potential functions V_k in Eq.(2.8) are given by

$$\begin{aligned}
V_1 = & -\frac{1}{2} \sum_i \left\{ Q_i (M_i - X_i)^2 + Q_{Ai} (\dot{M}_i - \dot{X}_i)^2 \right\} \\
& - \sum_i \left\{ c_{1i} (M_i - D_i) (D_i - X_i) + c_{1A} (\dot{M}_i - \dot{D}_i) (\dot{D}_i - \dot{X}_i) \right\} \\
& - \frac{1}{2} \sum_i \left[K_i (\gamma_i - \gamma_{iF})^2 + K_{iP} (B_i - B_{iF})^2 \right] \quad (2.9)
\end{aligned}$$

and for the uncertain terms

$$\begin{aligned}
V_2 = & -\frac{1}{2} \sum_i \left\{ L_i (M_i - D_i)^2 + L_{Ai} (\dot{M}_i - \dot{D}_i)^2 \right\} \\
& - \sum_i \left\{ J_i (M_i - D_i) (D_i - X_i) + J_{Ai} (\dot{M}_i - \dot{X}_i) (\dot{D}_i - \dot{X}_i) \right\} \\
& - \frac{1}{2} \sum_i \left[R_{ic} (\rho_{ic} - \rho_{icF})^2 + R_{icP} (A_{ic} - A_{icF})^2 \right] \quad (2.10)
\end{aligned}$$

The quantities $Q_i, Q_{Ai}, L_i, L_{Ai}, R_{ic}, R_{icP}, c_i, c_{iA}, J_i, J_{iA}, K_i, K_{iP}$ are adjustable weights, $\gamma_{iF}, \rho_{iF}, A_{iF}, B_{iF}$ are as yet unspecified functions of the state variables M_i . The Lagrange multipliers W_{ki} are the adjoint state variables given by

$$\frac{\partial \mathcal{L}_k}{\partial M_i} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}_k}{\partial \dot{M}_i} \right) = 0 \quad (2.11)$$

where $\frac{\partial \mathcal{L}_k}{\partial \dot{M}_i}$ are known as the conjugate momenta

$$p_{ki} = \frac{\partial \mathcal{L}_k}{\partial \dot{M}_i} \quad (2.12)$$

Taking variations in Eq.(2.7) and setting $\delta J_k = 0$, one obtains the following Euler-Lagrange equations of motion

$$\frac{\partial \mathcal{L}_1}{\partial B_i} = 0 \quad (2.13)$$

$$\frac{\partial \mathcal{L}_1}{\partial \gamma_i} = 0 \quad (2.14)$$

$$\frac{\partial \mathcal{L}_2}{\partial A_i} = 0 \quad (2.15)$$

$$\frac{\partial \mathcal{L}_2}{\partial \rho_i} = 0 \quad (2.16)$$

$$\begin{aligned} & \frac{\partial \mathcal{L}_1}{\partial M_i} + \sum_i \frac{\partial \mathcal{L}_1}{\partial B_{iF}} \frac{\partial B_{iF}}{\partial M_i} \\ & + \sum_i \frac{\partial \mathcal{L}_1}{\partial \gamma_{iF}} \frac{\partial \gamma_{iF}}{\partial M_i} - \frac{d}{dt} \left\{ \frac{\partial \mathcal{L}_1}{\partial \dot{M}_i} \right\} = 0 \end{aligned} \quad (2.17)$$

$$\begin{aligned} & \frac{\partial \mathcal{L}_2}{\partial M_i} + \sum_i \frac{\partial \mathcal{L}_2}{\partial A_{iF}} \frac{\partial A_{iF}}{\partial M_i} \\ & + \sum_i \frac{\partial \mathcal{L}_2}{\partial \rho_{iF}} \frac{\partial \rho_{iF}}{\partial M_i} - \frac{d}{dt} \left\{ \frac{\partial \mathcal{L}_2}{\partial \dot{M}_i} \right\} = 0 \end{aligned} \quad (2.18)$$

with the conditions

$$\mathcal{L}_k(T_0) = \mathcal{L}_k(T_F) = 0 \quad (2.19)$$

The controls, the uncertain terms and the parameters are determined by the results (2.13), (2.14), (2.15) and (2.16). From the structure of the potential functions (2.9), (2.10) and (2.19) ρ_{iF}, γ_{iF} must be equal to the initial values of their counterparts U_i, G_i at the beginning of the transient.

During the transient they are determined by the condition that the adjoint equations (2.17) and (2.18) must reduce to the form

$$\dot{p}_{1i} = -Q_i(M_i - X_i) - c_i(D_i - X_i) \quad (2.20)$$

$$\dot{p}_{2i} = -L_i(\dot{M}_i - \dot{D}_i) - J_i(\dot{D}_i - \dot{X}_i) \quad (2.21)$$

where

$$W_{1i} = p_{1i} + Q_{1A}(\dot{M}_i - \dot{X}_i) + c_{1A}(\dot{D}_i - \dot{X}_i) \quad (2.22)$$

$$W_{2i} = p_{2i} + L_{1A}(\dot{M}_i - \dot{D}_i) + J_{1A}(\dot{D}_i - \dot{X}_i) \quad (2.23)$$

The reason for the above conditions arises from the fact that the result of integrating (2.20) and (2.21) is invariant with respect to time reflection, if the initial (final) conditions are the same. To eliminate non-source terms in the expressions of the conjugate momenta we have to use the following subsidiary relations:

$$\frac{\partial \gamma_{jF}}{\partial M_i} + \sum_c \rho_{jc} \frac{\partial F_{jc}}{\partial M_i} = 0 \quad (2.24)$$

$$y_j \frac{\partial B_{jF}}{\partial M_i} - B_j \delta_{ij} + \sum_c \Delta_j A_{jc} \frac{\partial F_{jc}}{\partial M_i} = 0 \quad (2.25)$$

$$F_{jc} \frac{\partial \rho_{jcF}}{\partial M_i} + \rho_{jc} \frac{\partial F_{jc}}{\partial M_i} = 0 \quad (2.26)$$

$$\Delta_j \left[F_{jc} \frac{\partial A_{jc}}{\partial M_i} + A_{jc} \frac{\partial F_{jc}}{\partial M_i} \right] = B_j \delta_{ij} \quad (2.27)$$

where $y_i = X_i - M_i$, $\Delta_i = D_i - X_i$.

As it will be clearly shown in section (3.2.3) in Chapter 3 some terms of the control given by the algorithm can be related to the integral-differential terms in a PID. In order to add stability to the algorithm (specially in the discrete case) the derivative of the signal had to be included, as well as a proportional term in both the unknown term and the control. If we observe the expressions

of U_i and G_i we see that the terms $A_{ic}(D_i - X_i)$ in the case of the control and $B_{ic}(X_i - M_i)$ in the case of the unknown term represent proportional part of a PID with adaptive constants.

2.3 Solving the Differential Equations

Some important comments need to be made about the solution of the equations developed for the control algorithm. In the algorithm our plant is simulated by a set of differential equations that describes the behavior of the plant. If these set of equations is solved together with the set of equations describing the control and the internal model, the system has some information that is not available in a real plant for each integration step. Under these circumstances some terms in the control can be neglected, namely the proportional term in the control and the derivative and cross terms in the potential functions given by (2.9),(2.10), and even both Lagrangians can be reduced to only one.

In a real case we have to receive our data from a real plant. This data are sampled with a limited frequency (usually in power plants we deal with sampling times of the order of .5-1 sec), being these data the only information the control algorithm has available. One of the ways this can be simulated is running our plant model in a different computer than our control algorithm. In this case both the derivative terms in the potential functions and the proportional terms in the control are needed in order to make the system stable.

CHAPTER 3

APPLICATIONS

3.1 Introduction

In this chapter applications to some practical problems are illustrated. Because of the complexity of real systems, these are usually described by large sets of equations. Working with this kind of system has two main problems. Firstly the large number of equations is a frequent source of errors, making the development and implementation of a new technique very difficult. Secondly, the large number of equations adds difficulties to the algebra. From the pedagogical point of view this last reason is very important, because it makes it hard to show how to deal with the control technique.

Because of these two reasons, in the first part of this chapter we are going to use a very simple model using only one equation. Here we can show with great detail how to build a closed-loop control.

After this first application we shall move on to the real world. We shall deal with the case a LMR reactor. Because of the large number of equations of the original model [7] (more than 30), we had to reduce the problem to a more manageable size. This will be discussed in the third section of this chapter.

After this we shall show the starting equations for our control and some results.

Two different cases can be considered when we try to implement a control for any given system. The first one is when we integrate the plant model and the control algorithm together, and the second one is when they are integrated more realistically in different blocks. In the latter situation many instability problems arise which greatly complicate the problem. We shall consider this fact in the case of the single equation problem, with typical sampling times of 1 sec. The controls are also given to the plant as a discrete signal. Our control will be compared to a PID controller.

In the case of the LMR reactor, as the case of independent integration has been solved for the single equation problem, something new is tried. We shall integrate the plan and the control equations together, but the signals will be delayed by a given transfer function. In this way we shall illustrate the use of just one Lagrangian and how some terms can be avoided.

In this same case, but for independently integrated equations, we shall attempt a trade-off situation. We shall use a PID control, but we still shall be able to take advantage of the ability of the algorithm to estimate some unknown terms of the model.

3.2 Single Equation Model

In this section we are going to show with great detail how to implement the Lagrangian-Mechanics (LM) based control algorithm for a single equation system described in chapter 2, solving the plant equations and the control algorithm independently.

Consider a plant described by the following equation:

$$\dot{x}_1 = \frac{u_1 x_1}{\Lambda} + a x_1^2 \quad (3.1)$$

with the initial condition

$$x_1(t = 0) = x_{10} \quad (3.2)$$

where x_1 is the state variable, u_1 the control calculated in the algorithm, and Λ and a are two given constants, in our case:

$$\Lambda = 0.1218 \text{sec}^{-1}$$

$$a = -2.10^{-4} \text{sec}^{-1}$$

Notice that this equation has certain resemblance with that describing the neutronics of a nuclear reactor core. Changing the constant a to a positive value makes the system unstable.

3.2.1 Lagrangian Mechanics-Based Algorithm

In order to implement a control based on the general theory developed in chapter 2, we shall consider a model described by

$$\dot{m}_1 = \frac{u_1 m_1}{\Lambda} + g_1 \quad (3.3)$$

$$m_1(t = 0) = x_{10} \quad (3.4)$$

where m_1 is the state variable in the model, and g_1 the so called "unknown term". This term is going to follow the term ax_1^2 . This is a good advantage of the algorithm, because from an incomplete model we are going to obtain a part of the equation we did not know how to model. This will enable us to make some "predictions" in the behavior of the plant, and above all, it will be very useful in the case of diagnostics.

The control and the unknown term will have the following original expressions:

$$u_1 = A_p(d_1 - x_1) + \rho \quad (3.5)$$

$$g_1 = B_P(x_1 - m_1) + \gamma \quad (3.6)$$

where A_P, B_P, ρ and γ are functions to be determined by the algorithm, and d_1 is the demand to be followed by the state variable x_1 .

We set up two different Lagrangians, one for the unknown term

$$\begin{aligned} \mathcal{L}_1 = & -1/2Q_1(m_1 - x_1)^2 - 1/2Q_{1A}(\dot{m}_1 - \dot{x}_1)^2 - c_{1x}(m_1 - d_1)(d_1 - x_1) \\ & - c_{2x}(\dot{m}_1 - \dot{d}_1)(\dot{d}_1 - \dot{x}_1) - 1/2K_P(B_P - B_{PF})^2 - 1/2K_1(\gamma - \gamma_F)^2 \end{aligned}$$

$$+w_1 \left[\dot{m}_1 - \frac{A_P(d_1 - x_1)m_1}{\Lambda} - c_1 \frac{m_1}{\Lambda} - B_P(x_1 - m_1) - \gamma \right] \quad (3.7)$$

and for the control

$$\begin{aligned} \mathcal{L}_2 = & -1/2L_1(m_1 - d_1)^2 - 1/2L_{1A}(\dot{m}_1 - \dot{d}_1)^2 - c_1(m_1 - d_1)(d_1 - x_1) \\ & - c_2(\dot{m}_1 - \dot{d}_1)(\dot{d}_1 - \dot{x}_1) - 1/2R_P(A_P - A_{PF})^2 - 1/2R_1(\rho - \rho_F)^2 \\ & + w_{1a} \left[\dot{m}_1 - \frac{A_P(d_1 - x_1)m_1}{\Lambda} - c_1 \frac{m_1}{\Lambda} - B_P(x_1 - m_1) - \gamma \right] \end{aligned} \quad (3.8)$$

where B_{PF}, A_{PF}, γ_F and ρ_F are freedom functions to be determined.

Let us establish the Euler-Lagrange equations for our problem. The expressions for B_P, γ, A_P and ρ are given by $\partial \mathcal{L}_1 / \partial B_P = 0$, $\partial \mathcal{L}_1 / \partial \gamma = 0$, $\partial \mathcal{L}_2 / \partial A_P = 0$ and $\partial \mathcal{L}_2 / \partial \rho = 0$ respectively

$$B_P = B_{PF} - \frac{w_1}{K_P}(x_1 - m_1) \quad (3.9)$$

$$\gamma = \gamma_F - \frac{w_1}{K_1} \quad (3.10)$$

$$A_P = A_{PF} - \frac{w_{1a}m_1}{R_P\Lambda}(d_1 - x_1) \quad (3.11)$$

$$\rho = \rho_F - \frac{w_{1a}m_1}{\Lambda R_1} \quad (3.12)$$

We shall consider R_P arbitrarily large, without any loss of generality. This will enable us to take $B_P = B_{PF}$ and $A_P = A_{PF}$.

The expressions for the adjoints will be given by:

$$\frac{\partial \mathcal{L}_1}{\partial m_1} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}_1}{\partial \dot{m}_1} \right) = 0 \quad (3.13)$$

$$\frac{\partial \mathcal{L}_2}{\partial m_1} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}_2}{\partial \dot{m}_1} \right) = 0 \quad (3.14)$$

These result in the following:

$$\dot{p}_1 = -Q_1(m_1 - x_1) - c_{1x}(d_1 - x_1) \quad (3.15)$$

$$\dot{p}_{1a} = -L_1(m_1 - d_1) - c_1(d_1 - x_1) \quad (3.16)$$

$$w_1 = p_1 + Q_{1a}(\dot{m}_1 - \dot{x}_1) + c_{2x}(\dot{d}_1 - \dot{x}_1) \quad (3.17)$$

$$w_{1a} = p_{1a} + L_2(\dot{m}_1 - \dot{d}_1) + c_2(\dot{d}_1 - \dot{x}_1) \quad (3.18)$$

In order to reduce the differential equations describing the conjugate momenta to only the source terms, the following conditions arise:

$$\frac{\partial \gamma_F}{\partial m_1} + (x_1 - m_1) \frac{\partial B_{PF}}{\partial m_1} + \frac{\rho}{\Lambda} + \frac{A_P}{\Lambda}(d_1 - x_1) = B_P \quad (3.19)$$

$$\frac{m}{\Lambda} \frac{\partial \rho_F}{\partial m_1} + \frac{m_1}{\Lambda}(d_1 - x_1) \frac{\partial A_{PF}}{\partial m_1} + \frac{\rho}{\Lambda} + \frac{A_P}{\Lambda}(d_1 - x_1) = B_P \quad (3.20)$$

The integration of these four last equations is discussed in Appendix B. All the equilibrium conditions have to be satisfied at $t = 0$. From there we get the initial values u_{1F} and g_{1F} , as well as the integration constants.

The final expressions for the control and for the unknown term are given by:

$$g_{1F} = ax_{10}^2 \quad (3.21)$$

$$u_{1F} = \frac{\dot{d}_{10} - ax_{10}^2}{x_{10}} \Lambda \quad (3.22)$$

$$\begin{aligned} g_1 = g_{1F} + \frac{x_{10}}{\Lambda} [u_{1F} - A_P(d_1 - x_1)] \ln\left(\frac{x_{10}}{m_1}\right) \\ + B_P(x_1 - m_1) + \frac{m_1^2 w_{1a}}{4. \Lambda R_1} - \frac{w_1}{k_1} \end{aligned} \quad (3.23)$$

$$u_1 = [u_{1F} + A_P(d_1 - x_1)] \frac{x_{10}}{m_1} - \frac{m_1 w_{1a}}{2.R_1\Lambda} \quad (3.24)$$

As it can be seen in Fig. 3.1 the plant x_{1d} , as well as the model m_{1d} follow perfectly a given demand dem_{1d} .

The unknown term g_{1d} is the part of the equation we suppose that we do not know how to model. This term should follow what we have called $chk_1 = ax_1^2$, as it can be seen in Fig. 3.2. The resulting control and the conjugate momentum can be seen in Fig. 3.3.

In industry there are two problems that concern engineers: one is the demand following problem, and the other is the capability of the control to keep a given set point in case of any disturbance of state variables of the plant. To test this, we introduced a perturbation in a , what induces a perturbation in the plant.

In the first case we varied the initial value of $a = -4.10^{-4}$ to $a = 4.10^{-4}$ and value that is kept for 300 sec. Notice that the plant goes from a stable to an unstable dynamics (when $a \geq 0$ there is a positive eigenvalue). As it can be seen in Fig. 3.4 the plant matches the demand exactly, while the unknown term g_{1d} estimates the perturbation in a without any problem. In Fig. 3.5 the control and the conjugate momentum can be seen.

In the case of a smaller perturbation in amplitude in a but relatively high frequency we can see in Fig. 3.6 that the plant is kept in a given set point while the term g_{1d} clearly follows the changes in ax_1^2 (Fig. 3.7).

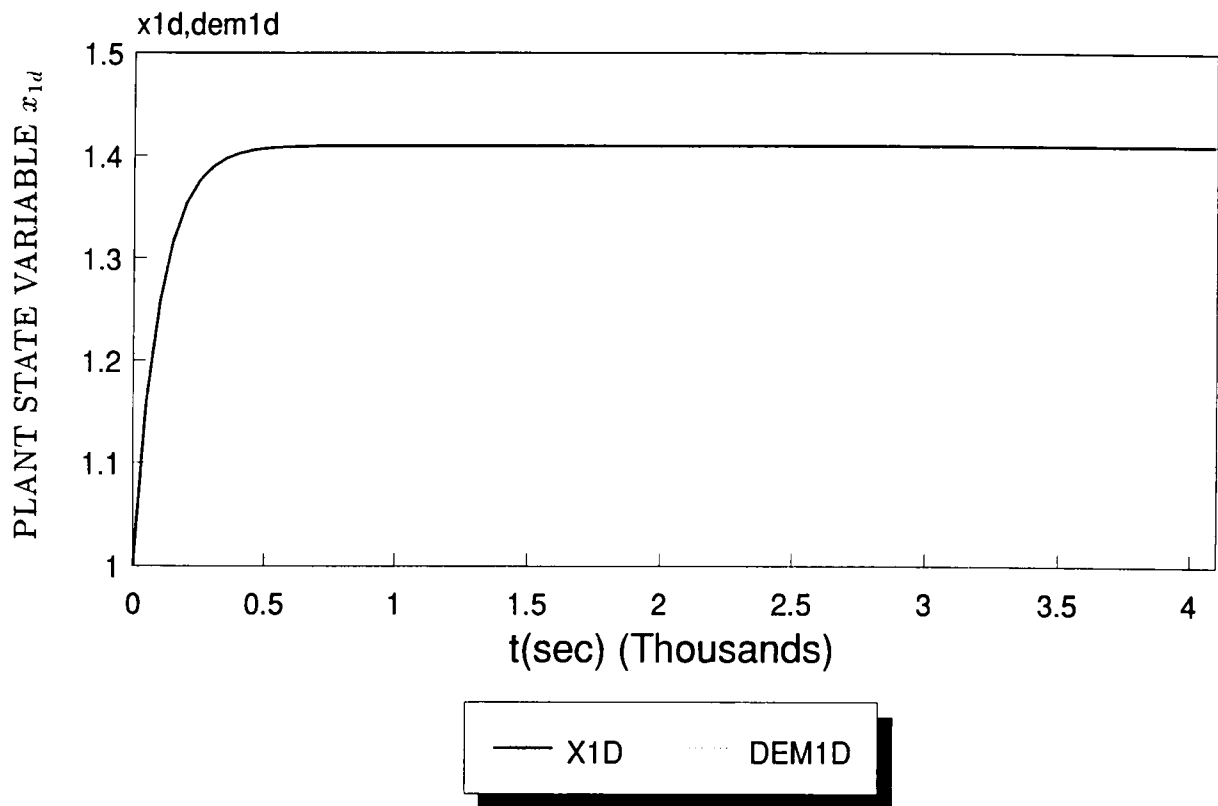


Figure 3.1: Demand following as a function of time. The plant x_{1d} follows exactly a given demand dem_{1d} in the case of solving the plant and control equations independently.

