NASA CR-188288

CSDL-T-1220

HIERARCHICAL MODELING FOR RELIABILITY ANALYSIS USING MARKOV MODELS

by

Arturo Fagundo

May 1994

Bachelor of Science and Master of Science Thesis
Massachusetts Institute of Technology

(NASA-CR-188288) HIERARCHICAL MODELING FOR RELIABILITY ANALYSIS USING MARKOV MODELS B.S./M.S. Thesis - MIT (Draper (Charles Stark) Lab.) 116 p

N94-32862

Unclas

G3/38 0011931

DRAPER
LABORATORY

The Charles Stark Draper Laboratory, Inc.
555 Technology Square, Cambridge, Massachusetts 02139-3563

Hierarchical Modeling for Reliability Analysis Using Markov Models

by

Arturo Fagundo

Submitted to the

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

in partial fulfillment of the requirements

for the degrees of

BACHELOR OF SCIENCE

and

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May, 1994

Signature of Author Arturo Fagundo
Department of Electrical Engineering and Computer Science
May 16, 1994

Certified by Wallace E. Vander Velde
Professor Wallace E. Vander Velde
Thesis Supervisor, Professor of Aeronautics and Astronautics

Certified by Philip S. Babcock III
Dr. Philip S. Babcock, IV
Company Supervisor, C.S. Draper Laboratory

Accepted by____________________________
F.R. Morgenthaler
Chair, Department Committee on Graduate Students

Hierarchical Modeling for Reliability Analysis Using Markov Models

by

Arturo Fagundo

Submitted to the department of Electrical Engineering and
Computer Science on May 16, 1994, in partial fulfillment of the
requirements for the degree of Master of Science
and Bachelor of Science in
Electrical Engineering

Abstract

Markov models represent an extremely attractive tool for the reliability analysis of many systems. However, Markov model state space grows exponentially with the number of components in a given system. Thus, for very large systems Markov modeling techniques alone become intractable in both memory and CPU time.

Often a particular subsystem can be found within some larger system where the dependence of the larger system on the subsystem is of a particularly simple form. This *simple dependence* can be used to decompose such a system into one or more subsystems. A hierarchical technique is presented which can be used to evaluate these subsystems in such a way that their reliabilities can be combined to obtain the reliability for the full system. This hierarchical approach is unique in that it allows the subsystem model to pass multiple aggregate state information to the higher level model, allowing more general systems to be evaluated.

Guidelines are developed to assist in the system decomposition. An appropriate method for determining subsystem reliability is also developed. This method gives rise to some interesting numerical issues. Numerical error due to roundoff and integration are discussed at length. Once a decomposition is chosen, the remaining analysis is straightforward but tedious. However, an approach is developed for simplifying the recombination of subsystem reliabilities. Finally, a real world system is used to illustrate the use of this technique in a more practical context.

Thesis Supervisor: Professor Wallace E. Vander Velde
Company Supervisor: Dr. Philip S. Babcok, IV

Acknowledgments

The past six years of school have been a test of endurance. But now that these times are coming to a conclusion, I would like to thank the following people for their help during this time in my life:

Primarily I would like to thank my father, Angel and mother Elsa for raising me. I would like to thank my brothers Ruben and Angel Jr. for all of their loving harassment over the last 24 years. I would also like to thank the rest of my family, for all of their support.

Special thanks to Amy Mattinson for seeing me through four years of M.I.T. Her constant companionship made my time here more enjoyable. Thanks to her grandfather John Mattinson, Sr. for his spelling help.

All of the members of the Redundancy Management and Operations Analysis group at Draper lab, both past and present for their constant help and encouragement. I would especially like to thank Phil Babcock for pointing me in the right direction. Thanks also to Jeffrey Zinchuk and Eric Davis for their help in producing this thesis.

Professor Vander Velde for his time and effort in following the progress of this thesis.

The people of INROADS/Boston, both students and administrators, for pushing me to do better, especially Walter Murray who was an inspiration as managing director and continues to be an even greater inspiration in retirement.

Marco Morales for the friendly competition I needed to get through some of my toughest classes. Thanks to José L. García for giving me the professional competition I needed to get into M.I.T.

This thesis was prepared at the Charles Stark Draper Laboratory, Incorporated, under contract NAS 9-18426.

Publication of this thesis does not constitute approval by the Charles Stark Draper Laboratory of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

I hereby assign my copyright of this thesis to The Charles Stark Draper Laboratory, Incorporated, Cambridge, Massachusetts.

Arturo Fagundo

Arturo Fagundo

Permission is hereby granted by the Charles Stark Draper Laboratory, Incorporated, to the Massachusetts Institute of Technology to reproduce and to distribute copies of this thesis document in whole or in part.

This thesis is dedicated to
my son, Christopher Fagundo,
without whom I would not have
had the motivation to complete
this work.