UNKOWN TERM

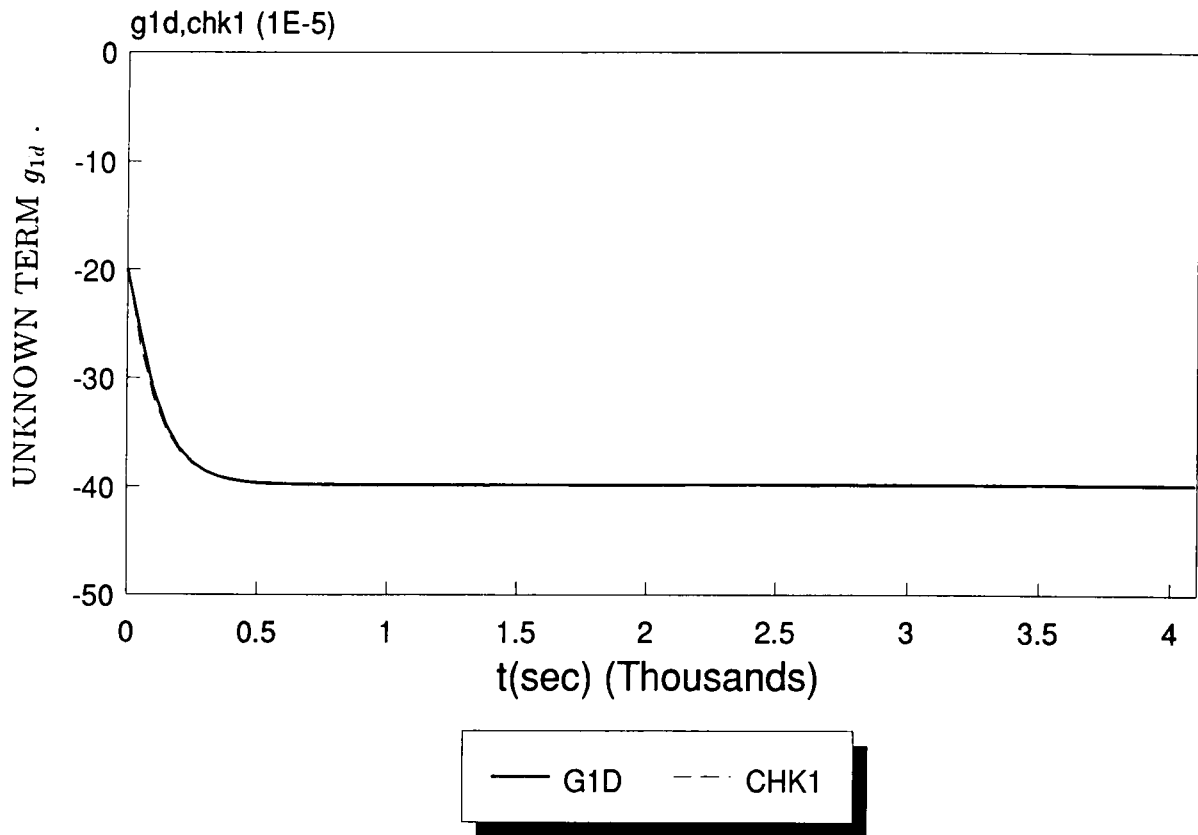


Figure 3.2: Unknown term g_1 . The term g_{1d} calculated by the algorithm reproduces exactly the unknown part of the plant dynamics.

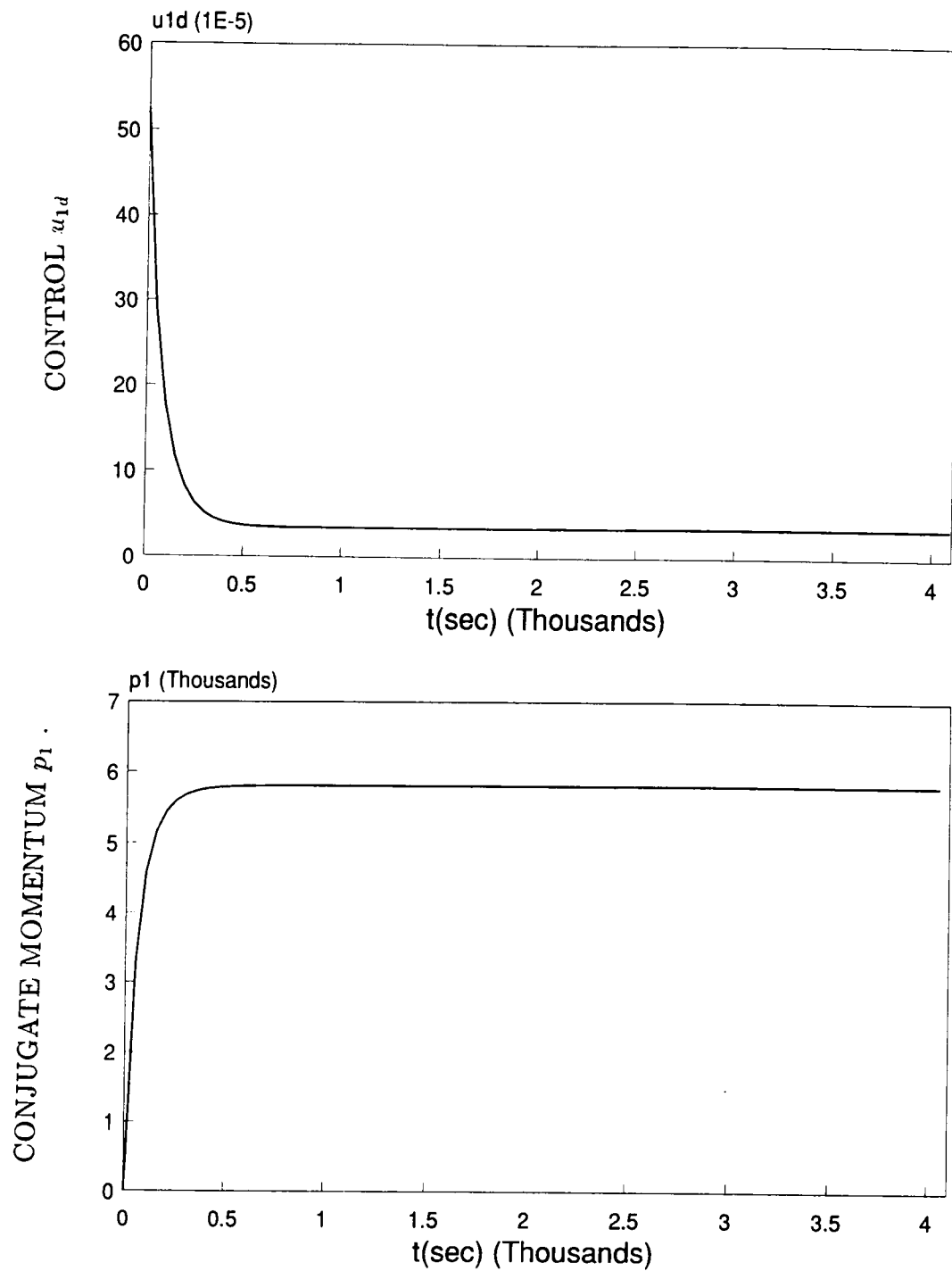


Figure 3.3: **Control and conjugate momentum.** In the top figure it can be seen how the control varies during the transient until it reaches a steady state. In the bottom figure the conjugate momenta p_1 are represented.

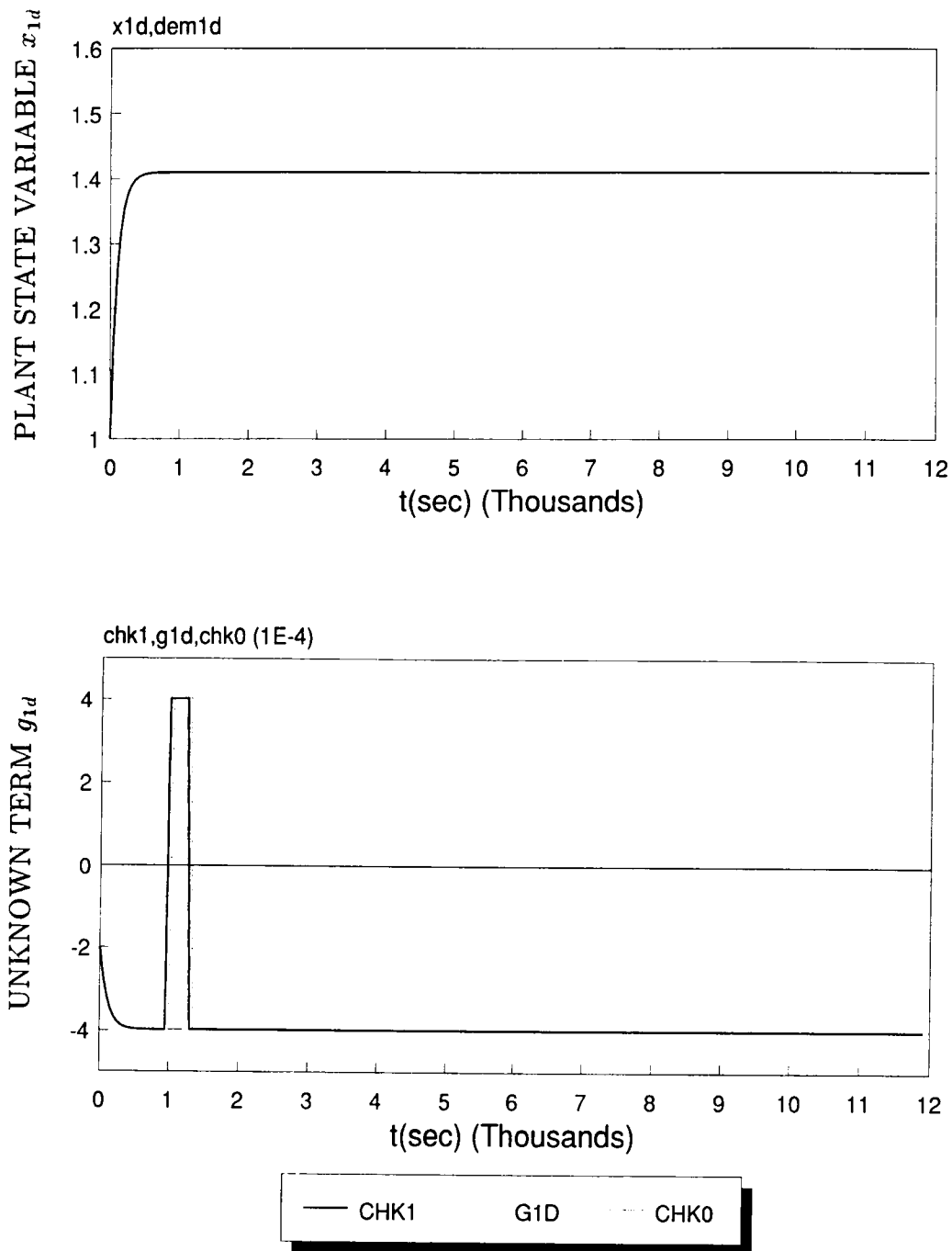


Figure 3.4: Unpredicted single perturbation in state variable. The plant is kept in a given set point, while the term g_{1d} follows exactly the perturbation in ax_1^2 .

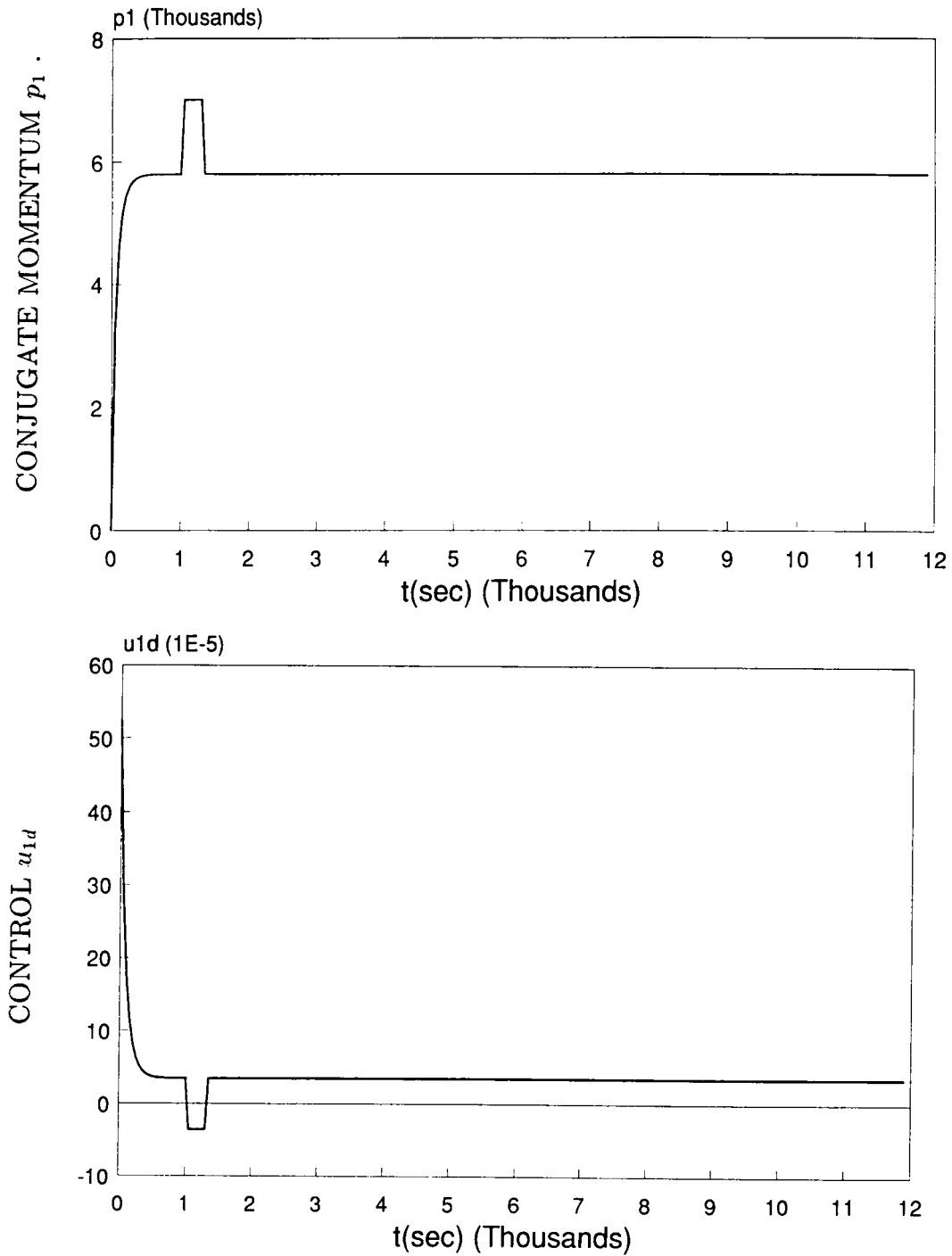


Figure 3.5: Control and momentum in the case of a single perturbation. The control u_{1d} and the conjugate momentum p_1 vary to compensate the variation in ax_1^2 .

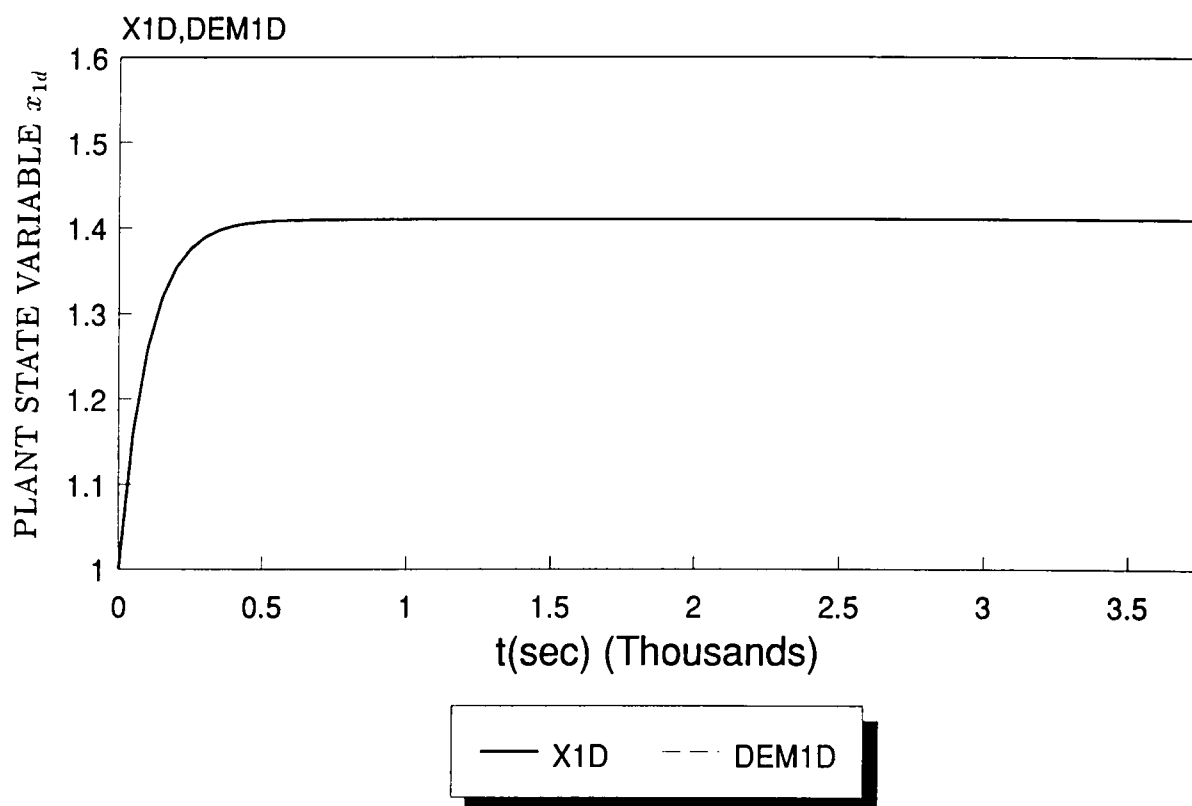


Figure 3.6: **Repetitive perturbation in the plant.** Even in the case of a repetitive perturbation in some of the parameters of the plant (ax_1^2) the plant is kept in a given set point.

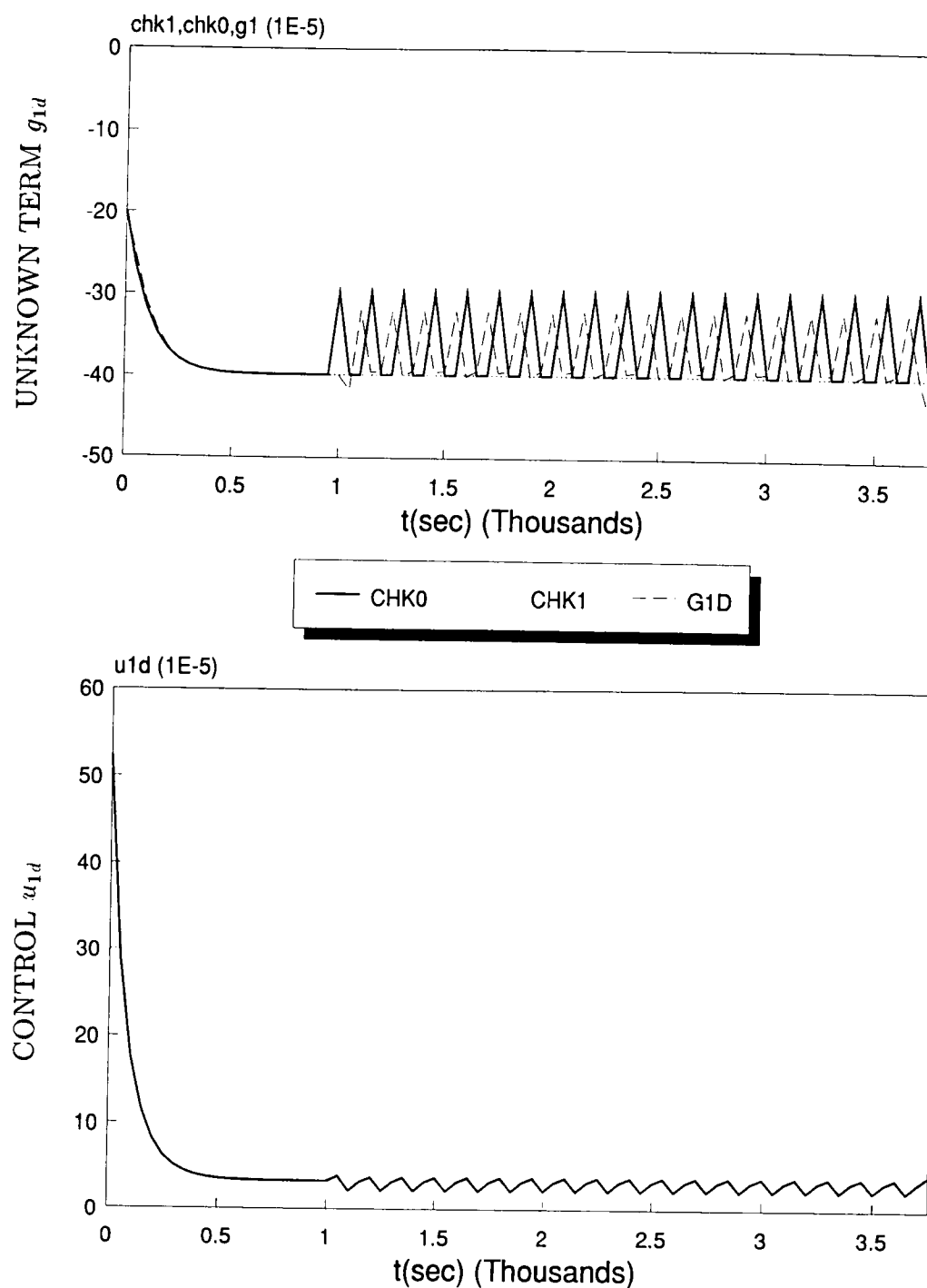


Figure 3.7: **Control for repetitive perturbation.** When $chk_1 = ax_1^2$ changes with a repetitive perturbation the term g_{1d} clearly follows the perturbation, while a small variation is suffered by the control.

3.2.2 Proportional Integral Differential Controller

A conventional PID control can be easily implemented for our simple system.

The control would be

$$u_1 = u_{1F} + k_P(d_1 - x_1) + k_D(\dot{d}_1 - \dot{x}_1) + k_I \int_0^t (d_1 - x_1) d\tau \quad (3.25)$$

where u_{1F} is the equilibrium value given by eq.(3.22), and k_P, k_D and k_I are constants. Because of the nonlinearity of the system these constants have been calculated using the Ziegler-Nichols step response method [3].

In the case of a single perturbation we can see in Fig. 3.8 that the system is kept in the given set point, though the control oscillates while the perturbation takes place. The control in the case of repetitive perturbations and the demand following problem can be seen in Fig. 3.9 and Fig. 3.10 respectively.

3.2.3 Conclusions

Both the LM control and the PID control are shown to be robust against any perturbation as well as they are capable of following a demand. However, a clear advantage of the LM control is the possibility of the system of reproducing the perturbation. This is very valuable in the case of diagnostics. If we are able to represent all the interesting terms in the form of "unknown terms", we shall be able to diagnose the source of an unsuspected problem .

If we compare the LM control and the PID control in the case of a single perturbation (Figs. 3.4 and 3.8), we do not see any oscillation in the case of the

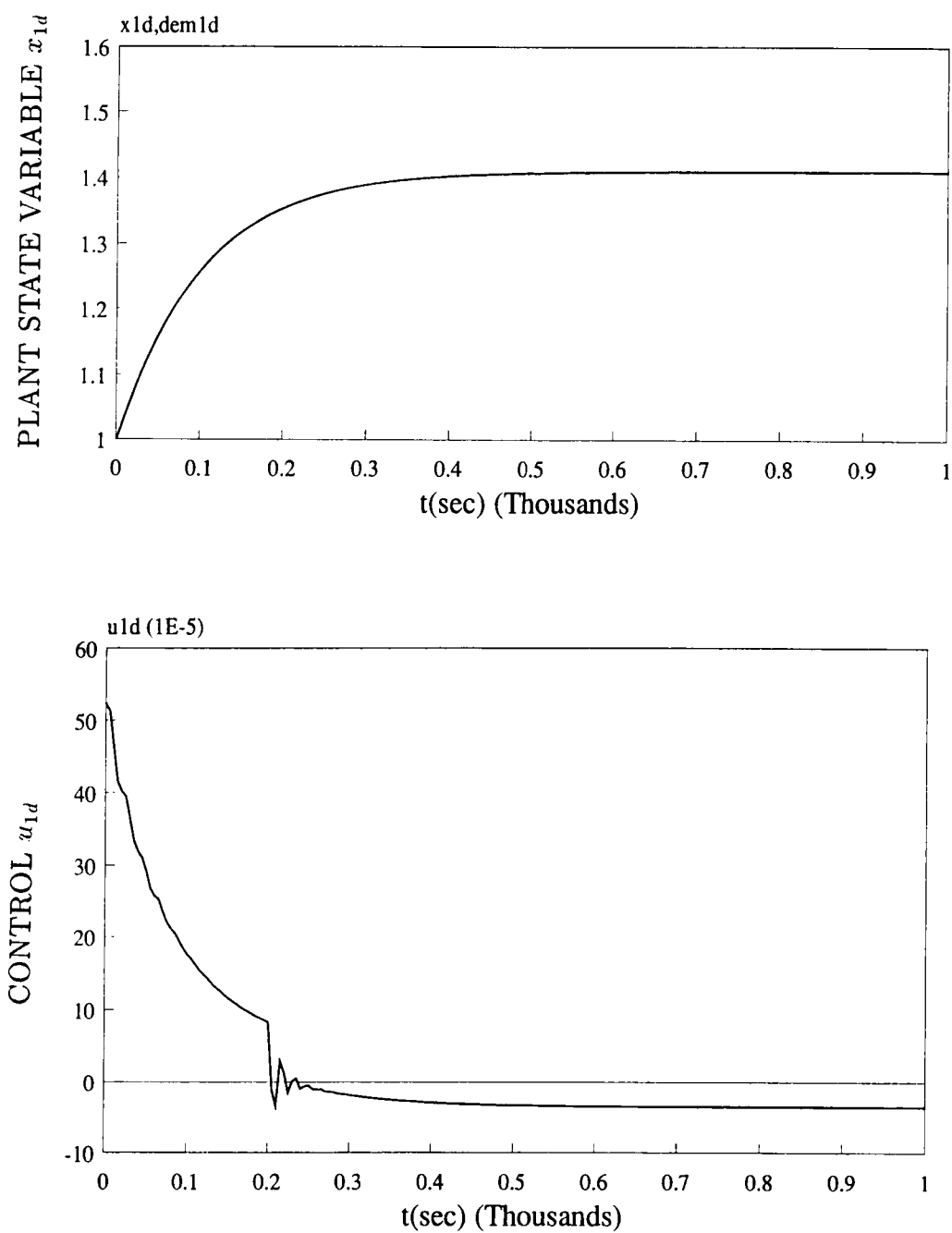


Figure 3.8: PID control behavior in the case of a single perturbation in ax_1^2 .

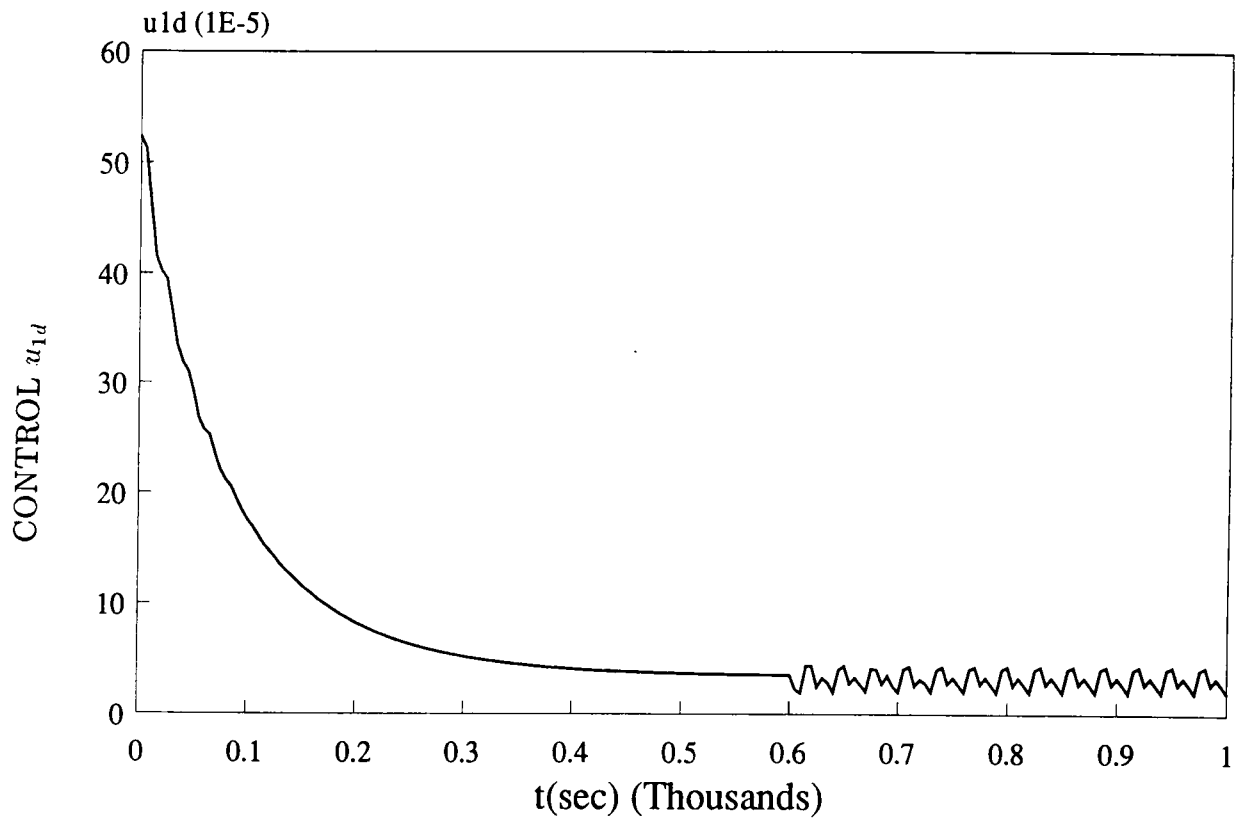


Figure 3.9: PID control behavior in the case of a repetitive perturbation in ax_1^2 .

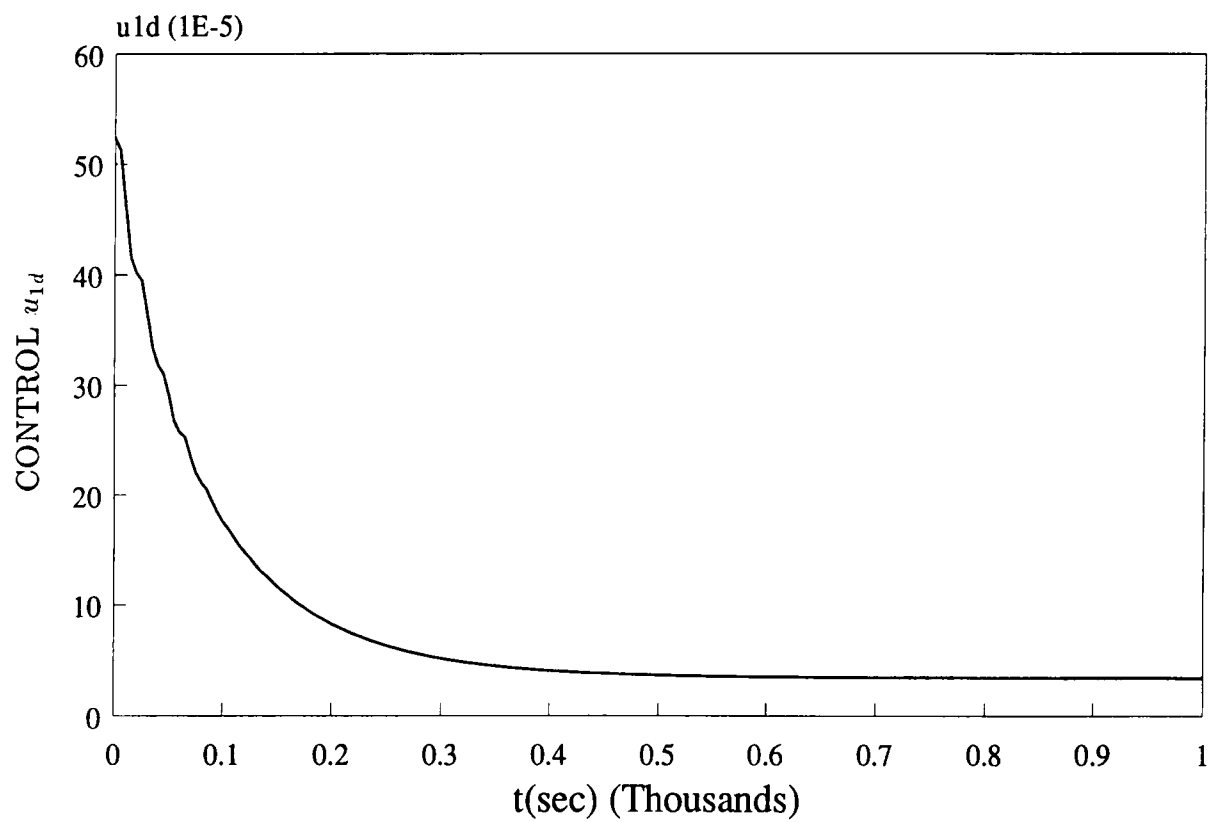


Figure 3.10: PID control behavior in the demand following problem.

LM control.

If we examine closely the control obtained from the LM principle we see that it is nothing more than an "adaptive PID". If we plug in the expression for w_{1a} in the control we see that

$$u_1 = (u_{1F} + A_P(d_1 - x_1) \frac{x_{10}}{m_1} - \frac{1}{2} \frac{m_1}{R_1 \Lambda} \left(-L_1 \int_0^t (m_1 - d_1) d\tau - c_1 \int_0^t (d_1 - x_1) d\tau + L_2(\dot{m}_1 - \dot{d}_1) + c_2(\dot{d}_1 - \dot{x}_1) \right)) \quad (3.26)$$

where the integral, proportional and derivative terms are clearly seen.

To conclude this section, let us see what happens if the proportional terms in the control and unknown term are dropped. As it can be seen in Fig. 3.11 in the case of a stable system the plant overshoots stabilizing after some time, while in the case of an unstable plant the system reaches a limit cycle. In Fig. 3.12 the derivative of the variable is plotted as a function of x_1 , showing how while in the stable case the system goes to an equilibrium point, in the unstable case the system does not reach a local equilibrium but only a global equilibrium.

3.3 Simplified Model of a Liquid Metal Reactor

The PRISM reactor is a new reactor concept, now in development in the United States. The plant [9] is composed by three power blocks interconnected through the electrical grid. Each power block is in turn composed by three modules connected to a common turbine as shown in Fig. 3.13 [10]. Each module

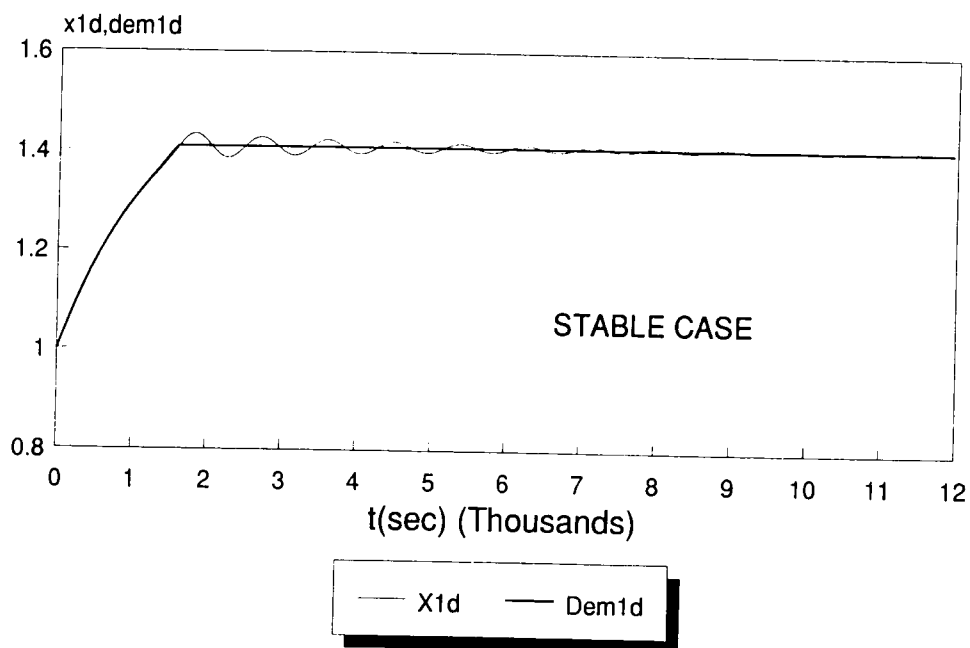
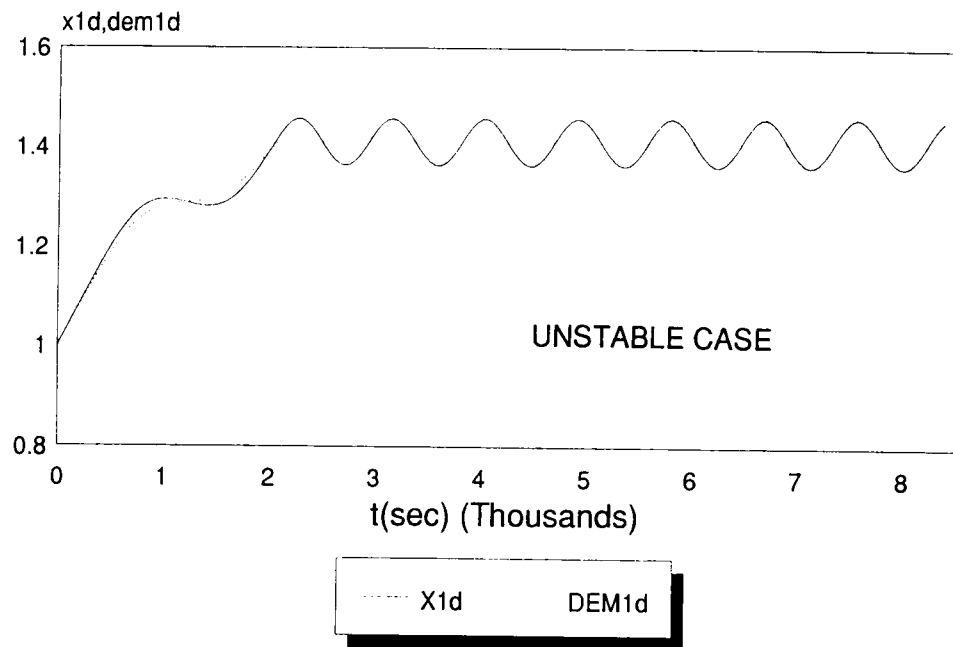


Figure 3.11: LM control without the proportional terms, in the cases of a stable and an unstable system.

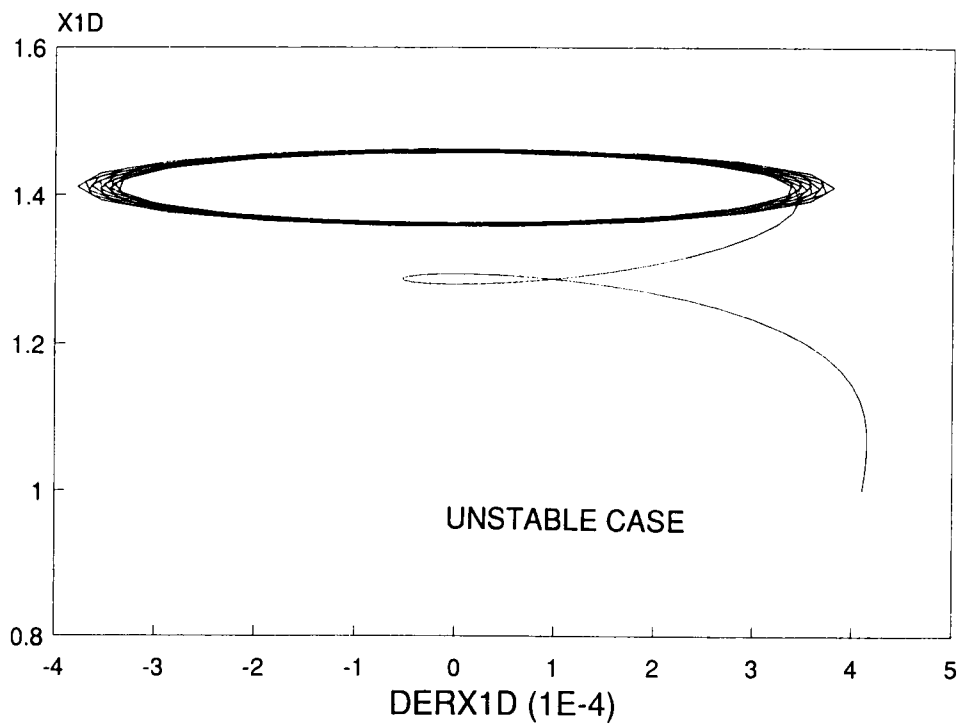
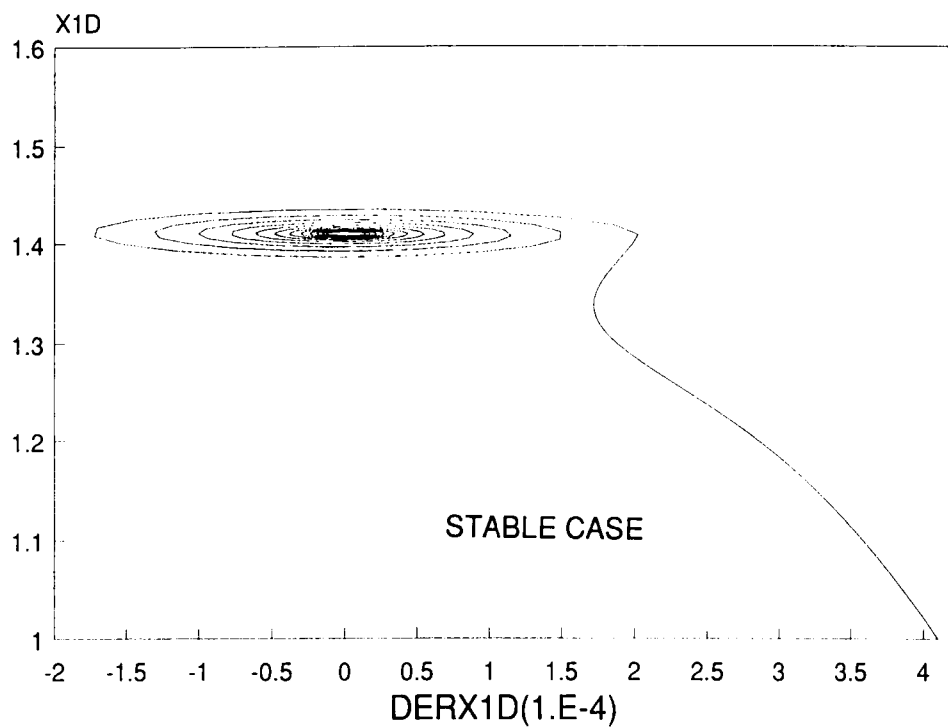


Figure 3.12: $\dot{x}_1$ as a function of x_1 in the case of a LM control with no proportional terms.