Table of Contents

Abstract

Acknowledgements

Table of Contents **9**

List of Figures **11**

List of Tables **13**

1 Introduction **15**

1.1 Background 15

1.2 Decomposition Based on the Law of Total Probability 16

1.3 An Alternative Approach to Hierarchical Modeling 17

1.4 Objectives and Overview 17

2 General Description **19**

2.1 Markov Processes 19

2.1.1 Numerical Integration Using Euler's Method 22

2.2 Reliability Analysis Using Markov Models 23

2.2.1 State Space Reduction Techniques 25

2.3 Overview of Hierarchical Modeling Process 28

2.4 Determination of the Effective Component Failure Rates 31

3 Decomposition and Recombination **34**

3.1 Guidelines for Decomposition 35

3.2 Exploring Decomposition 36

3.3 Global vs. Channelized Dependence 38

3.3 Multiple Level Decomposition 43

3.5 Conclusions 46

4 Properties of Effective Component Failure Rates **47**

4.1 Calculating the Effective Failure Rate 47

4.2 Affect of Exact Subsystem Architecture on Effective Rates 52

4.2.1 No Shared Resources 52

4.2.2 Shared Resources 55

4.2.3 Decomposition with Common Mode Failure 58
4.3 Defining States for the Approximate Subsystem 62
4.3.1 Sequence Dependencies 62
4.3.2 States with Zero Probability 65
4.3.3 State Space Reduction Techniques 68
4.4 Conclusion 71

5 Numerical Issues 72
5.1 Effects of Discretization 72
5.2 Choosing an Appropriate Tavg 73
5.2.1 Sensitivity to Roundoff Error 77
5.2.2 Sensitivity to Integration Error 77
5.3 A Numerical Example 78
5.3.1 Roundoff Error 78
5.3.2 Integration Error 83
5.4 Improved Integration 89
5.5 Dependence of Tavg on Component Failure Rates 93
5.6 Conclusions 94

6 Hierarchical Analysis of Some Sample Systems 96
6.1 Two Sample Systems 96
6.1.1 No External Dependence 96
6.1.2 Global Dependence 100
6.2 An Automated Markov Model Construction Tool 102
6.3 Space Station Freedom 103
6.3.1 Traditional Analysis of SSF 104
6.3.2 Hierarchical Analysis of Baseline SSF Configuration 105
6.3.3 Improving Baseline SSF Architecture 109
6.3.4 Hierarchical Analysis of Improved SSF Configuration 110
6.4 Conclusions 114

7 Summary and Conclusions 115
7.1 Summary of Thesis 115
7.2 Limitations and Suggestions for Further Work 116

References 118

List of Figures

Figure 2.1 Discrete-Transition State Transition Diagram....20
Figure 2.2 Continuous-Transition State Transition Diagram....21
Figure 2.3 Two Component Parallel System....23
Figure 2.4 Parallel Configuration Markov model....24
Figure 2.5 A Sample 'Processor Core'....25
Figure 2.6 Markov Model for 'Processor Core'....26
Figure 2.7 Aggregated Markov Model for 'Processor Core'....27
Figure 2.8 Truncated Markov model....28
Figure 2.9 Overview of Hierarchical Modeling Process....29
Figure 2.10 Exact System....29
Figure 2.11 Candidate Decomposition....30
Figure 2.12 Exact Subsystem and Corresponding Approximate Subsystem....30
Figure 2.13 Approximate System....31
Figure 2.14 Markov Model for Approximate 'Processor Core'....32
Figure 3.1 Exact System....35
Figure 3.2 Exact System - Case I....36
Figure 3.3 Approximate System - Case I....37
Figure 3.4 Exact system - Case II....37
Figure 3.5a No Channelized Dependence....39
Figure 3.5b Channelized Dependence....39
Figure 3.6 Global Dependence....42
Figure 3.7 Overview of Multiple Level Hierarchical Modeling....44
Figure 3.8 Exact Subsystem 1....45
Figure 4.1 General 3 - Component Model....48
Figure 4.2 Sample System with Arbitrary Cross Strapping....50
Figure 4.3 Approximate Subsystem Markov Model....50
Figure 4.4 Markov Model for 'Processor Core' - No Shared Resources....53
Figure 4.5 Markov Model for 'Processor Core' - Shared Resources....55
Figure 4.6 Approximate Subsystem Model with Common Mode Failure....58
Figure 4.7 Example System....63
Figure 4.8 System with Unreachable State....65
Figure 4.9 Revised Approximate Subsystem Model....67
Figure 4.10 Approximate Subsystem Model with State Aggregation....69

Figure 4.11 Approximate Subsystem Model with Truncation....70
Figure 5.1 Exact Power Subsystem....74
Figure 5.2 Approximate Power Subsystem....74
Figure 5.3 Markov Model for Approximate Power Subsystem....75
Figure 5.4 Euler's Method Applied to State Equations....78
Figure 5.5 Nature of Total Error....84
Figure 5.6 Euler vs. Exact Value for $4\lambda^e$(0 Comp)....89
Figure 5.7 Trapezoidal Rule of Integration Applied to State Equations....90
Figure 5.8 Trapezoidal vs. Exact Value for $4\lambda^e$(0 Comp)....92
Figure 6.1 Exact System - No External Dependence....96
Figure 6.2 Approximate System - No External Dependence....97
Figure 6.3 Markov Model for Approximate Subsystem - No External Dependence....98
Figure 6.4 Absolute Error in Reliability/Unreliability - No External Dependence....98
Figure 6.5 Relative Error in Reliability - No External Dependence....99
Figure 6.6 Relative Error in Unreliability - No External Dependence....99
Figure 6.7 Exact System - Global Dependence....100
Figure 6.8 Approximate System - Global Dependence....100
Figure 6.9 Markov Model for Approximate Subsystem - Global Dependence....101
Figure 6.10 Absolute Error, System Unreliability - Global Dependence....102
Figure 6.11 MB - 2 Exact System Architecture....104
Figure 6.12 Approximate Power Subsystem Markov model....106
Figure 6.13 MB - 2 Approximate System Architecture....108
Figure 6.14 Approximate Power Subsystem Markov Model - Redesign....111
Figure 6.15 Revised Approximate Power Subsystem Markov Model....113