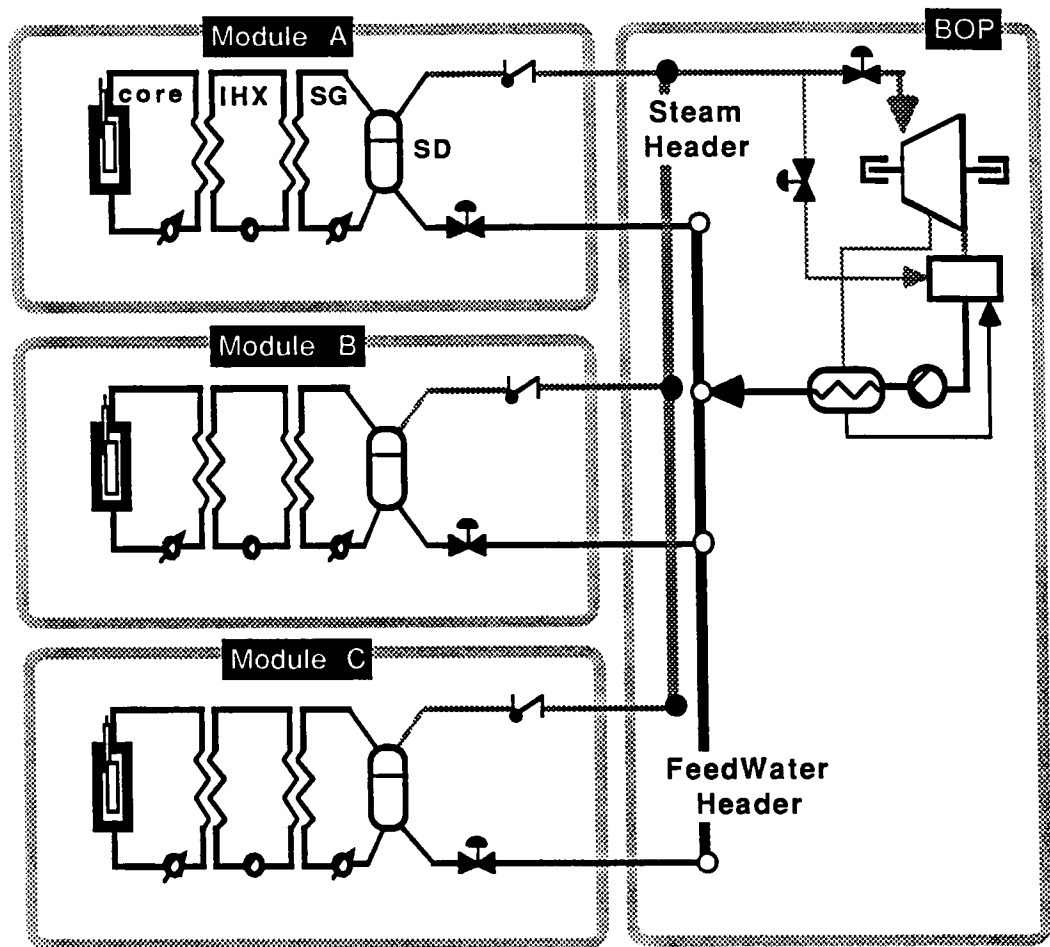


Figure 3.13: Original PRISM model formed by three modules[10].

is composed of a *Liquid Metal cooled Reactor* (LMR), with the corresponding primary, intermediate and tertiary heat transport system. This section (3.3) is mainly extracted from [7].

The model was developed by Rovere [7] under the following assumptions:

1. Thermodynamic equilibrium between phases.
2. Uniform temperature in water side of the steam generator for heat transfer computations.
3. Perfect mixture in steam drum.
4. Perfect separation of phases at steam drum dryers.
5. Negligible kinetic and potential energies.
6. No momentum effects considered.

Steam generators and steam drums of each module were represented by a set of differential equations. These sets of differential equations were coupled through the steam and feedwater headers which represented their boundary conditions.

As our main interest was the development of advanced control algorithms, the model was simplified in collaboration with L.Rovere with the specific purpose of having an easily traceable simulation code. Because of this, only one of the three original power blocks is considered. In this block the three loops were collapsed

into just two, modifying the time constants and heat exchanging coefficients so that we could reproduce the behavior of the original system.

The turbine section was also eliminated, modeling just two control valves considering infinite capacity sink and source : the one that goes from the steam drum to the steam header and the other feedwater valve. The system is shown in Fig. 3.14.

3.3.1 Power Generation Dynamics

The power generation dynamics has been modeled using the standard point kinetics formulation for the neutron population density with only one delayed neutron precursor group

$$\frac{dn}{dt} = \frac{(\rho - 1)n\beta}{\Lambda} + \frac{\beta c}{\Lambda} \quad (3.27)$$

$$\frac{dc}{dt} = \lambda(n - c) \quad (3.28)$$

where

n :normalized neutron density $\frac{N}{N_0}$, where $N_0 = N(t = 0)$.

c :normalized delayed neutron precursor concentration $\frac{C}{C_0}$, where $C_0 = C(t = 0)$.

β :effective precursor yield per fission.

Λ :prompt neutron generation time.

λ :precursor decay constant.

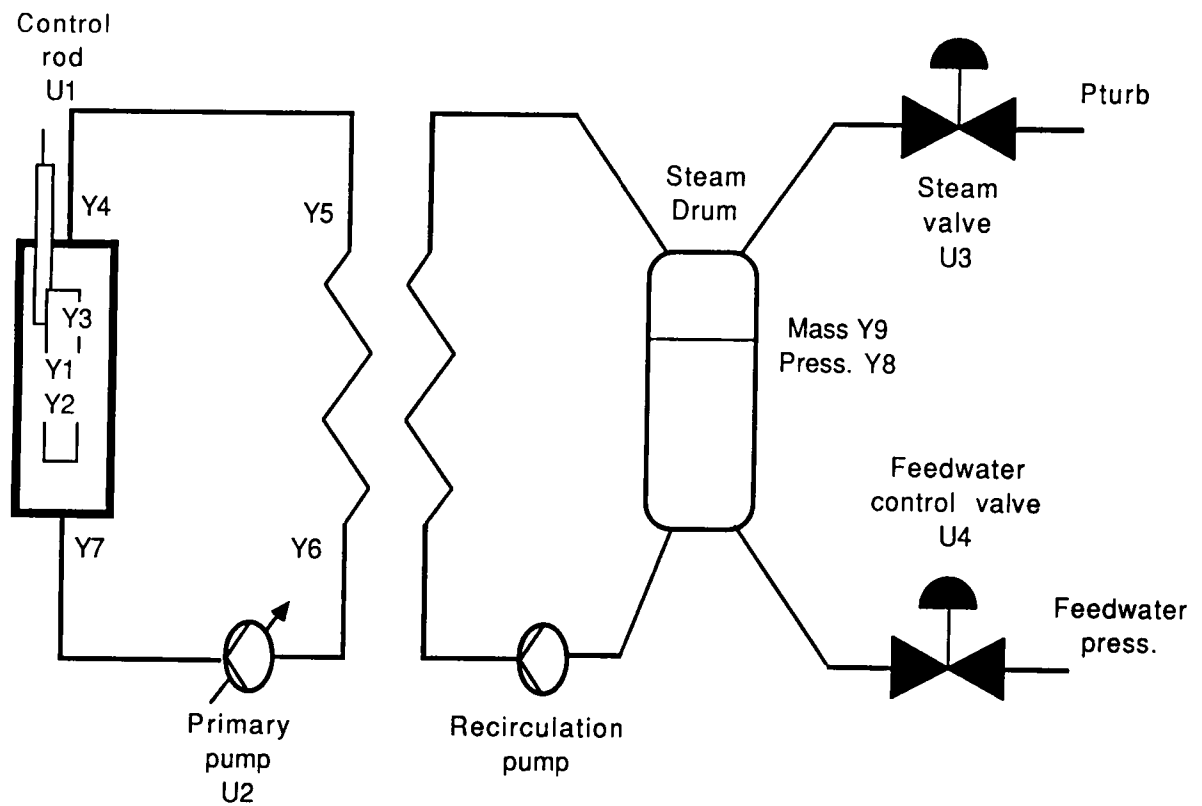


Figure 3.14: **Simplified model.** It has been reduced to one module collapsing the primary and secondary loops of the PRISM reactor in only one. The turbine has been eliminated.

ρ :total core reactivity in dollar units

The total core reactivity ρ is made up by several components:

$$\rho = \alpha_f T_f + \alpha_c T_c + \rho(u) + \rho_0 \quad (3.29)$$

where

α_f :normalized fuel feedback reactivity coefficient(accounting for Doppler and dimensional effects). $\alpha_f = \alpha_f T_r / \beta$ where T_r is a reference temperature.

α_c :normalized coolant feedback reactivity coefficient equal to $\alpha_c T_r / \beta$.

$\rho(u)$:control rod reactivity in dollars.

ρ_0 :initial reactivity in dollars.

T_f :normalized fuel temperature (T_f / T_r).

T_c :normalized coolant temperature (T_c / T_r).

3.3.2 Core Thermohydraulics

The core is represented by two state variables, fuel and coolant temperatures. These variables are obtained from the energy conservation equation for two average nodes. The fuel and the clad are assumed to have an uniform temperature throughout its volume and thus, its energy can be represented by a single variable.

The fuel temperature is determined by a balance between the power being generated inside and the power transferred to the coolant:

$$\frac{dT}{dt} = \frac{\phi_p n}{(MC)_f} - \frac{H_{fc}}{(MC)_f} (T_f - T_c) \quad (3.30)$$

where

n :neutron density.

ϕ_p :neutron density to thermal power factor.

H_{fc} :fuel to coolant heat transfer coefficient.

$(MC)_f$:fuel heat capacity.

The heat transfer coefficient for a sodium-cooled fuel element is computed by the following empirical relationship for the Nusselt number:

$$Nu = 4 + 0.33 \left(\frac{P}{D} \right)^{3.8} \left(\frac{Pe}{100} \right)^{0.86} + 0.16 \left(\frac{P}{D} \right)^{5.0} \quad (3.31)$$

$$Pe = \frac{\rho V D C}{k} \quad (3.32)$$

where

Nu : Nusselt number.

Pe : Peclet number.

V : fluid velocity.

D : hydraulic diameter.

P : wetted perimeter.

C : specific heat.

k : thermal conductivity.

ρ : density.

The overall heat transfer coefficient for nominal operating conditions is obtained from the steady state temperature values of the corresponding components.

The coolant temperature is evaluated from:

$$\frac{dT_c}{dt} = \frac{H_{fc}}{(MC)_c}(T_f - T_c) + \frac{W_p}{M_c}(T_c^{in} - T_c) \quad (3.33)$$

where

M_c : coolant mass in core.

T_c^{in} : core coolant inlet temperature.

3.3.3 Pipes

The transport delays in energy that take place inside the pipes connecting main components are approximated by first order time lags derived from the energy conservation equation for this kind of node:

$$\frac{dT_{out}}{dt} = \frac{W_c}{M_c}(T_{in} - T_{out}) \quad (3.34)$$

where

W_c : coolant mass flowrate.

M_c : coolant mass inside the pipe.

T_{in} : pipe inlet temperature.

T_{out} : pipe outlet temperature.

3.3.4 Steam Generator

In the original model there was an intermediate heat exchanger that has been eliminated for the sake of simplicity. In the steam generator there was an intermediate metal layer between sodium and water that has been also eliminated. All the time constants and coefficients have adjusted so that we can reproduce the original behavior of the system.

The steam generation inside the tubes is assumed to be in equilibrium at all times for given boundary conditions. It is also assumed that the water enters the steam generator at saturation temperature. This hypothesis allows the representation of its secondary side by a single node while the primary side must still be divided in several nodes to represent the transient temperatures more realistically.

For each node the energy conservation equation yields:

$$\frac{dT_{pgi}}{dt} = -\frac{H_{ps}(W_s)}{(MC_{pgi})}(T_{pgi} - T_{sat}) - \frac{W_s}{M_{pgi}}(T_{pgi} - T_i^{in}) \quad (3.35)$$

where

T_{pgi} : i^{th} primary side node fluid temperature.

T_i^{in} : i^{th} node inlet temperature.

T_{sat} : saturation temperature.

H_{ps} : primary secondary heat transfer coefficient.

M_{pgi} : primary side i^{th} node fluid mass.

The heat transferred to the secondary side of the steam generator is computed from the difference between steam primary tubes average temperature for each node and the secondary fluid saturation temperature.

$$Q = \sum_{i=1}^3 H_{ps}(T_{pgi} - T_{sat}) \quad (3.36)$$

where

Q : power transferred to the steam generator secondary fluid.

.

3.3.5 Steam Drum

The mass conservation equation for this component is

$$\frac{dM}{dt} = W_{fw} - W_{sd} \quad (3.37)$$

where

M : total mass of water in steam generator and steam drum.

W_{fw} : feedwater mass flow rate.

W_{sd} : steam flow leaving the steam drum.

The energy conservation equation is expressed in terms of flow across the node boundaries. Considering a homogeneous mixture of water and steam,

$$\frac{dU}{dt} = W_{fw}h_{fw} - W_{sd}h_g + Q \quad (3.38)$$

where

U : total energy of mixture.

h_{fw} : specific enthalpy of feedwater.

h_g : specific enthalpy of steam leaving the steam drum.

The thermodynamic properties of the fluid are evaluated at equilibrium, while the steam-water mixture properties are evaluated from the following relations:

$$v_m = (1 - x)v_f + xv_g \quad (3.39)$$

$$h_m = (1 - x)h_f + xh_g \quad (3.40)$$

$$X = \left(\frac{Vol_{sd}}{M} - v_f \right) \frac{1}{v_{fg}} \quad (3.41)$$

where

v_m, h_m : mixture specific volume and enthalpy.

v_f, h_f : water specific volume and enthalpy.

v_g, h_g : steam specific volume and enthalpy.

v_{fg} : $v_g - v_f$.

Vol_{sd} : steam drum volume.

x : steam quality.

Using these expressions (3.38) can be written as follows:

$$\frac{dP}{dt} = \frac{W_{fw}[h_{fw} - (h_m - v_m P)] - W_{sd}[h_g - (h_m - v_m P)] + Q}{M[\frac{dh}{dp} + P\frac{dv}{dp} - v]} \quad (3.42)$$

The mass flow rates are computed from the pressure difference between adjacent nodes using the following standard relationships:

$$W_{fw} = C_{fw} u_4 \sqrt{\frac{P_{fwh} - P}{v_{fw}}} \quad (3.43)$$

$$W_{sd} = C_{sl} \sqrt{\frac{P - P_{turt}}{v_g}} \quad (3.44)$$

where

C_{fw} : feedwater valve flow conductance.

C_{sl} : steam line flow conductance.

P_{turt} : turbine pressure.

P_{fwh} : feedwater header pressure.

u_4 : feedwater valve fractional opening area.

3.4 Control for a Simplified Model of a Liquid Metal Reactor

In this section we are going to describe the LM control for the simplified model of the LMR reactor integrating the plant equations and the control together, to illustrate the use of just one Lagrangian. In this case the derivative and proportional terms can be neglected. As a second option we shall implement a mixture of methods: a conventional PID control aided by an unknown term calculated with the variational principle, integrating both the plant and the control algorithm independently.

3.4.1 Lagrangian Mechanics-Based Control

In this section we are going to develop a LM control for the simplified model of the PRISM reactor. In we rename the variables according to Fig. 3.14, the model can be written as:

$$\dot{y}_1 = (\alpha_f y_3 + \alpha_c y_4 + \rho_c u_1 + \rho_0) \beta^* y_1 + \beta^* y_2 \quad (3.45)$$

$$\dot{y}_2 = \lambda(y_1 - y_2) \quad (3.46)$$

$$\dot{y}_3 = THN * y_1 + THF * (y_4 - y_3) \quad (3.47)$$

$$\dot{y}_4 = TTC * u_2(y_7 - y_4) + THC * (y_3 - y_4) \quad (3.48)$$

$$\dot{y}_5 = TTPHL * u_2(y_4 - y_5) \quad (3.49)$$

$$\dot{y}_6 = TTPX * u_2(y_5 - y_6) - \frac{HEAT}{TR * MPX * CPNA} \quad (3.50)$$

$$\dot{y}_7 = TTPCL * u_2(y_6 - y_7) \quad (3.51)$$

$$\dot{y}_8 = \frac{B_1 WFW + B_2 WSD + HEAT}{DEN} \quad (3.52)$$

$$\dot{y}_9 = \frac{WFW - WSD}{MSDR} \quad (3.53)$$

where

$y_1 : n(t)$

$y_2 : c(t)$

$y_3 : T_f$

$y_4 : T_c$

y_5 : Fluid inlet temperature to the heat exchanger

y_6 : Fluid outlet temperature from the heat exchanger

y_7 : Inlet temperature to the core.

y_8 : Mass in the steam drum

y_9 : Pressure in the steam drum

The rest of the constants can be found in Appendix A. The model we are going to use for our control is:

$$\dot{m}_1 = (\alpha_c y_4 + \rho_c u_1 + \rho_0) \beta^* m_1 + g_1 = H_1 \quad (3.54)$$

$$\dot{m}_4 = -TTCu_2m_4 - THCm_4 + g_4 = H_4 \quad (3.55)$$

$$\dot{m}_8 = \frac{B_1WFW + B_2WSD}{DEN} + g_8 = H_8 \quad (3.56)$$

$$\dot{m}_9 = \frac{WFW - WSD}{MSDR} = H_9 \quad (3.57)$$

where the m'_i 's are the corresponding model state variables, and g'_i 's are the unknown terms.

We are going to consider that the signals received from the plant are y_{si} for $i = 1, 4, 8, 9$. These signals are delayed with characteristical transfer functions as

$$\dot{y}_{si} = \frac{1}{\tau_i}(y_i - y_{si}), i = 1, 4, 8, 9 \quad (3.58)$$

where τ_i are constants that characterize the delay time. In order to solve the problem we are going to make the model exhibit the same set of delays pertaining the signals, hence we write

$$\dot{m}_{si} = \frac{1}{\tau_i}(m_i - m_{si}), i = 1, 4, 8, 9 \quad (3.59)$$

where m_{si} are the delayed model state variables.

As we are dealing with a continuous time problem, we can use only just one Lagrangian (see chapter 2)

$$\begin{aligned} \mathcal{L} = & -1/2 \sum_i Q_i (y_{si} - m_{si})^2 - 1/2 \sum_j K_j (g_j - g_{jF})^2 \\ & -1/2 \sum_l R_{l=1}^4 (u_l - u_{lF})^2 + \sum_i w_i (\dot{m}_i - H_i) \\ & + \sum_i z_i (\dot{m}_{si} - \frac{1}{\tau_i} (m_i - m_{si})) \end{aligned} \quad (3.60)$$

where $i = 1, 4, 8, 9; j = 1, 4, 8$, Q_i, K_j, R_l are constants, w_i, z_i are the adjoints and g_{jF}, u_{lF} are the dynamic equilibrium values of the unknown terms and the controls respectively. These will be determined by the condition of reducing the differential equations describing the conjugate momenta to the source terms.

The Euler-Lagrange equations for the system will yield the following expressions for the adjoint and unknown terms:

$$\dot{z}_i = -Q_i(y_{si} - m_{si}) + \frac{z_i}{\tau_i} \quad (3.61)$$

$$\dot{w}_i = -\frac{z_i}{\tau_i} \quad (3.62)$$

$$g_j = g_{jF} - \frac{w_j}{K_j} \quad (3.63)$$

and the controls

$$u_1 = \left(\frac{\dot{d}_1 - g_1}{\beta^* m_1} - \alpha_c m_4 - \rho_0 \right) \frac{1}{\rho_c} \quad (3.64)$$

$$u_2 = \frac{-THC m_4 + g_4 - \dot{d}_4}{m_4 + TTC} \quad (3.65)$$

$$u_3 = \frac{-B_1 \dot{d}_9 FW * MS DR + FW(\dot{d}_8 - g_8) DEN}{(B_1 + B_2) FT * FW} \quad (3.66)$$

$$u_4 = \frac{-B_2 \dot{d}_9 FT * MS DR + FT(\dot{d}_8 - g_8) DEN}{(B_1 + B_2) FT * FW} \quad (3.67)$$

where $d_i, i = 1, 4, 8, 9$ are the demands, and

$$FT = CTW \sqrt{\frac{m_8 - P_{turt}/P_R}{WG}} \quad (3.68)$$

$$FW = CFW \sqrt{\frac{PFWH}{P_R} - m_8} \quad (3.69)$$

The performance of the control algorithm is illustrated in several figures. As it can be seen in Fig 3.15 the power and the coolant temperature follow a given demand through the transient until they reach steady state at 30% of their original values. Though the control is receiving a delayed signal y_{si} , it is able to reproduce the original signal y_i, m_i . In Fig 3.16 we see how the pressure and the mass in the steam drum are kept constant.

As it can be seen in Fig 3.17 and Fig. 3.18 the unknown terms follow exactly their counterparts in the plant, this is

$$chk_1 = \alpha_f y_3 \beta^* y_1 + \beta^* y_2 \quad (3.70)$$

$$chk_4 = TTCu_2 y_7 + THC y_3 \quad (3.71)$$

$$chk_8 = HEAT/DEN \quad (3.72)$$

The controls can be seen in Fig. 3.19 and 3.20

3.4.2 Proportional Integral Differential Controller with Unknown Terms

In this section we shall describe how to implement the unknown terms using any kind of control, in this case a PID. Let us consider just the neutronics to make the equations simpler. The equation describing the neutronics can be rewritten as

$$\dot{x}_1 = (\alpha_f(x_3 - x_{30}) + \alpha_c(x_4 - x_{40} + u_1 - u_{10} - 1))\beta^* x_1 + \beta^* x_2 \quad (3.73)$$

$$\dot{x}_2 = \lambda(x_1 - x_2) \quad (3.74)$$

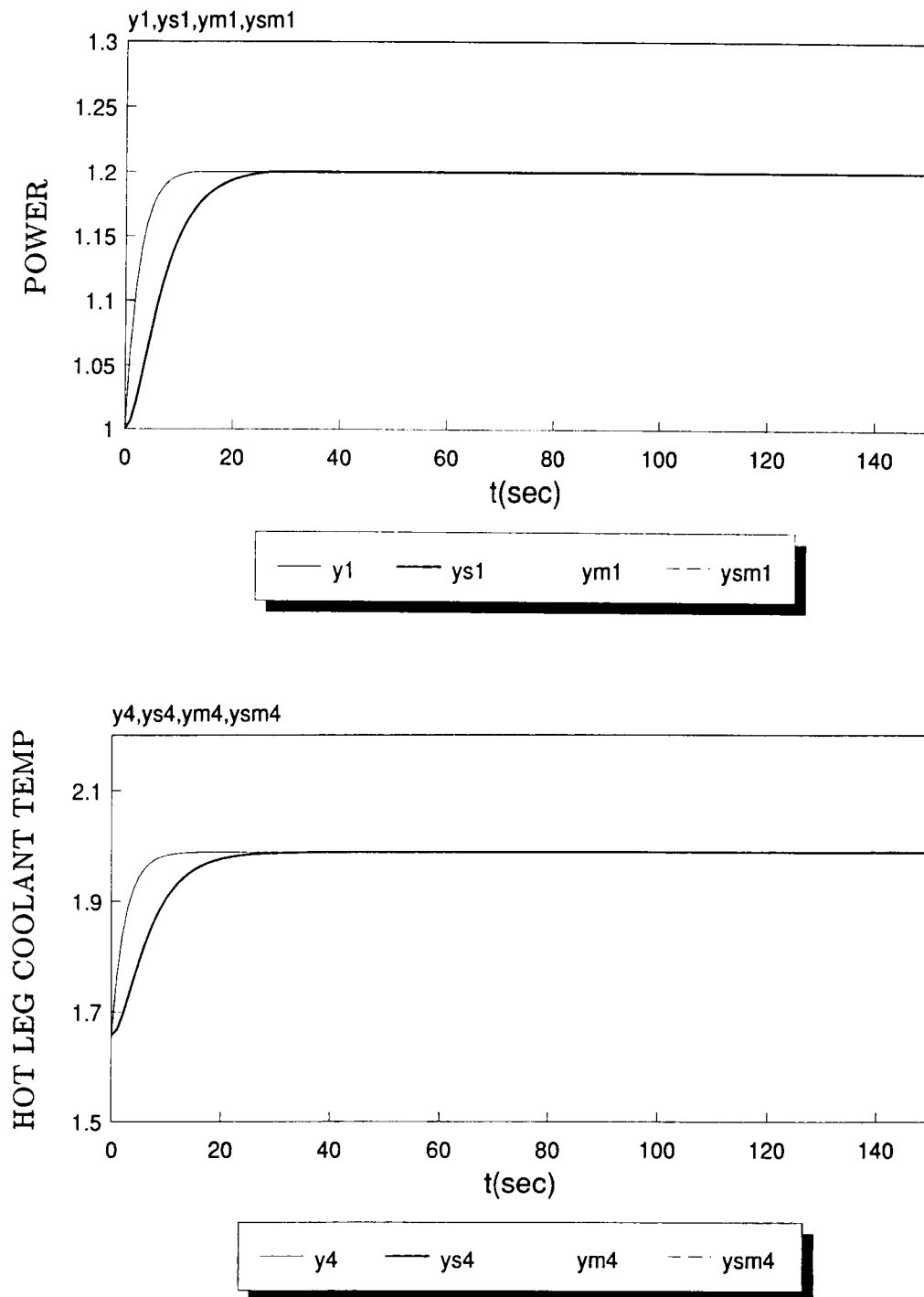


Figure 3.15: Power and Coolant Temperature. The controls reproduce the original signals making the plant follow a demand in power and temperature.