List of Tables

Table 4.1 Test Results - No Shared Resources.....54

Table 4.2 Test Results - Shared Resources.....57

Table 4.3 Test Results - Common Mode Failure.....61

Table 5.1 Evaluation Results for Exact Subsystem (taken at 1 hour time intervals).....79

Table 5.2 Effective Failure Rate (T_{avg} = 1.0 hour).....79

Table 5.3 Approximate Accuracy of λ^e (taken at 1 hour time intervals).....81

Table 5.4 Evaluation Results for Exact Subsystem (taken at 50 hour time intervals).....82

Table 5.5 Approximate Accuracy of λ^e (T_{avg} = 50.0 hour).....83

Table 5.6 λ^e Using Euler Approximation (T_{avg} = 50.0 hours).....85

Table 5.7 Approximate State Probabilities (T_{avg} = 50.0 hours).....85

Table 5.8 Total (absolute) Error in Approximate Subsystem Model Probabilities (T_{avg} = 50.0 hours).....86

Table 5.9 λ^e (T_{avg} = 100.0 hours).....87

Table 5.10 Total (absolute) Error in Approximate Subsystem Model Probabilities (T_{avg} = 100.0 hours).....87

Table 5.11 Comparison of Euler Failure Rate to Exact (T_{avg} = 100 hours).....88

Table 5.12 λ^e Using Trapezoidal Rule (T_{avg} = 50.0 hours).....91

Table 5.13 Total (absolute) Error Associated with Trapezoidal Rule (T_{avg} = 50.0 hours).....91

Table 5.14 Effect of Component Failure Rates on T_{avg}.....94

Table 6.1 Hierarchical vs. Traditional Analysis - Initial Architecture.....109

Table 6.2 Hierarchical vs. Traditional Analysis - Final Architecture.....114

Chapter 1

Introduction

1.1 Background

High reliability/availability systems are often designed using redundant architectures composed of moderately reliable components. While component reliabilities can be determined by individual testing, overall reliability cannot reasonably be determined in this way. Thus, for redundant systems, analytical methods must be used to determine overall reliability [Babcock1].

For redundant systems which achieve high reliability, Markov models are often used to determine overall system reliability [Babcock1]. The construction and validation of these Markov models is a time consuming and error prone process. To mitigate these problems, the construction and evaluation of such models has been automated by the Computer Aided Markov Evaluator [Hutchins].

Markov model state space grows exponentially with the number of components in a given system. A variety of techniques are already used to help control Markov model state space explosion, including model truncation and state aggregation [Babcock1, ch. 6]. Unfortunately, state aggregation provides only limited state space reduction. Model truncation provides more significant state space reduction at the expense of introducing an error bound on system reliability. For highly reliable, short mission time systems, truncation provides tight bounds with a manageable number of states. However, if the system has a large number of components, and a "long" mission time (relative to the component mean time to failure) then truncation provides either unacceptably wide reliability bounds or an intractably large number of states.

Since Markov model state space grows exponentially, evaluating the system in several segments would take dramatically less time than the evaluation of a full system composed of these segments. Since these segments are relatively small, an accurate reliability estimate could be obtained for each. This would produce an accurate estimate for the reliability of many large systems which could not reasonably be evaluated by a more traditional analysis. For these reasons, the reliability analyst would like to be able to decompose a large system into subsystems which could be evaluated separately and then recombined to obtain the reliability for the overall system.

Occasionally, redundant systems can be decomposed into completely independent subsystems. In this case, the reliability for the overall system is simply the product of the reliability for each one of the independent subsystems [Bazovsky]. Alternatively, the reliability for the overall system can sometimes be expressed in a reasonably simple closed form expression which involves the reliabilities and unreliabilities of redundant subsystems. These closed form expressions for system decomposition can often be derived through an application of the Law of Total Probability [Bazovsky, Ch. 13].

1.2 Decomposition Based on the Law of Total Probability

The law of total probability can be used to express the reliability for a system in terms of the conditional reliabilities and unreliabilities of individual components and/or subsystems. If A and B are defined as events in a probability space, then the law of total probability states

$$\text{Prob}(A) = \text{Prob}(A \mid B)\text{Prob}(B) + \text{Prob}(A \mid \text{Not } B)\text{Prob}(\text{Not } B)$$

If A is defined as the event that the overall system is operational, and B is defined as the event that a particular component, or subsystem, is operational, then the above law can be used to determine system reliability. Also, we can denote the probability that a system or component is operational by R, to indicate a reliability. Similarly, the probability that a system or component is not operational will by denoted by Q, to indicate unreliability. Thus, we can rewrite the law of total probability in a form which is more indicative of its application within the context of reliability analysis.

$$R(A) = R(A \mid B)R(B) + R(A \mid \overline{B})Q(B)$$

Repeated application of the law of total probability to the above expression may eventually lead to a sum of terms which can be evaluated numerically. Occasionally, an entire system may be decomposed into an expression containing solely component reliabilities. However, a more useful application of this approach would allow the reliability analyst to break the system down into an expression composed of the reliability of various <u>independent</u> subsystems. In this case, the law of total probability need only be applied a small number of times to yield an expression which can be evaluated fairly easily. Here the overall reliability is simply the product of the reliability of each of the independent subsystems. This approach has successfully been used in the evaluation of an integrated aircraft flight control system [Motyka].

Application of the law of total probability sometimes produces terms which are difficult to evaluate. For example, Prob(A | B) may not be easily evaluated if the events A and B have non-trivial dependencies. The derivation of these dependencies may be as intractable as the original model. Consequently, for systems which cannot easily be broken down into independent subsystems, this method becomes anywhere from tedious to intractable. For such systems a different approach to system decomposition must be taken.

1.3 An Alternative Approach to Hierarchical Modeling

An alternative approach to the type of decomposition presented above is to decompose the system into a set of subsystems with fewer restrictions on the dependencies that can exist between subsystems. To introduce this concept consider a system for which a single subsystem is extracted for separate evaluation. Results of the evaluation of the subsystem will then be merged back into the remainder of the system to obtain results for the entire system.

The system we start with will be called the *exact system.* The subsystem that is extracted will be referred to as the *exact subsystem.* The exact subsystem is replaced with an *approximate subsystem* composed of a small number of components. This set of components is selected so as to capture all of the direct interactions between the exact subsystem and the remainder of the exact system. This set of components will be substituted back into the remainder of the exact system. The result is an *approximate system* that mimics the behavior of the exact system but has fewer components. Hence, the Markov model used to estimate the reliability for the approximate system, and consequently the corresponding exact system, has a smaller, more tractable state space.

In order to capture this interaction, the components used in the approximate subsystem should behave like the exact subsystem. In particular, the reliability of the approximate subsystem should be the same as the reliability of the exact subsystem. This simple set of requirements forms the basis for a hierarchical modeling technique that is more broad in application than an approach based entirely on the law of total probability.

1.4 Objectives and Overview

The purpose of this thesis is to develop a method for hierarchical system modeling which can be used in conjunction with current Markov modeling techniques. General guidelines for decomposing the exact system into subsystems which can be evaluated separately will be presented. One method for evaluating the resulting subsystems will be described. Also, the feasibility of this technique will be demonstrated by the evaluation of some small examples. Finally, the usefulness of this technique will be demonstrated on a real world system.

The next chapter begins with a description of the application of Markov models to reliability analysis. Chapter 2 also contains an overview of the hierarchical modeling process. Decomposition of the exact system is presented. One method of analysis for the resulting subsystems is described. Finally, the recombination and evaluation of the resulting approximate system is discussed. An example is presented to illustrate this process.

System decomposition and recombination are explored in more detail in chapter 3. Several examples are used to highlight the benefits and problems associated with this technique. In particular, rules are presented to help the analyst determine whether or not a given decomposition is valid for this sort of modeling. Guidelines are also given to help the analyst choose an appropriate and beneficial decomposition.

The example presented in chapter 2 is used again in chapter 4 to discuss the numerical evaluation of the exact subsystem and the development of parameters for the approximate subsystem in much greater detail. This chapter explains what information is needed from the evaluation of the exact subsystem. This clarifies the distinction between this sort of hierarchical modeling and the decomposition of systems with completely independent subsystems.

Chapter 5 analyzes some of the numerical issues which arise from this hierarchical modeling process. The choice of certain analysis parameters can produce large errors and the potential for numerical instability. These problems are demonstrated with a sample system, and general guidelines are presented to help the analyst choose appropriate parameter values.

The entire process is demonstrated in chapter 6 on an actual system: the Space Station Freedom. In chapter 7, the major results of this thesis are summarized and the limitations of this research are discussed.

Chapter 2

General Description

2.1 Markov Processes

Some random processes can be modeled by a series of dependent trials or events. Such systems are said to contain memory. Some processes contain memory of a very simple form. In a Markov process, successive trials depend only on the previous trial. A more precise statement of the Markov condition is that conditioned on the present state of the system, the future is independent of the past [Drake]. Notice however that the Markov condition is highly dependent on the definition of the system states. In fact, such a process is completely characterized by the definition of its states and the transition probabilities between states.

Consider the case of a broken traffic light which changes to a random color at deterministic points in time. Define the state of the traffic light as its color. Let the random variables S_n denote the state, or color of the traffic light at time n. Assume that this process satisfies the Markov condition, that $Pr(S_{n+1}|S_n,S_{n-1},...,S_0) = Pr(S_{n+1}|S_n)$. Also assume that the probability of a correct transition (e.g., red to green) is 0.8, and that the probability of an incorrect transition (e.g., red to yellow) is 0.1. A graphical representation of this Markov process can be found in the state transition diagram, or Markov model of figure 2.1. The state transition probabilities are represented by the numerical values discussed above.

The probability distribution as a function of time for the Markov model of figure 2.1 is defined by the system of equations 2.1, and the initial probability distribution. Here $P_i[n]$ is the probability of being in state i, (i = 0, 1, 2) at time n. Equation 2.1a defines the flow of probability into and out of state 0 (red). Equations 2.1b and 2.1c are associated in a similar fashion with states 1 (green) and 2 (yellow) respectively. Equation 2.1d simply indicates that the states 0, 1, and 2 define a complete probability sample space. Finally, notice that this system of equations defines an iterative formula for the state probabilities of the Markov model.

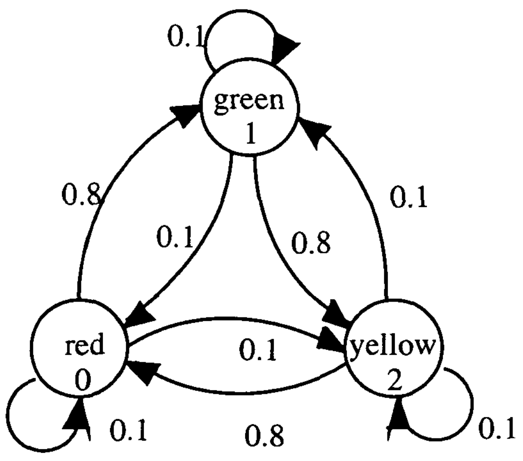


Figure 2.1: Discrete-Transition State Transition Diagram

$$P_0[n+1] = 0.1P_0[n] + 0.1P_1[n] + 0.8P_2[n] \tag{2.1a}$$

$$P_1[n+1] = 0.8P_0[n] + 0.1P_1[n] + 0.1P_2[n] \tag{2.1b}$$

$$P_2[n+1] = 0.1P_0[n] + 0.8P_1[n] + 0.1P_2[n] \tag{2.1c}$$

$$P_0[n] + P_1[n] + P_2[n] = 1.0 \tag{2.1d}$$

The broken traffic light example presented above represents a discrete-state discrete-transition Markov process. One distinguishing characteristic of such a process is that state transitions occur at deterministic points in time.