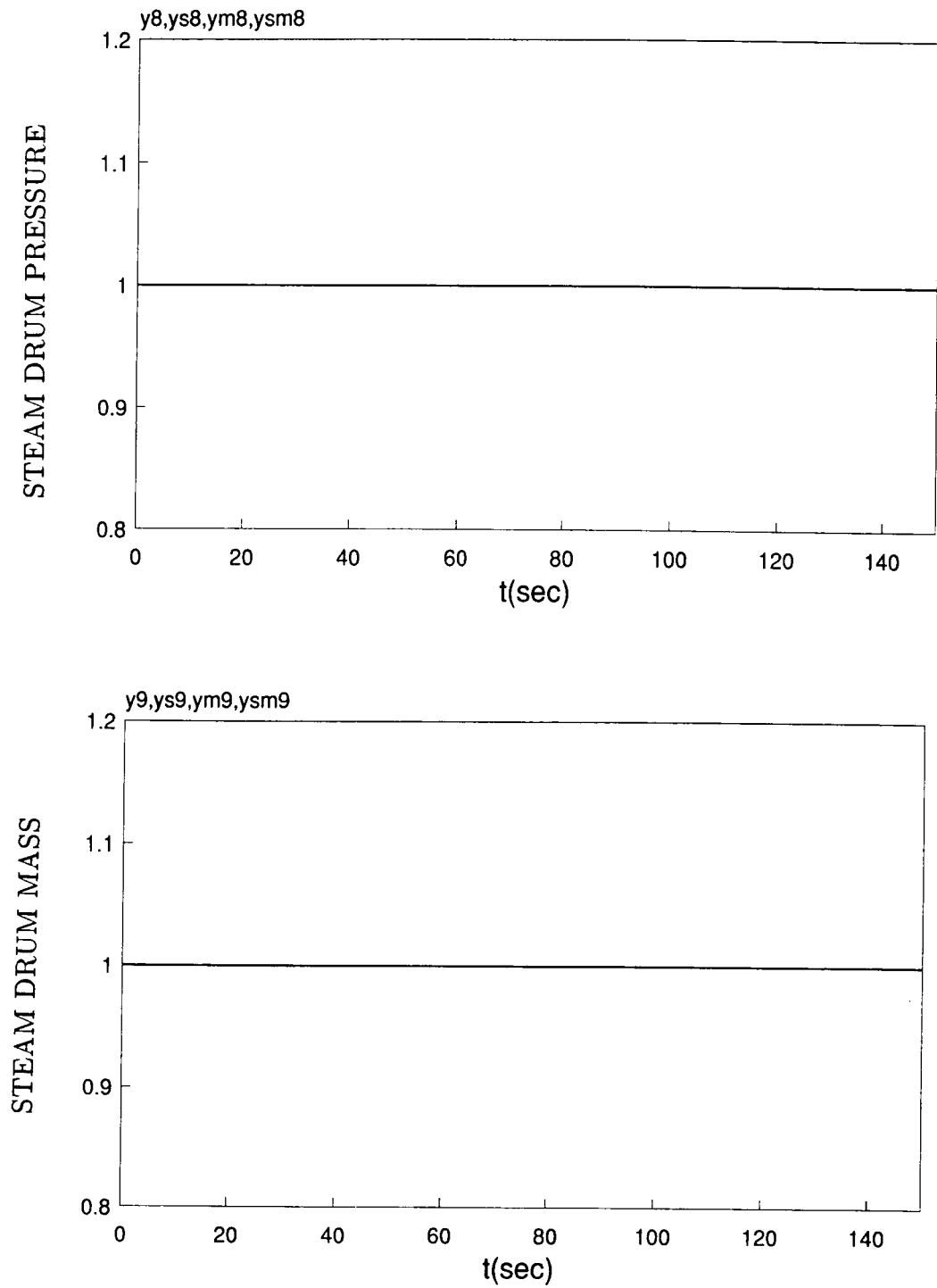


Figure 3.16: Mass and Pressure in the Steam Drum. The mass and pressure in the steam drum follow a given constant demand.

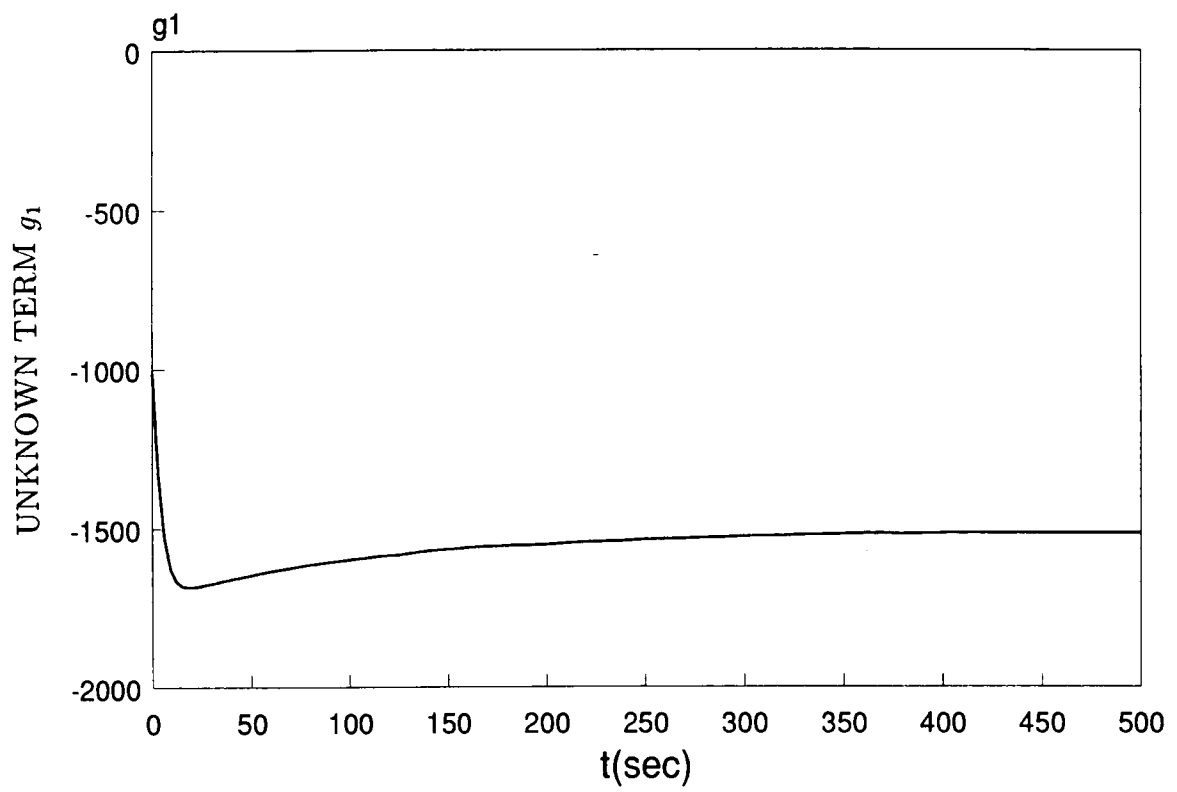


Figure 3.17: Unknown term g_1 . The unknown term g_1 follows exactly chk_1 .

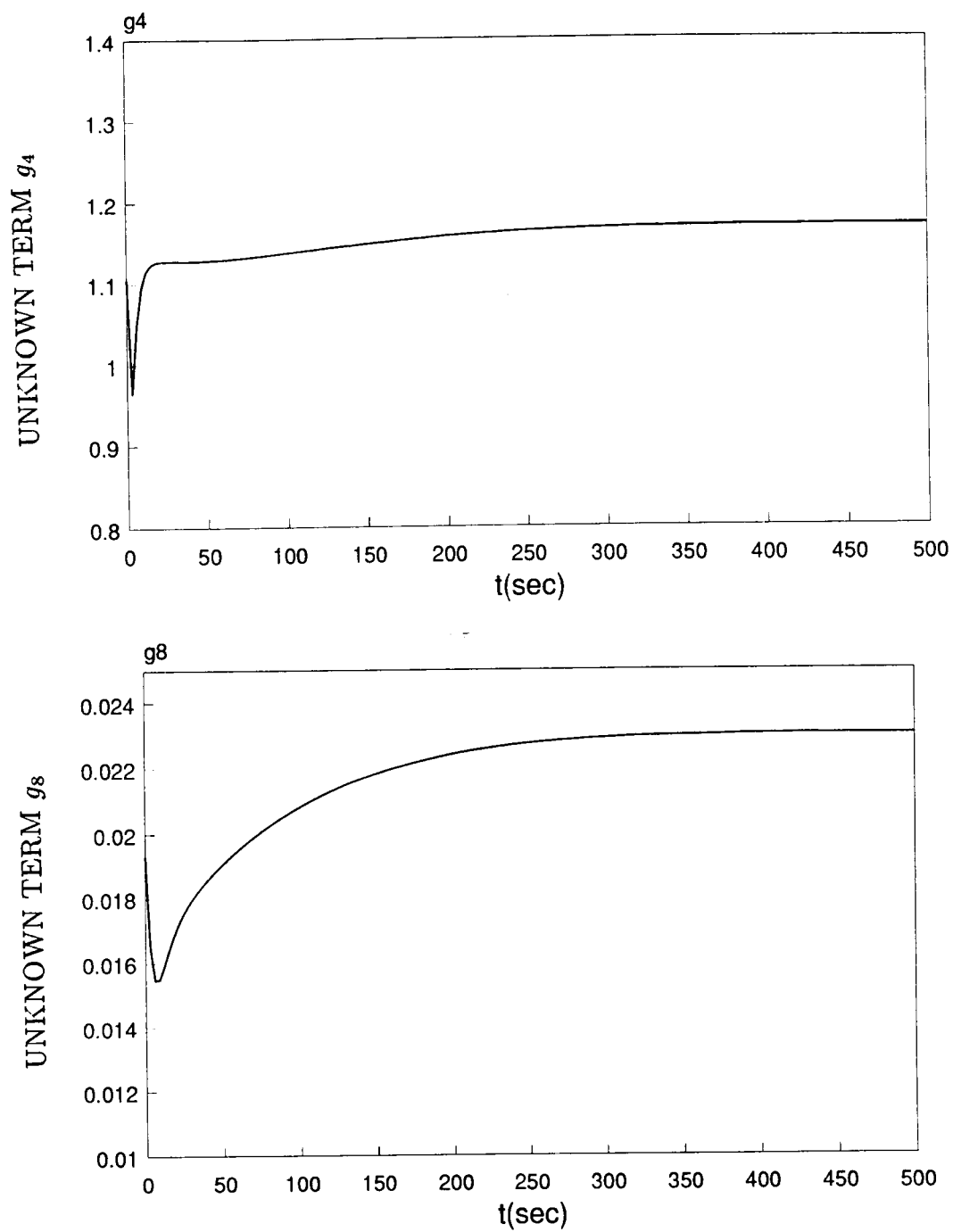


Figure 3.18: Unknown terms g_4 and g_8 for the LMR.

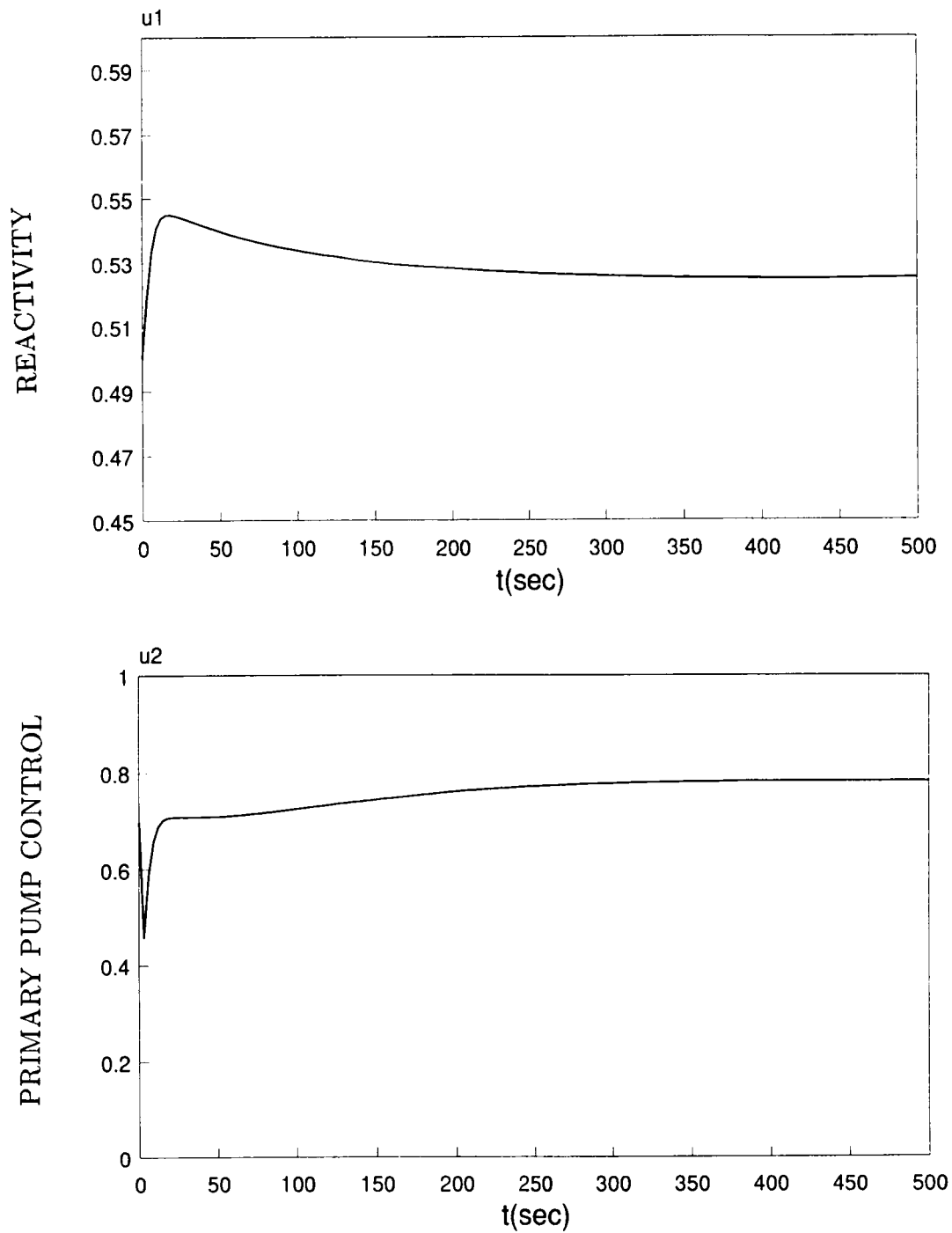


Figure 3.19: Shape of the reactivity control and the primary pump control in the LMR.

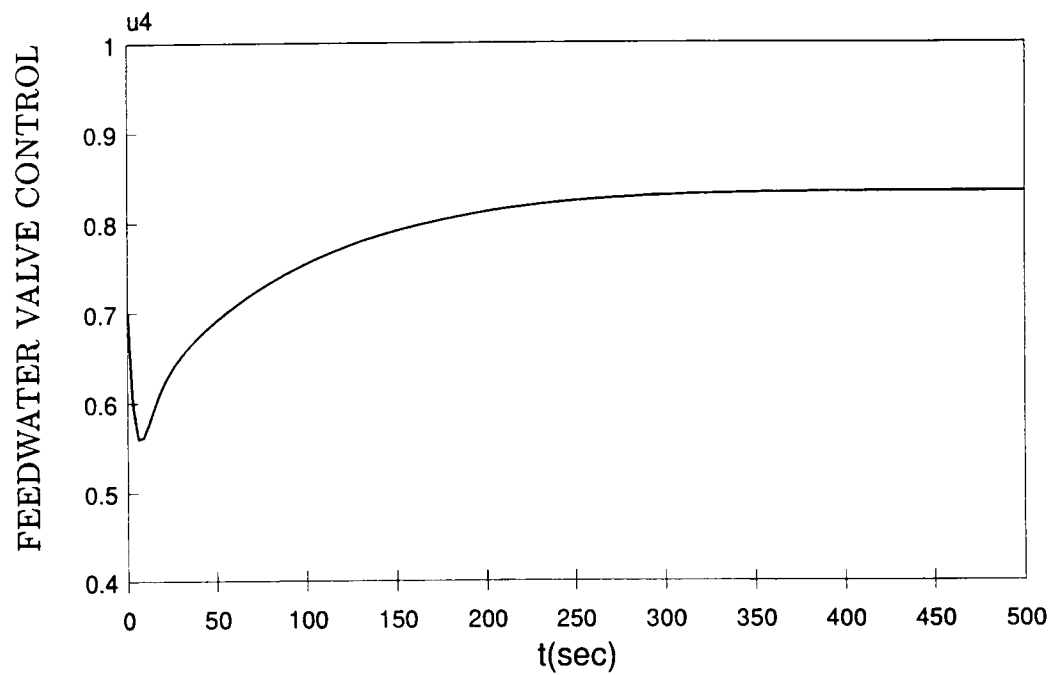
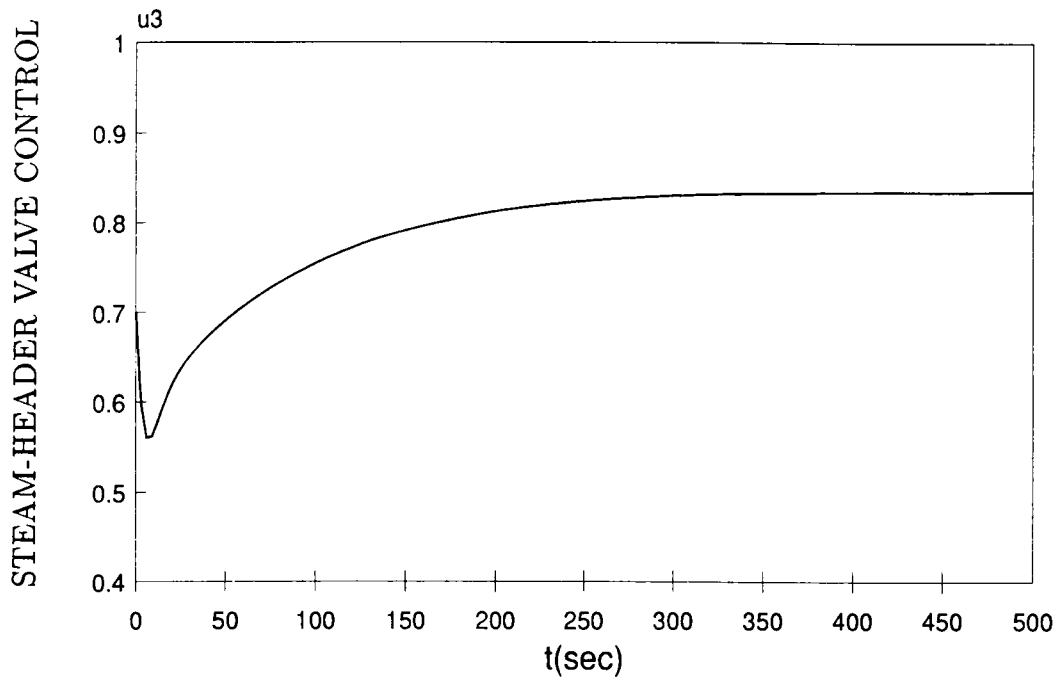


Figure 3.20: Shape of the variation in the steam-header and feedwater valves u_3 and u_4 .

where $\beta^* = \frac{\beta}{\Lambda}$, and x_{30}, x_{40} are the initial values for the fuel and coolant normalized temperatures respectively.

Let us consider the following model

$$\dot{m}_1 = (u_1 - u_{10} - 1)\beta^* m_1 + g_1 \quad (3.75)$$

$$m_{10} = x_{10} \quad (3.76)$$

where g_1 should follow $(\alpha_f(x_3 - x_{30}) + \alpha_c(x_4 - x_{40})\beta^* x_1 + \beta^* x_2)$.