If the transition time for a Markov process is a continuous random variable then the resulting system is referred to as a discrete-state continuous-transition Markov process. This type of Markov process is characterized by transition 'rates' $p_{ij}(t)$ which define the flow of probability from state i to state j. These rates represent the conditional probability of making a transition to state j given the system is in state i in a differential time from t to $t + dt$.

If the broken traffic light discussed above makes transitions between states at random times, then this process can be described graphically by the continuous-time transition diagram of figure 2.2. In this diagram, transition rates replace transition probabilities. Notice the absence of self transitions in figure 2.2. If a transition is not taken in differential time then the process remains in the same state. Also, the discrete time state equations are now replaced by the system of differential equations 2.2.

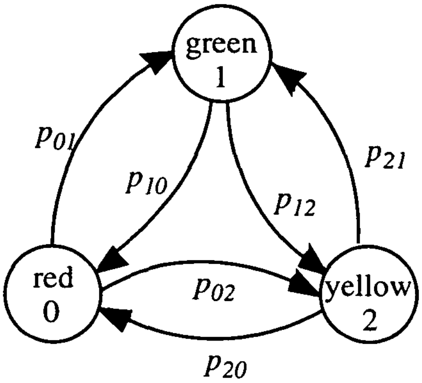


Figure 2.2: Continuous-Transition State Transition Diagram

$$\frac{dP_0}{dt} = -(p_{01} + p_{02})P_0 + p_{10}P_1 + p_{20}P_2 \tag{2.2a}$$

$$\frac{dP_1}{dt} = -(p_{10} + p_{12})P_1 + p_{01}P_0 + p_{21}P_2 \tag{2.2b}$$

$$\frac{dP_2}{dt} = -(p_{20} + p_{21})P_2 + p_{02}P_0 + p_{12}P_1 \tag{2.2c}$$

This system of equations can also be presented in the matrix form of equation.

$$\frac{d}{dt}\begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \end{bmatrix} = \begin{bmatrix} -(p_{01}+p_{02}) & p_{10} & p_{20} \\ p_{01} & -(p_{10}+p_{12}) & P_{21} \\ p_{02} & p_{12} & -(p_{20}+p_{21}) \end{bmatrix} \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \end{bmatrix} \tag{2.3a}$$

$$[A] = \begin{bmatrix} -(p_{01}+p_{02}) & p_{10} & p_{20} \\ p_{01} & -(p_{10}+p_{12}) & P_{21} \\ p_{02} & p_{12} & -(p_{20}+p_{21}) \end{bmatrix} \tag{2.3b}$$

$$\vec{P}(t) = \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \end{bmatrix} \tag{2.3c}$$

$$\frac{d\vec{P}(t)}{dt} = [A]\vec{P}(t) \tag{2.3d}$$

If the transition rates are time-invariant, then the process is considered homogeneous. In this case, the solution to equation 2.2 is the time-invariant matrix exponential, commonly referred to as the matrix exponential [McCarragher]. A variety of methods exists for calculating the matrix exponential [Moler]. In practice, the matrix exponential is often calculated by numerical integration of the system of equations. We will assume for the next section that all transition rates are time-invariant.

2.1.1 Numerical Integration Using Euler's Method

One particularly simple numerical integration technique is Euler's method. This technique involves replacing derivatives with respect to time with differences divided by a small change in time (i.e., d/dt ↦ Δ/Δt). The matrix equation of 2.3 was modified according to Euler's algorithm by replacing $\frac{d\vec{P}(t)}{dt}$ with $(\vec{P}[(n+1)\Delta t]-\vec{P}[\Delta t])\frac{1}{\Delta t}$. The result is

$$\vec{P}[(n+1)\Delta t] = \vec{P}[n\Delta t] + \Delta t[A]\vec{P}[n\Delta t] \qquad (2.4a)$$

The matrix and vector quantities, [M] and $\vec{P}[n]$ can be defined according to equations 2.4b and 2.4c, where I_3 is the 3x3 identity matrix.

$$[M] = I_3 + \Delta t[A] \qquad (2.4b)$$

$$\vec{P}[n] = \vec{P}[n\Delta t] \qquad (2.4c)$$

$$\vec{P}[n+1] = [M]\vec{P}[n] \qquad (2.4d)$$

These quantities can be used to produce the iterative formula, 2.4d for the state probabilities P_o, P_1, and P_2. Equation 2.4d can be 'stepped' forward in time to estimate the state probabilities at several points in time.

To estimate $\vec{P}[n]$, given $\vec{P}[0]$ would require n matrix-vector multiplications. For a Markov model with m states, each matrix-vector multiplication involves m^2 scalar multiplications. In order to estimate the state probability vector at time n by this 'matrix stepping' routine would require nm^2 scalar multiplications.