The control will be given by

$$u_1 = u_{1F} + R_P(x_1 - d_1) + R_I \int_0^t (x_1 - d_1) d\tau + R_D(\dot{x}_1 - \dot{d}_1) \quad (3.77)$$

$$u_{1F} = u_{10} + \frac{\dot{d}_{10}}{\beta^* x_{10}} \quad (3.78)$$

where u_{1F} is the control at $t = 0$ that makes $\dot{x}_{10} = \dot{d}_{10}$, and u_{10} represents the initial value in equilibrium.

To implement the unknown term we shall only need one Lagrangian, that can be written as:

$$\begin{aligned} \mathcal{L} = & -1/2Q_1(m_1 - x_1)^2 - 1/2Q_{1A}(\dot{m}_1 - \dot{x}_1)^2 - c_{1B}(\dot{m}_1 - \dot{d}_1)(m_1 - x_1) \\ & -1/2K_1(g_1 - g_{1F})^2 + w_1(\dot{m}_1 - (u_1 - u_{10} - 1)\beta^* m_1 - g_1) \end{aligned} \quad (3.79)$$

From $\partial \mathcal{L} / \partial g_1 = 0$ we get

$$g_1 = g_{1F} - \frac{w_1}{K_1} \quad (3.80)$$

where g_{1F} is not specified yet. From

$$\frac{\partial \mathcal{L}}{\partial m_1} - \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{m}_1} \right) = 0 \quad (3.81)$$

we get the expressions for the conjugate momentum

$$p_1 = w_1 - Q_{1A}(\dot{m}_1 - \dot{x}_1) - c_{1B}(m_1 - x_1) \quad (3.82)$$

$$\dot{p}_1 = -Q_1(m_1 - x_1) - c_{1B}(\dot{m}_1 - \dot{d}_1) \quad (3.83)$$

while the expression for g_{1F} is given by the condition that the differential equation for the conjugate momentum is reduced just to the source terms.

$$g_{1F} = \dot{d}_{10} - \beta^*(u_1 - u_{10} - 1)m_1 \quad (3.84)$$

If we use eq.3.77 and name $k_D = Q_{1A}/K_1$, $k_P = c_{1B}/k_1$ and $k_I = Q_1/k_1$ the final expression for the g_1 will be

$$\begin{aligned} g_1 = & \dot{d}_{10} \left(1 - \frac{m_1}{x_{10}} \right) \\ & - \beta^* m_1 \left(R_P(x_1 - d_1) + R_D(\dot{x}_1 - \dot{d}_1) + R_I \int_0^t (x_1 - d_1) d\tau - 1 \right) \\ & - \frac{p_1}{K_1} - k_D(\dot{m}_1 - \dot{x}_1 - k_P(m_1 - x_1)) \end{aligned} \quad (3.85)$$

As it can be seen in Fig. 3.21 the plant follows the demand with some delay, while the model follows the plant almost exactly. The control is shown in Fig. 3.22, while the unknown term is shown in Fig. 3.23. As it can be seen the algorithm is able to follow the term $chk_1 = (\alpha_f(x_3 - x_{30}) + \alpha_c(x_4 - x_{40})\beta^*x_1 + \beta^*x_2)$ very accurately.

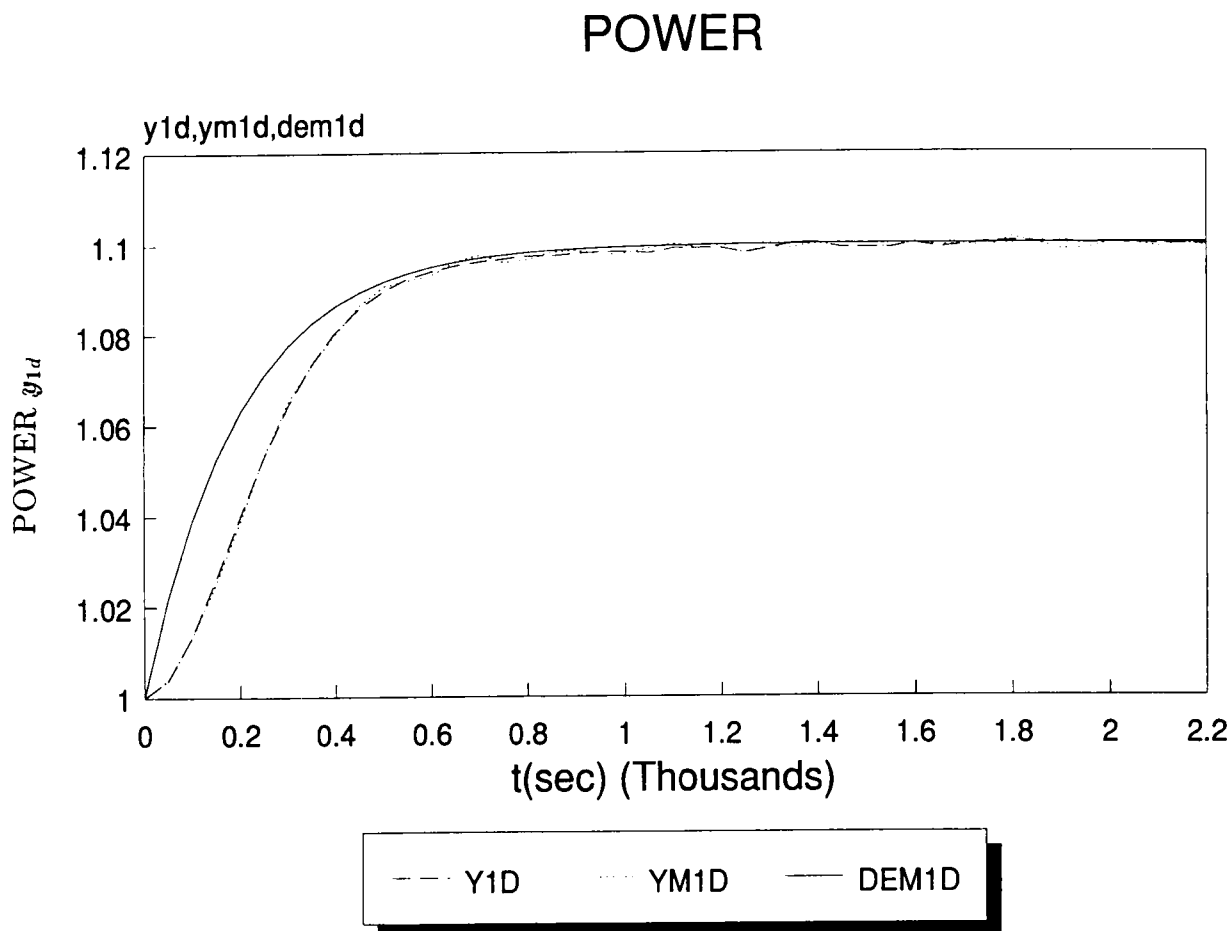


Figure 3.21: LM and PID control for the neutronics. The plant x_{1d} follows the demand dem_{1d} with some delay, until it reaches steady state in the case of solving the plant and the control algorithm independently.

CONTROL

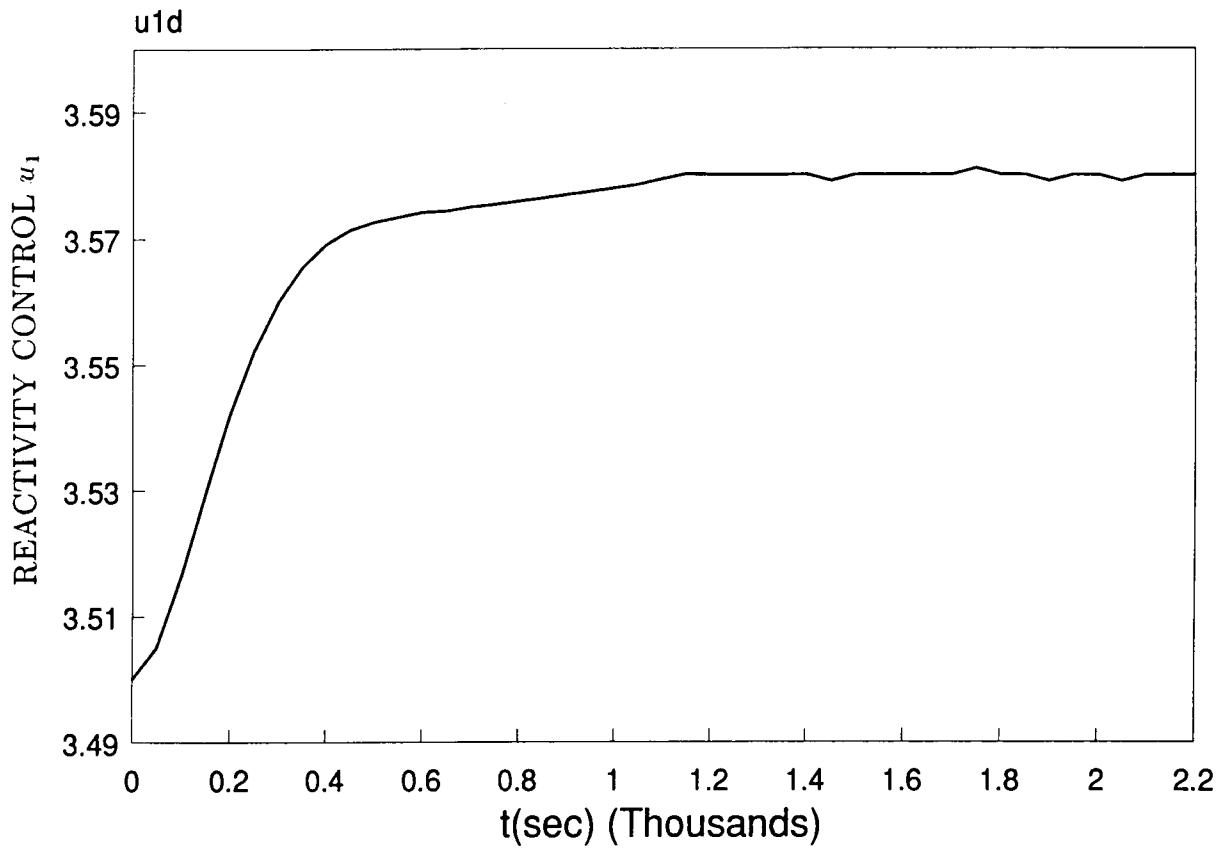


Figure 3.22: PID control for the neutronics.

UNKNOWN TERM

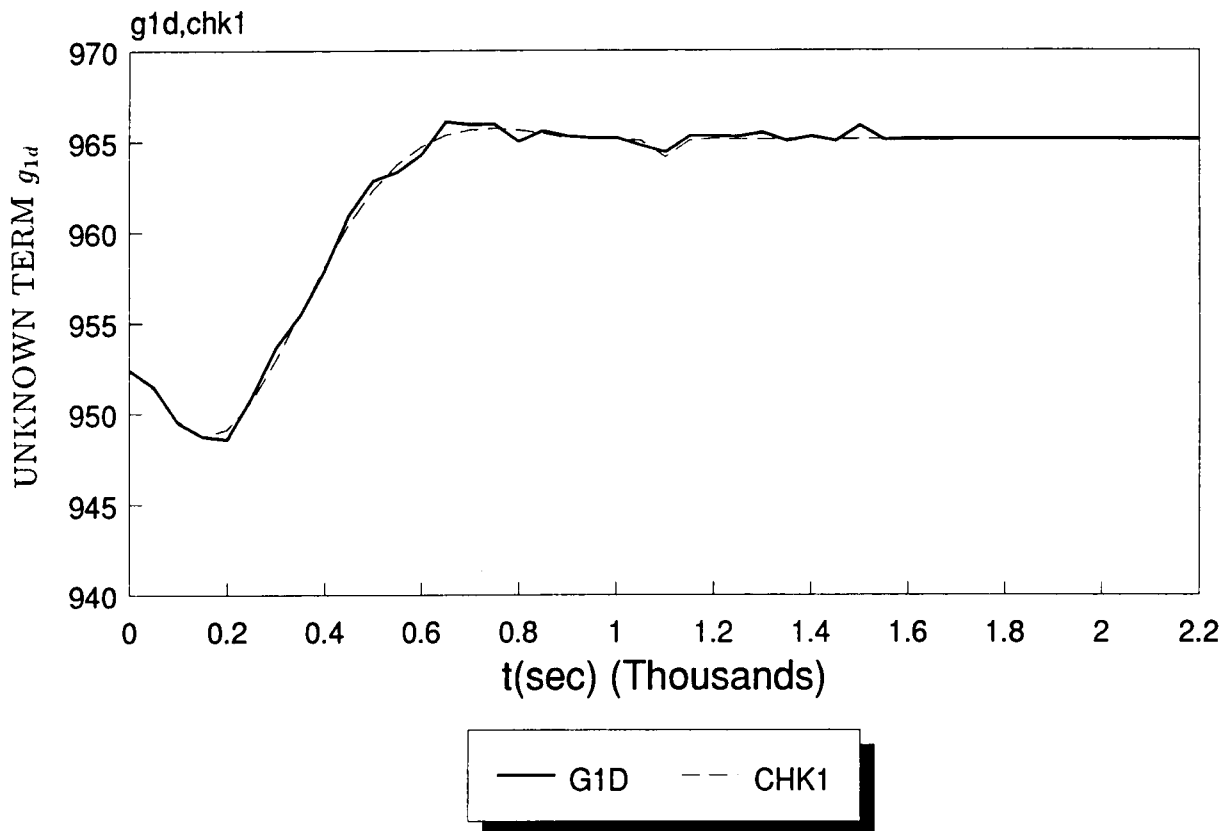


Figure 3.23: Unknown term with PID. The unknown term follows $chk_1 = (\alpha_f(x_3 - x_{30}) + \alpha_c(x_4 - x_{40})\beta^*x_1 + \beta^*x_2)$ with great precision.

CHAPTER 4

CONCLUSIONS AND RECOMMENDATIONS

4.1 Conclusions

Starting from a variational principle we have derived a demand- following, closed-loop control algorithm within the framework of Lagrangian mechanics.

The two main challenges facing this work: developing a general PMP based control theory and avoiding the numerical instabilities that arise when we integrate the plant model and the control algorithm independently have been successfully met.

The control of the neutronics loop in the LMR reactor, which has been simulated by 9 nonlinear coupled ordinary differential equations, has been achieved with a control algorithm based on an internal model of the plant in which the set of describing equations contain uncertain terms to account for the unknown parts the actual plant dynamics.

From the present study the following conclusions have been drawn:

1. To avoid the numerical integration instabilities, in the integration of the differential equations describing the evolution of the conjugate momenta, one has to eliminate in their equations all the terms but the source terms.

This procedure eliminates the instabilities observed in previous PMP-based formalisms.

2. In the case of solving the model describing the plant and the control algorithm independently, the Lagrangian must contain terms in the time derivatives of the signals, demands and models. Otherwise the integration of the differential equations becomes hopelessly unstable. This point illustrates the reason behind the unstable behavior of previous PMP-based control algorithms, where the cost function (the potential function in the Lagrangian formalism) only contained terms in the functions and not on their derivatives.
3. The algorithm is stable against repetitive, unanticipated disturbances, which are tracked with the uncertain term methodology. This feature is of obvious interest for plant diagnostic purposes.
4. In the present algorithm we can replace the Lagrangian for the control part of the algorithm by a simple PID controller, and use only the Lagrangian for the determination of the uncertain term. This result is of relevance for the nuclear industry, as it would make possible to enhance existing technology with the diagnostic capabilities inherent to the use of uncertain terms.

4.2 Recommendations

The following recommendations are suggested:

1. Extension of the algorithm to include noise-contaminated signals.
2. Application of the algorithm to a hierarchical scheme for the control of multi-nodal power plants, containing several reactor cores and steam generators connected to a single steam turbine.

REFERENCES

REFERENCES

1. L.S.Pontryagin et al., "The Mathematical Theory of Optimal Processes", John Wiley and Sons Inc., New York, 1962.
2. D.G.Luenberger., "Introduction to Dynamic Systems", John Wiley and Sons Inc., New York, 1979.
3. Karl J. Astrom, "Adaptive Control", Addison-Wesley, 1989.
4. R.Rosztoczy, A.P.Page and L.E.Weaver, "Reactor Kinetics and Control", Proc. of Symposium, TID-7662, April 1964.
5. Carlos March-Leuba and R.B.Pérez, " Optimal Control Theory of Load-Following and Parameter-Tracking of Nonlinear Systems: An Application of Pontryagin Maximum Principle to Reactor Dynamics", ORNL/TM-10662, December 1987.
6. C.March-Leuba, " Study and Development of Advanced Control Techniques for Nuclear Reactors and Robots", Dissertation for the Doctor of Philosophy Degree, The University of Tennessee, Knoxville, May 1989.
7. Luis A. Rovere, " A Multimodular Reactor Model for Supervisory Control Studies", Internal Report, ORNL ACTO program, 1988.
8. J.A. Llamas, R.B.Pérez and C.March-Leuba, "Closed Loop Optimal Control: Application to Nonlinear Nuclear Reactor Systems"; 7th Power Plant

Dynamics, Control and Testing Symposium, Volume 2, pages 75.1-75.13; Knoxville, Tennessee, May 1989.

9. Dayal, Y., "Advanced PRISM Plant Control System", 7th Power Plant Dynamics, Control and Testing Symposium, May 1989, Knoxville, Tennessee, USA.
10. P.J.Ostaduy, C.R.Brittain and L.A.Rovere, " Supervisory Control in a Distributed, Hierarchical Architecture for a Multimodular LMR", IAEA/IWGFR Specialists Meeting, Advanced Control for Fast Reactors, June 1989, Argonne National Laboratory, Argonne, IL, USA.

APPENDICES

APPENDIX A

GENERAL OPTIMAL CONTROL THEORY

OPTIMAL CONTROL THEORY

A.1 Introduction

The general structure of an optimal control problem is very straightforward [2]. In its simplest version, there is a given dynamic system for which input functions can be specified. There is also an objective function whose value is determined by system behavior, and is in some sense a measure of the quality of that behavior. The optimal control problem is that of selecting the input function so as to optimize (maximize or minimize) the objective function.

A.2 The Basic Optimal Control Problem

This section formulates a basic continuous-time optimal control problem and develops the associated Pontryagin maximum principle.

The problem is formulated as follows: one is given a system, defined on a fixed time interval $0 \leq t \leq T$

$$\dot{x}(t) = f(x(t), u(t)) \quad (\text{A.1})$$

a (fixed) initial condition

$$x(0) = x_0 \quad (\text{A.2})$$

a set of allowable controls

$$u(t) \in U \quad (\text{A.3})$$

and an objective function

$$J = \psi(x(T)) + \int_0^T l(x(t), u(t)) dt \quad (\text{A.4})$$

to be minimized.

Generally we assume that the state $x(t)$ is n dimensional and the control $u(t)$ is m dimensional. The set of allowable controls U is an important component of the optimal control problem. In some cases, this set is the whole m -dimensional space in which case there is no real constraint on $u(t)$. In other cases, however, this set takes the form of inequalities on the components of $u(t)$.

In the objective function, both ψ and l are real-valued functions of their respective arguments. The term $\psi(x(T))$ is the contribution to the objective of the final state. For example, this form of objective arises if it is desired to control an object so as to attain maximum velocity in a given time. The integral term represents a contribution that accumulates over time. Such a term arises, for example, if the objective is to minimize total fuel expenditure of a machine or to maximize total production in a production facility. A specific problem may have either ψ or l identically equal to zero (but not both).

In our problem the unknown is the control function $u(t)$ on $0 \leq t \leq T$. Once this function is specified, it determines, in conjunction with the system equation and the initial condition, a unique state trajectory $x(t)$, $0 \leq t \leq T$. This trajectory and the control function then determine a value of J according to the objective function. The problem is to find the control function, satisfying the constraint, which leads to the largest possible value of J .

A.2.1 The Modified Objective and the Hamiltonian.

In many respects optimal control is a natural extension of the maximization of a function of a single variable. The conditions for maximization are derived by considering the effect of small changes near the maximum point.

To characterize an optimal control, we shall trace out the effect of an arbitrary small change in $u(t)$ and require that it be nonimproving for the objective. That is, we start with the assumption that the control function $u(t)$ is optimal; we take a small change in $u(t)$ and determine the corresponding change in the objective J . This change should be negative (nonimproving) if the original $u(t)$ is optimal. However, because a change in $u(t)$ also changes $x(t)$, it is difficult to carry out this plan and directly determine the net influence on the value of the objective. Therefore, a somewhat indirect approach is helpful. The trick that is used is to adjoin to J some additional terms, which sum to zero. In particular we form a *modified objective function*

$$\bar{J} = J - \int_0^T \lambda(t)^T [\dot{x}(t) - f(x(t), u(t))] dt \quad (\text{A.5})$$

The term in brackets is zero for any trajectory. The coefficient n-vector $\lambda(t)$ is at this point arbitrary. It is clear, however, that for any choice of λ the value of $\bar{J}$ is the same as that of J for any $x(t)$ and $u(t)$ satisfying (A.1). We can therefore consider the problem of maximizing $\bar{J}$ rather than J . The flexibility in the choice of $\lambda(t)$ can be used to make the problem as simple as possible.

For the convenience we define the *Hamiltonian function*

$$H(\lambda, x, u) = \lambda^T f(x, u) + l(x, u) \quad (\text{A.6})$$

In terms of the Hamiltonian, the modified objective takes the explicit form

$$\bar{J} = \psi(x(T)) + \int_0^T \{H(\lambda(t), x(t), u(t)) - \lambda(t)^T \dot{x}(t)\} dt \quad (\text{A.7})$$

The Hamiltonian is therefore fundamental for consideration of this modified objective.

Suppose now that a nominal control function $u(t)$, satisfying the constraints $u(t) \in U$, is specified. This determines a corresponding state trajectory $x(t)$. Now we consider a small change in the control function to a new function $v(t) \in U$. This change is small in the sense that the integral of absolute value of the difference is small (for each component of the control); that is

$$\int_0^T |u_i(t) - v_i(t)| dt \leq \epsilon \quad (\text{A.8})$$

for each i , where ϵ is very small. Therefore, the actual change can be large over a very short interval of time.