Notice from equation 2.4c that $\vec{P}[1] = [M]\vec{P}[0]$ and $\vec{P}[2] = [M]\vec{P}[1]$. Applying equation 2.4d gives $\vec{P}[2] = [M]^2\vec{P}[0]$, and in general $\vec{P}[n] = [M]^n\vec{P}[0]$. However, multiplying two matrices involves m^3 scalar multiplications. Clearly, multiplying the matrix [M] n times requires much more computation than n matrix-vector multiplications.

Notice that if n is even we can express $\vec{P}[n] = [M]^{n/2}\vec{P}[n/2] = [M]^{n/2}[M]^{n/2}\vec{P}[0]$. Here, the matrix [M] is multiplied n/2 times and then squared. This only requires $(1 + n/2)m^3$ scalar multiplications. In general if n is an integer power of 2, (i.e., $\log_2 n$ is an integer)

$[M]^n$ can be determined through repeated squaring (i.e., $[M][M]=[M]^2 \rightarrow [M]^2[M]^2=[M]^4 \ldots$). This 'matrix doubling' routine requires only $(\log_2 n)m^3$ multiplications. Thus for large n, and relatively small m, matrix doubling may be less computationally demanding than matrix stepping [Moler, p. 809].

The next section will illustrate how Markov models are used in reliability analysis. However, it is important to bear in mind that when actual reliability estimates are sought, this problem reduces to the numerical integration of a system of differential equations. Although many different numerical integration methods exist, Euler's method provides us with a very simple approach to solving this problem.

2.2 Reliability Analysis Using Markov Models

The two component system of figure 2.3 represents an active parallel configuration. In order for the system to operate, either component A or component B must be operational. In a 'standby' configuration, the first component starts in an operating state and the second component remains idle until the first component fails. However, in this example the system will be called 'active' because both components are in operation until a component failure occurs.

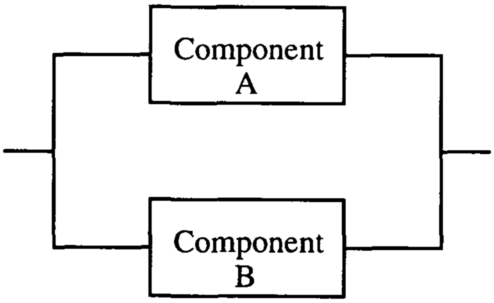


Figure 2.3: Two Component Parallel System

The operation of the parallel system in figure 2.3 can be characterized by the Markov reliability model of figure 2.4. Notice that each state in the Markov model represents a distinct operational or failed state of the system. Also, transitions between these states result from either a component failure or repair. Components A and B fail at rates λ_A and λ_B respectively. These components are repaired at rates μ_A and μ_B. Notice that the Markov model captures different failure sequences in different states (i.e., A then B vs. B then A failed). Also, since the reliability of the system is the probability of continuous operation, no repair transitions are permitted out of the two states which correspond to system failure. These states may also be referred to as system loss states.

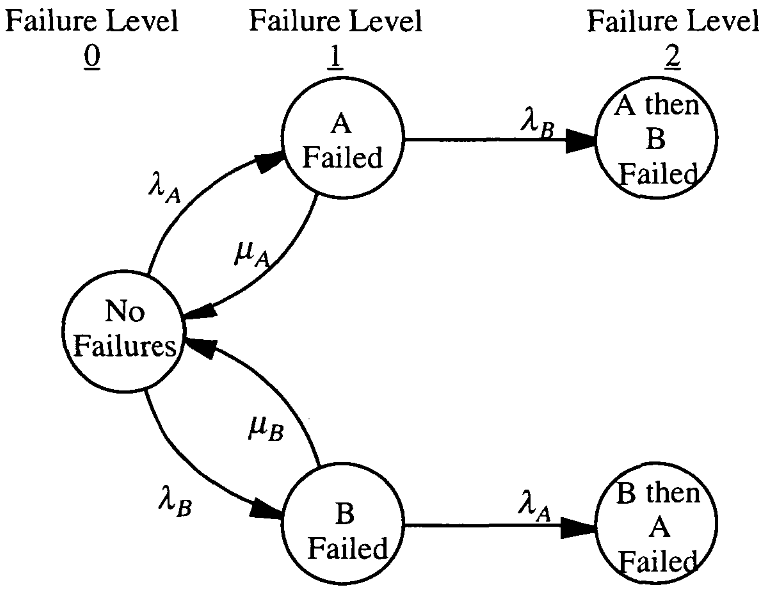


Figure 2.4: Parallel Configuration Markov model

The Markov model of figure 2.4 can be evaluated to obtain the probability in each of its states as a function of time. The states of this Markov model are broken up into *failure levels*, (i.e., 0, 1, 2). Each failure level contains all states with the same number of failed components (e.g., failure level 1 contains states with either A failed or B failed). The probability of being in all of the operational states of figure 2.4 can be summed to obtain the reliability for the system.

If the systems analyst is only interested in the overall system reliability, then the two system loss states of figure 2.4 may be combined into one state without penalty. Such a state aggregation results in a Markov model whose state space is an exhaustive list of all combinations of failed and unfailed components in the system with no distinctions made for order of failure. The number of states in this model is necessarily 2^n, where n denotes the number of components in the system (i.e., the number of combinations of n binary objects). Since sequence dependencies only increase the state space of the Markov model, this aggregated model provides a lower bound on state space growth as a function of the number of components in a system. We conclude that without any model reduction techniques, Markov model state space grows at least exponentially with the number of components, n.

2.2.1 State Space Reduction Techniques

Two common state space reduction techniques are *exact state aggregation* and *model truncation*. [Babcock1, ch. 6]. In order to describe these techniques, the following terms are introduced: We say that a component is *unfailed* when the component itself is operational. We say that a component is *functional* when both the component itself is operational, and all of its supporting equipment is operational. Finally, in order to simplify this problem we assume in the following and in the remainder of this thesis that all systems are non-repairable (i.e., $\mu = 0$).