This new control leads to a new trajectory, which we write as $x(t) + \delta x(t)$. The change $\delta x(t)$ is very small for all t because the state depends (essentially) on the integral of the control function.

If we define $\delta \bar{J}$ as the corresponding change in the modified objective, we have

$$\delta \bar{J} = \psi(x(T) + \delta x(T)) - \psi(x(T))$$

$$+ \int_0^T [H(\lambda, x + \delta x, v) - H(\lambda, x, u) - \lambda^T \delta \dot{x}] dt \quad (\text{A.9})$$

An integration by parts yields

$$\begin{aligned} \int_0^T \lambda_T \delta \dot{x} dt &= \lambda(T)^T \delta x(T) - \lambda(0)^T \delta x(0) \\ &\quad - \int_0^T \dot{\lambda}^T \delta x dt \end{aligned} \quad (\text{A.10})$$

Therefore, we have

$$\begin{aligned} \delta \bar{J} &= \psi(x(T) + \delta x(T)) - \psi(x(T)) - \lambda(T)^T \delta x(T) + \lambda(0)^T \delta x(0) \\ &\quad + \int_0^T [H(\lambda, x + \delta x, v) - H(\lambda, x, u) + \dot{\lambda}^T \delta x] dt \end{aligned} \quad (\text{A.11})$$

We now approximate this expression to first order (that is, to the order of ϵ) by using differential expressions for small differences. The final expression will be

$$\begin{aligned} \delta \bar{J} &= [\psi_x(x(T)) - \lambda(T)^T] \delta x(T) + \lambda(0)^T \delta x(0) \\ &\quad + \int_0^T [H_x(\lambda, x, u) + \dot{\lambda}^T] \delta x dt \\ &\quad + \int_0^T [H(\lambda, x, v) - H(\lambda, x, u)] dt + p(\epsilon) \end{aligned} \quad (\text{A.12})$$

where $p(\epsilon)$ denotes terms that are of smaller order than ϵ . This then is the general expression for the change in $\bar{J}$ resulting from an arbitrary change in $u(t)$. We next simplify this expression by proper selection of the function $\lambda(t)$.

A.2.2 The Adjoint Equation.

Note that $\delta x(0) = 0$, since a change in the control function does not change the initial state. Thus, the second term on the right hand side of (A.12) is always

zero. We shall select $\lambda(t)$ to make all other terms vanish except for the last integral. This is accomplished by selecting $\lambda(t)$ as the solution to the *adjoint differential equation*

$$-\dot{\lambda}(t)^T = H_x(\lambda(t), x(t), u(t)) \quad (\text{A.13})$$

with final condition

$$\lambda(T)^T = \psi_x(x(T)) \quad (\text{A.14})$$

Equation (A.13) is a linear (time-varying) differential equation in the unknown (row) vector $\lambda(t)^T$. Associated with this system is a final condition on $\lambda(T)^T$. Thus, one can consider solving the adjoint equation by moving backward in time from T to 0. This determines a unique solution $\lambda(t)$.

With this particular $\lambda(t)$, the expression (A.12) for $\delta \bar{J}$ becomes simply

$$\delta \bar{J} = \int_0^T [H(\lambda, x, v) - H(\lambda, x, u)] dt + p(\epsilon) \quad (\text{A.15})$$

Since $\lambda(t)$, $x(t)$, and $u(t)$ are known and are independent of $v(t)$, this expression gives a simple direct way to calculate the approximate consequence of a change to a new control function $v(t)$. We can use this expression to deduce the condition for optimality.

If the original control function u is optimal, it follows that for any t

$$H(\lambda, x, v) \leq H(\lambda, x, u) \quad (\text{A.16})$$

for all $v \in U$.

A.2.3 The Maximum Principle.

Suppose $u(t) \in U$ and $x(t)$ represent the optimal control and state trajectory for the optimal control problem (A.1). Then there is an adjoint trajectory $\lambda(t)$ such that together $u(t)$, $x(t)$ and $\lambda(t)$ satisfy

$$\dot{x}(t) = f(x(t), u(t)) \quad (\text{A.17})$$

$$x(0) = x_0 \quad (\text{A.18})$$

$$-\dot{\lambda}(t)^T = \lambda(t)^T f_x(x(t), u(t)) + l_x(x(t), u(t)) \quad (\text{A.19})$$

$$\lambda(T)^T = \psi_x^*(x(T)) \quad (\text{A.20})$$

For all t , $0 \leq t \leq T$, and all $v \in U$

$$H(\lambda(t), x(t), v) \leq H(\lambda(t), x(t), u(t)) \quad (\text{A.21})$$

where H is the Hamiltonian

$$H(\lambda, x, u) = \lambda^T f(x, u) + l(x, u) \quad (\text{A.22})$$

This set of conditions can be regarded as a set of equations. The last condition, the maximum condition, is essentially a set of m static equations.

A.3 Problems with Terminal Constraints

The basic optimal control problem already discussed is referred to as a free endpoint problem since the final value of the state vector is completely arbitrary.

There are, however, many problems in which the terminal value of the state vector is constrained in some manner.

The maximum principle can be extended to such problems. This extension is simple and very natural in terms of the form of the final result.

The only new consideration needed to be made is that since $x_i(T)$ is constrained for $i = 1, 2, \dots, r$, it follows that $\delta x_i(T) = 0, i = 1, 2, \dots, r$ for all possible trajectories. Therefore, it is not necessary to specify all components of $\lambda(T)$ in order to guarantee that the first term on the right-hand side of (A.12) vanishes. Only components corresponding to possibly nonzero components of $\delta x(T)$ must be specified. Accordingly, it is necessary only to require

$$\lambda_i(T) = \psi_x(x(T))_i \quad (\text{A.23})$$

for $i = r + 1, r + 2, \dots, n$. This guarantees that all component products in the first term of (A.12) vanish. Thus, the general rule is: If $x_i(T)$ is constrained, then $\lambda_i(T)$ is free; if $x_i(T)$ is free, then $\lambda_i(T)$ is constrained.

Maximum Principle.

Let $x(t), u(t) \in U$ be an optimal solution to the problem with terminal constraints. Then there is an adjoint trajectory $\lambda(t)$ and a constant $\lambda_0 \geq 0$ with $(\lambda_0, \lambda(t)) \neq 0$ such that together $u(t), x(t), \lambda(t)$ and λ_0 satisfy

$$\dot{x}(t) = f(x(t), u(t)) \quad (\text{A.24})$$

$$x(0) = x_0 \quad (\text{A.25})$$

$$x_i(T) = \bar{x}_i, i = 1, 2, \dots, r \quad (\text{A.26})$$

$$-\dot{\lambda}(t)^T = \lambda(t)^T f_x(x(t), u(t)) + \lambda_0 l_x(x(t), u(t)) \quad (\text{A.27})$$

$$\lambda_i(T) = \lambda_0 \psi_x(x(T))_i, i = r + 1, r + 2, \dots, n \quad (\text{A.28})$$

$$(\text{A.29})$$

For all $t, 0 \leq t \leq T$, and all $v \in U$

$$H(\lambda(t), x(t), v) \leq H(\lambda(t), x(t), u(t)) \quad (\text{A.30})$$

where H is the Hamiltonian

$$H(\lambda, x, u) = \lambda^T f(x, u) + \lambda_0 l(x, u) \quad (\text{A.31})$$

The presence of the constant λ_0 is to account for the degenerate situations. These correspond to $\lambda_0 = 0$. In these cases the objective does not enter the conditions. In well formulated problems, however, $\lambda_0 \geq 0$, and without loss of generality one may then set $\lambda_0 = 1$.

A.4 Free Terminal Time Problems

In some problems the terminal time is not fixed, but is allowed to vary. The terminal time is therefore another variable that can be selected in order to attain the maximum possible objective value. As a matter of fact this is going to be the case we are going to deal with in our algorithm.

The formulation is almost identical to the previous one, except that the terminal time T is not specified. Explicitly, the problem is: Given

$$\dot{x}(t) = f(x(t), u(t)) \quad (\text{A.32})$$

$$x(0) = x_0 \quad (\text{A.33})$$

$$u(t) \in U \quad (\text{A.34})$$

$$x_i(t) = \bar{x}_i, i = 1, 2, \dots, r \quad (\text{A.35})$$

$$(\text{A.36})$$

find $T \geq 0$ and $u(t), 0 \leq t \leq T$ so as to maximize

$$J = \psi(x(T)) + \int_0^T l(x(t), u(t)) dt \quad (\text{A.37})$$

Thus, all of the maximum principle conditions must apply here as well, we require, however, one additional condition from which the unknown value of T can be determined.

The modified objective is

$$\bar{J} = \psi(x(T)) + \int_0^T [l(x, u) + \lambda^T f(x, u) - \lambda \dot{x}] dt \quad (\text{A.38})$$

The important new feature is that the change in the terminal value of the state is not $\delta x(T)$, since the final time itself changes. The new terminal state is actually $x(T) + \delta x(T + dT)$. If we denote the new terminal state by $x(T) + dx(T)$, then to a first order approximation

$$dx(T) = \delta x(T) + \dot{x}(T)dT = \delta x(T) + f(x(T), u(T))dT \quad (\text{A.39})$$

A procedure similar to that used before, and selecting $\lambda^T(t)$ to satisfy the adjoint equation, we get

$$d\bar{J} = [\psi_x(x(T)) - \lambda(T)^T] dx(T)$$

$$\begin{aligned}
& + \left[\lambda^T(T) f(x(T), u(T)) + l(X(T), u(T)) \right] dT \\
& + \int_0^T [H(\lambda, x, v) - H(\lambda, x, u)] dt
\end{aligned} \tag{A.40}$$

The first term vanishes if conditions are imposed on the components $\lambda_i(T), i = r + 1, r + 2, \dots, n$, the same as for the problem with fixed terminal time. The only new condition is that the Hamiltonian must vanish at the final time, i.e.

$$H(\lambda(T), x(T), u(T)) = 0 \tag{A.41}$$

A.5 Discrete-Time Problems

Optimal control can also be formulated for discrete-time systems. The resulting necessary conditions are quite similar to those for continuous-time systems, although their form is slightly weaker than the maximum principle.

The basic discrete-time optimal control consists of a dynamic system (defined for $k = 0, 1, 2, \dots, N - 1$)

$$x(k + 1) = f(x(k), u(k)) \tag{A.42}$$

an initial condition

$$x(0) = x_0 \tag{A.43}$$

and an objective

$$J = \psi(x(N)) + \sum_{k=0}^{N-1} l(x(k), u(k)) \tag{A.44}$$

Note that in this formulation there is no constraint on $u(k)$. Otherwise, the formulation and interpretation are entirely analogous to the continuous-time problem.

The modified objective is in this case taken to be

$$\bar{J} = J - \sum_{k=0}^{N-1} [\lambda(k+1)^T x(k+1) - \lambda(k+1)^T f(x(k), u(k))] \quad (\text{A.45})$$

where the $\lambda(k)$ are yet to be selected. This can be written as

$$\begin{aligned} \bar{J} = \psi(x(N)) + \sum_{k=0}^{N-1} [H(\lambda(k+1), x(k), u(k)) - \lambda(k)^T x(k)] \\ - \lambda(N)^T x(N) + \lambda(0)^T x(0) \end{aligned} \quad (\text{A.46})$$

Following the same procedures as in the previous sections, we come up with the following theorem

Discrete-Time Optimality

Suppose $u(k)$ and $x(k)$, $k = 0, 1, \dots, N$ represent the optimal control and state trajectory for the optimal control problem. Then there is an adjoint trajectory $\lambda(k)$, $k = 0, 1, \dots, N$ such that together $u(k)$, $x(k)$ and $\lambda(k)$ satisfy

$$x(k+1) = f(x(k), u(k)) \quad (\text{A.47})$$

$$x(0) = x_0 \quad (\text{A.48})$$

$$\lambda(k) = \lambda(k+1) f_x(x(k), u(k)) + l_x(x(k), u(k)) \quad (\text{A.49})$$

$$\lambda(N) = \psi_x(x(N)) \quad (\text{A.50})$$

$$H_u(\lambda(k+1), x(k), u(k)) = 0 \quad (\text{A.51})$$

where H is the Hamiltonian.

This is quite analogous to the maximum principle derived previously, except that instead of requiring that the Hamiltonian be maximized, the condition is that the Hamiltonian must have zero derivative with respect to the control variables.

This general pattern applies to the discrete-time analogs of continuous-time problems treated before.

APPENDIX B
MATHEMATICAL BACKGROUND

When we derived the general expression for the unknown and control terms in the case of a single equation, in order to get just the source terms in the differential equations for the momenta we got the following conditions:

$$\frac{\partial \gamma_F}{\partial m_1} + (x_1 - m_1) \frac{\partial B_{PF}}{\partial m_1} + \frac{\rho}{\Lambda} + \frac{A_P}{\Lambda} (d_1 - x_1) = B_P \quad (\text{B.1})$$

$$\frac{m}{\Lambda} \frac{\partial \rho_F}{\partial m_1} + \frac{m_1}{\Lambda} (d_1 - x_1) \frac{\partial A_{PF}}{\partial m_1} + \frac{\rho}{\Lambda} + \frac{A_P}{\Lambda} (d_1 - x_1) = B_P \quad (\text{B.2})$$

$$(\text{B.3})$$

To integrate the set of differential equations we can divide these two equations in the following four:

$$\frac{\partial \gamma_F}{\partial m_1} + \frac{\rho_F}{\Lambda} = \frac{m_1 w_{1a}}{R_1 \Lambda^2} \quad (\text{B.4})$$

$$(x_1 - m_1) \frac{\partial B_{PF}}{\partial m_1} + \frac{d_1 - x_1}{\Lambda} A_{PF} = B_{PF} \quad (\text{B.5})$$

$$\frac{m_1}{\Lambda} \frac{\partial \rho_F}{\partial m_1} + \frac{\rho_F}{\Lambda} = \frac{m_1 w_{1a}}{R_1 \Lambda^2} \quad (\text{B.6})$$

$$\frac{m_1}{\Lambda} (d_1 - x_1) \frac{\partial A_{PF}}{\partial m_1} + \frac{1}{\Lambda} (d_1 - x_1) A_{PF} = B_{PF} \quad (\text{B.7})$$

Equations (B.4) and (B.6) are very easy to integrate. From the first one we get

$$\rho_F = \frac{w_{1a} m_1}{2. \Lambda R_1} + \frac{u_{1F} x_{10}}{m_1} \quad (\text{B.8})$$

and from the second one

$$\gamma_F = g_{1F} + \frac{m_1^2 w_{1a}}{4. R_1 \Lambda^2} + u_{1F} \frac{x_{10}}{\Lambda} \ln\left(\frac{x_{10}}{m_1}\right) \quad (\text{B.9})$$

On the other hand equations (B.5) and (B.7) are coupled. To solve this problem we resort to perturbation theory. We shall consider A_P and B_P as an

equilibrium term plus a first order perturbation:

$$A_P = A^0 + \epsilon A^1 \quad (\text{B.10})$$

$$B_P = B^0 + \epsilon B^1 \quad (\text{B.11})$$

In eq.(B.5) we shall multiply the term in A_P by ϵ , and in eq.(B.7) we shall do the same with the term in B_P . If we only consider till first order perturbations, and call $x_1 - m_1 = y$ and $d_1 - x_1 = \Delta$, we will get the following set of differential equations:

$$-y \frac{\partial B^0}{\partial y} = B^0 \quad (\text{B.12})$$

$$y \frac{\partial B^1}{\partial y} = B^1 - \frac{\Delta}{\Lambda} A^0 \quad (\text{B.13})$$

$$m_1 \frac{\partial A^0}{\partial m_1} + A^0 = 0 \quad (\text{B.14})$$

$$m_1 \frac{\partial A^1}{\partial m_1} + A^1 = B^0 \quad (\text{B.15})$$

that with the initial conditions give:

$$A_P = A_{P0} \frac{x_{10}}{m_1} \quad (\text{B.16})$$

$$B_P = B_{P0} + A_{P0} (d_1 - x_1) \frac{x_{10}}{\Lambda} \ln\left(\frac{x_{10}}{m_1}\right) \quad (\text{B.17})$$

APPENDIX C

ACSL SOURCE CODE

***** "

PROGRAM LCONTROL

" Model of core, primary, secondary and tertiary loops "

" of a multimodular nuclear power plant. "

"* The reactor module is composed by a Liquid Metal Reactor

"(LMR)"

" with the corresponding primary and intermediate heat transport"

" loops. The module also comprises a recirculating steam "

" generator and a steam drum. "

" The module is connected to a common steam header that feeds "

" the turbine. Feedwater system includes pump, feedwater header "

" and preheater. "

" Interfaced to a shared memory block for I/O. "

" The signals from the plant are considered to be delayed "

"*

*" by Luis Rovere and Jose A. Llamas

"*

*"

"* version 5.0

"*

"* Aug 27th, 1989

"*

"*

"*

"*

*****"

"-----"

MACRO KMOD1(M)

" Constants for module 1 "

CONSTANT	ALPF1	=	-15.E-6,	ALPC1	=	7.0E-6
CONSTANT	BETAT1	=	4.0E-3,	ELE1	=	0.42E-5
CONSTANT	RHO01	=	-3.2565,	RHOC1	=	7.E0
CONSTANT	AMBDA1	=	1.0E-2			
CONSTANT	HAFC1	=	5.195122E6,	HAPMX1	=	71.000000E6
CONSTANT	POW01	=	426.E6,	PRES01	=	6.65298
CONSTANT	MSD01	=	6.0E4			

```

    CONSTANT U1= .5, .9, .7, .7

MACRO END $"KMOD1"

"-----"
MACRO KCONT(M)
"  Constants for CONTROLLER "

    CONSTANT      ALPCM  =      7.0E-6
    CONSTANT      BETATM =      4.0E-3,
    CONSTANT      RHOOM  =     -3.2565,
    CONSTANT      AMBDAM =      1.0E-2
    CONSTANT      HAFCM  =    5.195122E6
    CONSTANT      POWOM  =     426.E6,
    ELEM          =    0.42E-5
    RHOCM         =      7.E0

    PRESOM = 6.65298

    CONSTANT      MSDOM  =      6.0E4
    CONSTANT      UOM    = .5, .9, .7, .7
    CONSTANT      UM     = .5, .9, .7, .7

"CONTROL DELAYS"
    CONSTANT      NDELAY = 0.1
    CONSTANT      FDELAY = 0.0
    CONSTANT      TDELAY = 0.0
    CONSTANT      WDELAY = 0.0

MACRO END $"KCONT"
"-----"
MACRO KCOPD(M)
"  Constants for CONTROLLER "

    CONSTANT      ALPCS  =      7.0E-6
    CONSTANT      BETATS =      4.0E-3,
    CONSTANT      RHOOS  =     -3.2565,
    CONSTANT      AMBDAS =      1.0E-2
    CONSTANT      HAFCS  =    5.195122E6
    CONSTANT      POWOS  =     426.E6,
    ELES          =    0.42E-5
    RHOCs         =      7.E0

    PRESOS = 6.65298

    CONSTANT      MSDOS  =      6.0E4
    CONSTANT      UOS    = .5, .9, .7, .7
    CONSTANT      UMS     = .5, .9, .7, .7

MACRO END $"KCOPD"
"-----"
MACRO KCOMD(M)
"  Constants for CONTROLLER "

    CONSTANT      ALPCD  =      7.0E-6
    CONSTANT      BETATD =      4.0E-3,
    CONSTANT      RHOOD  =     -3.2565,
    CONSTANT      AMBDAD =      1.0E-2
    CONSTANT      HAFCD  =    5.195122E6
    CONSTANT      POWOD  =     426.E6,
    ELED          =    0.42E-5
    RHOCd         =      7.E0

    PRESOD = 6.65298

    CONSTANT      MSDOD  =      6.0E4

```

```

CONSTANT      UOD   = .5, .9, .7, .7
CONSTANT      UD    = .5, .9, .7, .7

```

MACRO END \$"KCOMD"

"-----"

MACRO ICMOD(M)

```

AFUEL_M      = ALPF_M*TR / BETAT_M
ACOOOL_M     = ALPC_M*TR / BETAT_M
BELE_M       = BETAT_M / ELE_M

```

" Define state variables initial condition "

```

YO_M(1) = POWO_M/FTP
YO_M(2) = YO_M(1)

YO_M(3) = 550.000/TR
YO_M(4) = 468.000/TR

YO_M(5) = YO_M(4)

YO_M(6) = 354.000/TR

YO_M(7) = YO_M(6)

YO_M(8) = PRESO_M/PR
YO_M(9) = MSDO_M/MSDR

```

MACRO END \$"ICMOD"

"-----"

MACRO ICCONT(M)

```

ACOOOL_M     = ALPC_M*TR / BETAT_M
BELE_M       = BETAT_M / ELE_M

```

" Define state variables initial condition "

```

YO_M(1) = POWO_M/FTP

YO_M(4) = 468.000/TR

YO_M(8) = PRESO_M/PR
YO_M(9) = MSDO_M/MSDR

```

MACRO END \$"ICCONT"

"-----"

MACRO CTRL(M)

PROCEDURAL(U1=UM)

```

" U1(1) = DELAY(UM(1), 0.5, NDELAY,100)"
" U1(2) = DELAY(UM(2), 0.9, FDELAY,100)"

```

```

"    U1(3) = DELAY(UM(3), 0.7, TDELAY,100)"
"    U1(4) = DELAY(UM(4), 0.7, WDELAY,100)"

    U1(1) = UM(1)
    U1(2) = UM(2)
    U1(3) = UM(3)
    U1(4) = UM(4)

```

```

END $"OF PROCEDURAL"
MACRO END $"CTRL"

```

```

"-----"
MACRO PROP(M)
" Calculate thermodynamic properties "

```

```

    PROCEDURAL(VF_M,VG_M,VFG_M,DVF_M,DVG_M,DVFG_M,      ...
               HF_M,HG_M,HFG_M,DHF_M,DHG_M,DHFG_M,      ...
               YT3_M=Y_M)

```

```

    PRES_M      = Y_M(8)*PR

```

```

    CALL WVF ( VF_M,PRES_M)
    CALL WVG ( VG_M,PRES_M)
    CALL WDVF(DVF_M,PRES_M)
    CALL WDVG(DVG_M,PRES_M)
    VFG_M      = VG_M - VF_M
    DVFG_M     = DVG_M - DVF_M

```

```

    DVGH = -0.02

```

```

    CALL WHF ( HF_M,PRES_M)
    CALL WHG ( HG_M,PRES_M)
    CALL WDHF(DHF_M,PRES_M)
    CALL WDHG(DHG_M,PRES_M)
    HFG_M     = HG_M - HF_M
    DHFG_M    = DHG_M - DHF_M

```

```

    CALL TSAT ( T3_M , PRES_M)

```

```

    YT3_M = T3_M/TR

```

```

END $"OF PROCEDURAL"
MACRO END $"PROP"

```

```

"-----"

```

```

MACRO COEFP(M)
PROCEDURAL( DEN_M ,B1_M ,B2_M ,HEAT_M      ...

           = Y_M,                          ...
             VF_M,VG_M,VFG_M,DVF_M,DVG_M,DVFG_M,  ...
             HF_M,HG_M,HFG_M,DHF_M,DHG_M,DHFG_M,  ...
             DVGH, YT3_M                      )

```

```

" Water-Steam mixture properties "

```

```

X_M = ( VOLSD/(Y_M(9)*MSDR) - VF_M ) / VFG_M
VMIX_M = (1.-X_M)*VF_M + X_M*VG_M
HMX_M = (1.-X_M)*HF_M + X_M*HG_M

DHMIP_M = HF_M*
DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2 ...
      +DVF_M/VFG_M) ...
+DHF_M*(1.0-(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M) ...
-HG_M*DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2 ...
+DHG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M-HG_M*DVF_M/VFG_M

DVMIP_M = ...

VF_M*(DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2+DVF_M/VFG_M) ...
+DVF_M*(1.0-(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M) ...
+DVG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M ...

-VG_M*DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2-VG_M*DVF_M/VFG_M

" Compute the eqn. coeff."

DEN_M = Y_M(9)*MSDR*(DHMIP_M+ Y_M(8)*PR*DVMIP_M -
VMIX_M)*PR

B1_M = (HCOND - HMX_M) + VMIX_M * Y_M(8) * PR
B2_M = (HMX_M - HG_M) + VMIX_M * Y_M(8) * PR

HEAT_M = HAPS*( Y_M(6)-YT3_M ) * TR

END $"OF PROCEDURAL "
MACRO END $"COEFP"

"-----"
MACRO COEFM(M)
PROCEDURAL( DEN_M , B1_M , B2_M , HEAT_M ...

= Y_M, ...
VF_M, VG_M, VFG_M, DVF_M, DVG_M, DVFG_M, ...
HF_M, HG_M, HFG_M, DHF_M, DHG_M, DHFG_M, ...
DVGH, YT3_M )

" Water-Steam mixture properties "

X_M = ( VOLSD/(Y_M(9)*MSDR) - VF_M ) / VFG_M
VMIX_M = (1.-X_M)*VF_M + X_M*VG_M
HMX_M = (1.-X_M)*HF_M + X_M*HG_M

DHMIP_M = HF_M* ...
      (DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2 ...
      +DVF_M/VFG_M) ...
+DHF_M*(1.0-(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M) ...