State Aggregation

The system in figure 2.5 requires at least one of the two processors to be functional. In order for processor 1 to be functional, either processor 1 and memory 1 must be unfailed, or processor 1 must be unfailed and processor 2 must be functional (i.e., processor 1 can borrow processor 2's memory, via processor 2). Similarly for processor 2.

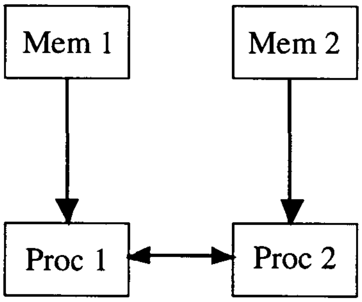


Figure 2.5: A Sample 'Processor Core'

The full, unaggregated Markov model for the system of figure 2.5 is shown in figure 2.6. The states in this model are labeled by a list of their failure level and order within that failure level (e.g., the first state is labeled 0,1). Notice that λ_M is the failure rate for a memory unit, and λ_P is the failure rate for a processor unit. Notice that there are no exit transitions from system loss states to any subsequent failure levels since it is assumed that the system has stopped operating in these states. States grouped by dashed lines have the same functional "condition", such that the outcome of all subsequent failures do not depend on the distinctions among these states. Thus, these states can be aggregated exactly without any loss of model accuracy.

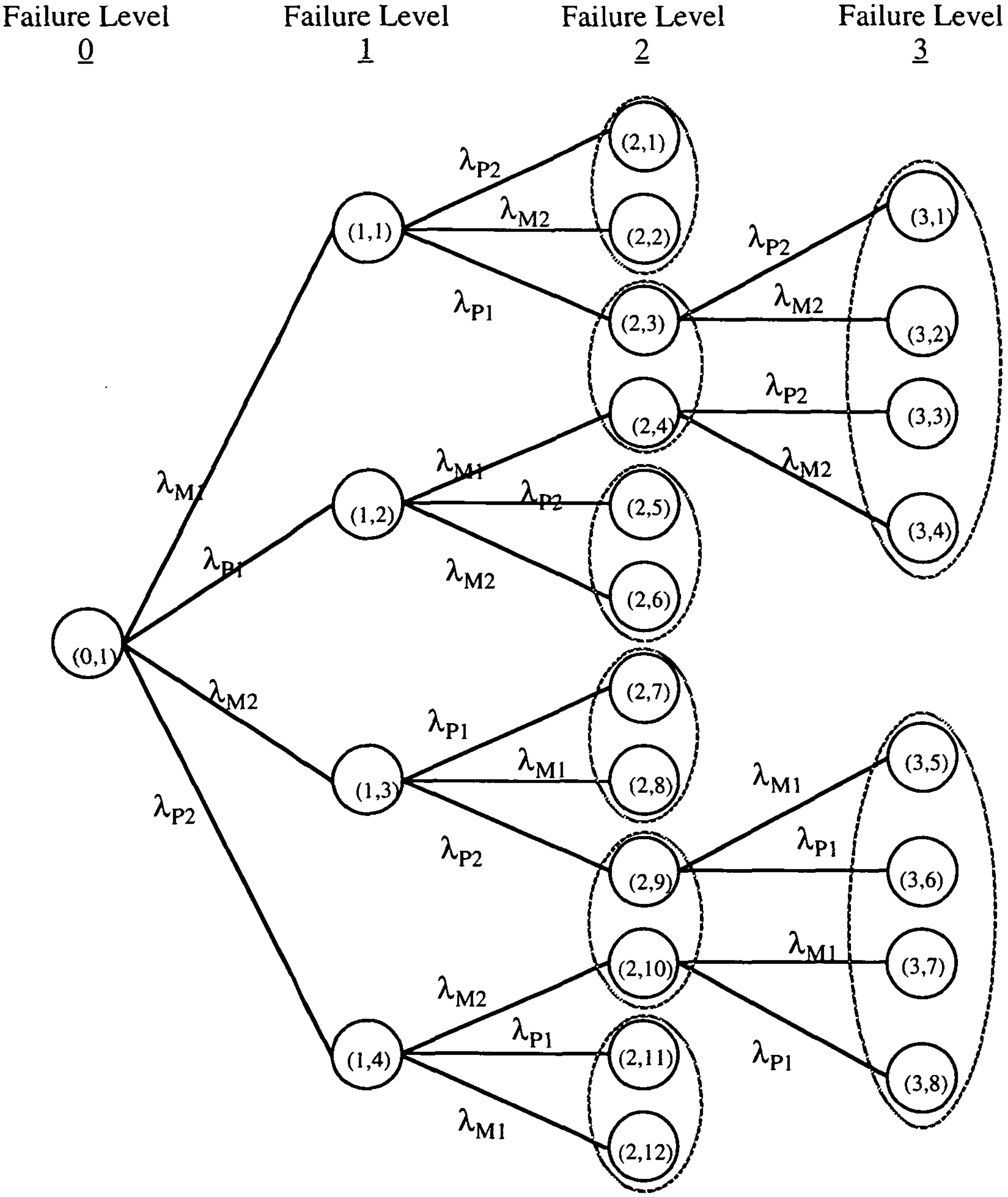


Figure 2.6: Markov Model for 'Processor Core'

If all of the indicated states are aggregated, the Markov model of figure 2.7 results. The reliability estimate obtained from this model is exactly the same as the estimate obtained from the unaggregated model (i.e., there is no approximation involved in this type of state aggregation).