```

```

-HG_M*DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2 ...
+DHG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M-HG_M*DVF_M/VFG_M

DVMIP_M = ...

VF_M*(DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2+DVF_M/VFG_M)...
+DVF_M*(1.0-(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M) ...
+DVG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M ...

-VG_M*DVFG_M*(VOLSD/(Y_M(9)*MSDR)-VF_M)/VFG_M**2-VG_M*DVF_M/VFG_M

" Compute the eqn. coeff."

DEN_M = Y_M(9)*MSDR*(DHMIP_M+ Y_M(8)*PR*DVMIP_M - ...
        VMIX_M)*PR

B1_M = (HCOND - HMIX_M) + VMIX_M * Y_M(8) * PR
B2_M = (HMIX_M - HG_M) + VMIX_M * Y_M(8) * PR

END $"OF PROCEDURAL "
MACRO END $"COEFM"

"-----"
MACRO JOINT(M)

" Compute mass flowrates to and from headers "

WSD_M = CTV*U_M(3)*SQRT( ABS(Y_M(8)-PTURB/PR)/VG_M )

WFW_M = CFW*U_M(4)*SQRT( ABS(PFWH/PR-Y_M(8)))

MACRO END $"JOINT"

"-----"
MACRO PRISEC(M)
PROCEDURAL( YDOT_M
            = Y_M      ,U_M      , ...
              THF_M    ,THC_M    , ...
              HG_M     ,DHG_M    , ...
              B1_M     ,B2_M     , ...
              DEN_M    ,          , ...
              HEAT_M   ,          , ...
              WSD_M    ,          , ...
              WFW_M )

" Compute state variables derivatives for a generic module "

YDOT_M(1) = ( AFUEL_M*Y_M(3) + ACOOL_M*Y_M(4) ...
              + RHOC_M*U_M(1) + RHO0_M ) ...
              *BELE_M *Y_M(1) + BELE_M *Y_M(2)

YDOT_M(2) = AMBDA_M*( Y_M(1) - Y_M(2) )

YDOT_M(3) = THN*Y_M(1) + THF_M*( Y_M(4) - Y_M(3) )

```

```

YDOT_M(4) =   TTC  *U_M(2)*( Y_M(7) - Y_M(4) )      ...
              + THC_M *( Y_M(3) - Y_M(4) )

YDOT_M(5) =   TTPHL*U_M(2)*( Y_M(4) - Y_M(5) )

YDOT_M(6) =   TTPX  *U_M(2)*( Y_M(5) - Y_M(6) )      ...
              - HEAT_M/(TR*MPX*CPNA)

YDOT_M(7) =   TTPCL*U_M(2)*( Y_M(6) - Y_M(7) )

YDOT_M(8) = ( B1_M*WFW_M + B2_M*WSD_M + HEAT_M )/DEN_M

YDOT_M(9) = ( WFW_M - WSD_M )/MSDR

END $"OF PROCEDURAL"
MACRO END $"PRISEC"

"-----"
MACRO MODEL(M)
PROCEDURAL( YDOT_M
              = Y_M      ,U_M      ,      ...
                THC_M    ,          ,      ...
                HG_M     ,DHG_M    ,      ...
                B1_M     ,B2_M     ,      ...
                DEN_M    ,          ,      ...
                HEAT_M   ,          ,      ...
                WSD_M    ,          ,      ...
                WFW_M    ,          ,      ...
                G1,G4,G8)

"  Compute state variables derivatives for a generic module  "

YDOT_M(1) = ( ACOOL_M*Y_M(4) + RHOC_M*U_M(1) + RHO0_M) ...
              *BELE_M *Y_M(1) + G1
YDOT_M(2) = 0.
YDOT_M(3) = 0.
YDOT_M(4) = -TTC  *U_M(2)*Y_M(4) - THC_M * Y_M(4) + G4
YDOT_M(5) = 0.
YDOT_M(6) = 0.
YDOT_M(7) = 0.
YDOT_M(8) = ( B1_M*WFW_M + B2_M*WSD_M )/DEN_M + G8

YDOT_M(9) = ( WFW_M - WSD_M )/MSDR

END $"OF PROCEDURAL"
MACRO END $"MODEL"

" Compute the delayed plant signals"
MACRO PDEL(M)
PROCEDURAL(YSDOT_M= TAU, Y1, YS1 )

YSDOT_M(1)= (Y1(1)-YS1(1))/TAU(1)
YSDOT_M(2)= 0.

```

```

        YSDOT_M(3)= 0.
        YSDOT_M(4)= (Y1(4)-YS1(4))/TAU(4)
        YSDOT_M(5)= 0.
        YSDOT_M(6)= 0.
        YSDOT_M(7)= 0.
        YSDOT_M(8)= (Y1(8)-YS1(8))/TAU(8)
        YSDOT_M(9)= (Y1(9)-YS1(9))/TAU(9)
END $ "OF PROCEDURAL"
MACRO END $ "PDEL"

```

```

" Compute the delayed model signals"
MACRO MDEL(M)
PROCEDURAL(YSDOT_M= TAU, YSM, YM )

```

```

        YSDOT_M(1)=(YM(1)-YSM(1))/TAU(1)
        YSDOT_M(2)= 0.
        YSDOT_M(3)= 0.
        YSDOT_M(4)=(YM(4)-YSM(4))/TAU(4)
        YSDOT_M(5)= 0.
        YSDOT_M(6)= 0.
        YSDOT_M(7)= 0.
        YSDOT_M(8)=(YM(8)-YSM(8))/TAU(8)
        YSDOT_M(9)=(YM(9)-YSM(9))/TAU(9)
END $ "OF PROCEDURAL"
MACRO END $ "MDEL"

```

```

MACRO ADJOIZ(M)
PROCEDURAL(ZDOT = YSM,YS1,Z,QA)

```

```

        ZDOT(1) = -QA(1)*(YS1(1)-YSM(1)) + Z(1)/TAU(1)
        ZDOT(2) = 0.
        ZDOT(3) = 0.
        ZDOT(4) = -QA(4)*(YS1(4)-YSM(4)) + Z(4)/TAU(4)
        ZDOT(5) = 0.
        ZDOT(6) = 0.
        ZDOT(7) = 0.
        ZDOT(8) = -QA(8)*(YS1(8)-YSM(8)) + Z(8)/TAU(8)
        ZDOT(9) = -QA(9)*(YS1(9)-YSM(9)) + Z(9)/TAU(9)
END $ "OF PROCEDURAL"
MACRO END $ "OF ADJOIZ"

```

```

MACRO ADJOIW(M)
PROCEDURAL(WDOT=Z,TAU )

```

```

        WDOT(1)= -Z(1)/TAU(1)
        WDOT(2)= 0.
        WDOT(3)= 0.
        WDOT(4)= -Z(4)/TAU(4)
        WDOT(5)= 0.
        WDOT(6)= 0.
        WDOT(7)= 0.
        WDOT(8)= -Z(8)/TAU(8)
        WDOT(9)= -Z(9)/TAU(9)
END $ "OF PROCEDURAL"

```

MACRO END \$ "OF ADJOIW"

MACRO OPTI(M)

PROCEDURAL (U_M,G1,G4,G8 = Y_M,W,B1_M,B2_M,DEN_M)

FW = CFW*SQRT(ABS(PFWH/PR-Y_M(8)))
FT = CTV*SQRT(ABS(Y_M(8)-PTURB/PR)/VG_M)

G1 = G1F - W(1)/K(1)
G4 = G4F - W(4)/K(2)
G8 = G8F - W(8)/K(3)

U_M(1) =
((DD1-G1)/(BELE_M*Y_M(1))-ACOOOL_M*Y_M(4)-RHO0_M)/RHOC_M
U_M(2) = (-THC_M*Y_M(4)+G4-DD4)/(Y_M(4)*TTC)

DETERM = (B1_M+B2_M)*FT*FW

U_M(3) = (- B1_M*DD9*FW*MSDR + FW*(DD8-G8)*DEN_M)/DETERM
U_M(4) = (B2_M*DD9*FT*MSDR + FT*(DD8-G8)*DEN_M)/DETERM

END \$"OF PROCEDURAL"

MACRO END \$"OPTI"

"-----"
PROGRAM LCONTROL

ARRAY
Y1(9), Y01(9), YDOT1(9), ...
U1(4), ...
UMIN(4), UMAX(4), ...
UOM(4),UM(4),W(9),YM(9),R(4),K(4),Q(9), ...
YOM(9),W0(9),WDOT(9),YDOTM(9), ...
Z0(9),Z(9),YS1(9),YSM(9),QA(9),TAU(9), ...
YSDOT1(9), YSDOTM(9), Y0D(9), Y0S(9),ZDOT(9), ...
U0D(4), UD(4), U0S(4),UMS(4)

INITIAL

CONSTANT	FTP =	426.E6	
CONSTANT	MPHL =	130000.,	MPCL = 130000.
CONSTANT	MCC =	10442E0,	MPG = 100000.
CONSTANT	MPX =	42000.	
CONSTANT	MF =	30000.E0,	HAPS = 5943329.1
CONSTANT	PFWH =	9.,	HCOND = 723254.45
CONSTANT	VPRI =	3235.70,	PTURB = 4.5
CONSTANT	CPNA =	1283.2E0,	CPFUEL = 350.0
CONSTANT	CPSTEL =	450.0E0,	CPH2O = 4186.0
CONSTANT	MSDR =	6.0E4,	POWR = 426.E6
CONSTANT	PR =	6.65298E0,	TR = 282.323
CONSTANT	HR =	2776.264E3	

```

CONSTANT          VOLSD  =          120.0

CONSTANT          CTV    = 97.4311289  ,          CFW    =  499.07955

CONSTANT UMIN =      0.0E0,   0.0E0,   0.0E0,   0.0E0
CONSTANT UMAX =      1.0E0,   1.0E0,   1.0E0,   1.0E0

CONSTANT W0 = 0.,0.,0.,0.,0.,0.,0.,0.,0.
CONSTANT Q =      ...
1.E15,1.E15,1.E15,1.E15,1.E15,1.E15,1.E15,1.E15,1.E15
CONSTANT R = 1.,1.,1.,1.
CONSTANT K = 1.E-9, 1.E-9, 1.E-9, 1.E-9
CONSTANT DD8=0., DD9=0., TCHAR=.3, PERC=.2
CONSTANT Z0= 0.,0.,0.,0.,0.,0.,0.,0.,0.
CONSTANT TAU=10.,10.,10.,10.,10.,10.,10.,10.,10.
CONSTANT QA=      ...
1.E15,1.E15,1.E15,1.E15,1.E15,1.E15,1.E15,1.E15,1.E15

CONSTANT INI = 1.

CONSTANT TMAX = 100.
MININTERVAL MINT = 1.E-4
MAXINTERVAL MAXT = 2.E0
CINTERVAL CINT = 2.E0
ALGORITHM IALG = 2

KMOD1("1")
KCONT("M")
KCOPD("M")
KCOMD("M")

MCF      = MF*CPFUEL
MCCC     = MCC *CPNA
MCPX     = MPX *CPNA
MCPHL    = MPHL*CPNA
MCPCL    = MPCL*CPNA

" Time constants for primary and secondary loops "

THN      = FTP  / (MCF*TR)
TTC      = VPRI / MCC
TTPHL    = VPRI / MPHL
TTPX     = VPRI / MPX
TTPCL    = VPRI / MPCL

THF1     =  HAFCl / MCF
THC1     =  HAFCl / MCCC

" Initial Condition for state variables"
ICMOD("1")
ICCONT("M")
ICCONT("S")
ICCONT("D")

" Time constants for model "

```

```

      THCM      =   HAFCM  /  MCCC

END $" OF INITAL "

DYNAMIC

      DERIVATIVE

PROCEDURAL(G1F,G4F,G8F=YM)

"  Initialize unknowns  "
  IF(INI.EQ.0) GOTO JUMP
    PROP("M")
    COEFM("M")
    JOINT("M")

      G1F  =   -(ACCOOLM*YM(4) + RHOCM*UM(1)  + RHOOM  )      ...
              *BELEM *YM(1)

      G4F  =   +TTC*UM(2)*YM(4) + THCM * YM(4)

      G8F  =   -( B1M*WFWM + B2M*WSDM )/DENM
      INI  = 0.

JUMP.. CONTINUE

END $"OF PROCEDURAL"

      D1= PERC*Y01(1)*(1.-EXP(-TCHAR*T))+Y01(1)
      DD1=PERC*TCHAR*Y01(1)*EXP(-TCHAR*T)

      D4=Y01(4)+PERC*Y01(4)*(1.-EXP(-TCHAR*T))
      DD4=PERC*TCHAR*Y01(4)*EXP(-TCHAR*T)
      PROP("1")
      PROP("M")

      COEFP("1")
      COEFM("M")

      OPTI("M")
      CTRL("M")

      JOINT("1")
      JOINT("M")

      ADJOIW("M")
      ADJOIZ("M")
      MODEL("M")

      MDEL("M")
      PDEL("1")

      PRISEC("1")

```

```

Y1 = INTVC(YDOT1,Y01)
YM = INTVC(YDOTM,Y0M)
YS1= INTVC(YSDOT1,Y0S)
YSM= INTVC(YSDOTM,Y0D)
W  = INTVC(WDOT,W0)
Z  = INTVC(ZDOT,Z0)

END $"OF DERIVATIVES"

TERMT(T.GE.TMAX)

PREPAR (T,Y1,WFW1,U1,WSD1)

END $" OF DYNAMIC"

END $" OF PROGRAM"
      SUBROUTINE TSAT(TEMP,PRES)
C    WATER & STEAM PROPERTIES FUNCTIONS FOR SATURATION CONDITION
C
C    RANGE OF VALIDITY FOR WATER 1.0 - 10.0 MPa , ERROR < 4%
C    "          "          STEAM 3.5 - 10.0 MPa , ERROR < 4%
C
C-----
C    Computes the water saturation temp.( C) as a function
C    of press.(MPa)

      DATA A0/-230.45E0/,A1/-.38927E4/,A2/-.948654E1/

      TEMP = A0 + A1/(ALOG(PRES)+A2)

      RETURN
      END

C-----
      SUBROUTINE WVF(VF,P)

C    Computes the water saturation specific volume (m3/kg) as a C
C    function
C    of press.(MPa)

      DATA A0/.107009E-2/,A1/.637843E-4/,A2/-.683063E-5/,
.      A3/.639983E-6/,A4/-.211171E-7/

      VF=A0+A1*P+A2*P*P+A3*P*P*P+A4*P*P*P*P

      RETURN
      END

C-----
      SUBROUTINE WDFV(DVF,P)

C    Computes the water specific volume (m3/kg.MPa) derivative with
C    respect

```

C to press. as a function of press.(MPa) along the saturation C line

```
DATA A0/.668674E-4/,A1/-.171251E-4/,A2/.315895E-5/,  
A3/-.256982E-6/,A4/.81458E-8/
```

```
DVF=A0+A1*P+A2*P*P+A3*P*P*P+A4*P*P*P*P
```

```
RETURN  
END
```

C-----
SUBROUTINE WHF(HF,P)

C Computes the water saturation specific enthalpy (j/kg) as a C function

C of press.(MPa)

```
DATA A0/.576237E3/,A1/.220352E3/,A2/-.337418E2/,  
A3/.310046E1/,A4/-.109807E0/
```

```
HF=(A0+A1*P+A2*P*P+A3*P*P*P+A4*P*P*P*P)*1.E3
```

```
RETURN  
END
```

C-----
SUBROUTINE WDHf(DHF,P)

C Computes the water specific enthalpy (j/kg.MPa) derivative C with respect

C to press. as a function of press.(MPa) along the saturation C line

```
DATA A0/.241197E3/,A1/-.905910E2/,A2/.174901E2/,  
A3/-.157502E1/,A4/.535953E-1/
```

```
DHf=(A0+A1*P+A2*P*P+A3*P*P*P+A4*P*P*P*P)*1.E3
```

```
RETURN  
END
```

C-----
SUBROUTINE WVG(VG,P)

C Computes the steam saturation specific volume (m3/kg) as a C function

C of press.(MPa)

```
HG=(.276719E4+.273499E2*P-.637456E1*P*P+.472159*P*P*P  
-.151951E-1*P*P*P*P)*1000.
```

```
VG=.00837663+.0207226*P-.000760027*P*P-4.7374E-8*HG  
+2.22965E-14*HG*HG-6.38715E-9*P*HG
```

```
RETURN  
END
```

```

C-----
      SUBROUTINE WDVG(DVG,P)

C   Computes the steam specific volume (m3/kg.MPa) derivative with
C   respect
C   to press. as a function of press.(MPa) along the saturation C
C   line

      DATA A0/-.795834E-1/,A1/.311403E-1/,A2/-.489687E-2/,
      .      A3/.348681E-3/,A4/-.930521E-5/

      DVG=A0+A1*P+A2*P*P+A3*P*P*P+A4*P*P*P*P

      RETURN
      END
C-----
-----

```

```

      SUBROUTINE WHG(HG,P)

C   Computes the steam saturation specific enthalpy (j/kg) as a C
C   function
C   of press.(MPa)

      DATA A0/.276719E4/,A1/.273499E2/,A2/-.637456E1/,
      .      A3/.472159E0/,A4/-.151961E-1/

      HG=(A0+A1*P+A2*P*P+A3*P*P*P+A4*P*P*P*P)*1.E3

      RETURN
      END
C-----

```

```

      SUBROUTINE WDHG(DHG,P)

C   Computes the steam specific enthalpy (j/kg.MPa) derivative C
C   with respect
C   to press. as a function of press.(MPa) along the saturation C
C   line

      DATA A0/.223988E2/,A1/-.102479E2/,A2/.104933E1/,
      .      A3/-.492306E-1/,A4/.581488E-3/

      DHG=(A0+A1*P+A2*P*P+A3*P*P*P+A4*P*P*P*P)*1.E3

      RETURN
      END
C-----

```

```

      SUBROUTINE VFPT(VF,P,T)

C   Computes the water specific volume as a function of P & T C
C   using
C   a linear relationship

```

```
DATA VF0 / 1.1181425E-3/,  
.      DVFT/ 1.2768527E-6/,  
.      DVFP/ -8.993426E-7/
```

```
VF = VF0 + DVFT*(T-172.77866) + DVFP*(P-7.5)
```

```
RETURN  
END
```

```
C-----  
SUBROUTINE DVFPTP(DVFP,P,T)
```

```
C Computes the derivative of the water spec. vol with respect to  
C P
```

```
DATA DVFP0 / -8.993426E-7/
```

```
DVFP = DVFP0
```

```
RETURN  
END
```

VITA

José Antonio Llamas Arjona was born in Madrid, Spain, the 25th of April of 1964. He attended the Universidad Politécnica de Madrid, where he obtained the Degree of Ingeniero Industrial with a major in Nuclear Engineering. In the summer of 87 he spent 4 months working with Dr. R.B.Pérez at the Nuclear Engineering Department at the University of Tennessee, Knoxville, university he enrolled in Fall'88 in the pursue of his M.S. degree.