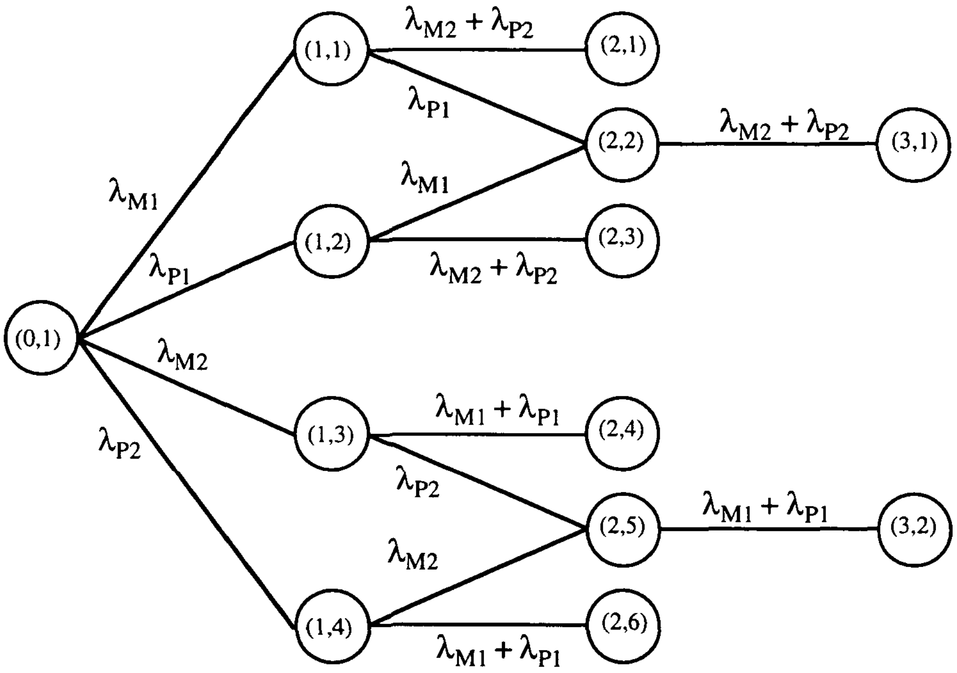


Figure 2.7: Aggregated Markov Model for 'Processor Core'

Model Truncation

Model truncation involves aggregating all states beyond a certain failure level into one state. Truncation introduces an approximation into the calculation of the system reliability. Since the exact operational condition of the system is not known in this "truncation" state, bounds are obtained by assuming it is either all operational or all failed. Thus the bounds for system reliability are obtained by summing all operational states (without the truncation state) and also by summing all operational states including the truncation state. The true reliability lies between these two bounds. A truncated version of the processor core model is shown in figure 2.8.

For the model of figure 2.8, the system reliability is bounded by the following inequalities:

$$R(Sys) \geq P(0,1) + P(1,1) + P(1,2) + P(1,3) + P(1,4)$$

$$R(Sys) \leq P(0,1) + P(1,1) + P(1,2) + P(1,3) + P(1,4) + P(Trunc\ State)$$

As long as the probability in the truncation state is small, the truncated model provides a good approximation to the untruncated models in figures 2.7 and 2.6. Finally notice that even on this simple example the resulting model state space has been reduced substantially from figure 2.6.

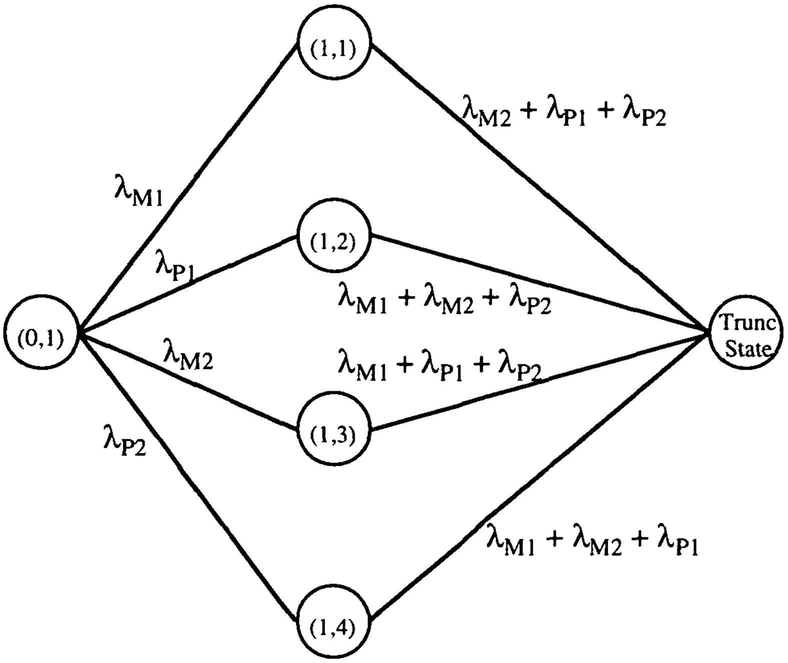

(1,1)
$\lambda_{M2} + \lambda_{P1} + \lambda_{P2}$
λ_{M1}
(1,2)
λ_{P1}
$\lambda_{M1} + \lambda_{M2} + \lambda_{P2}$
(0,1)
Trunc State
λ_{M2}
$\lambda_{M1} + \lambda_{P1} + \lambda_{P2}$
(1,3)
λ_{P2}
$\lambda_{M1} + \lambda_{M2} + \lambda_{P1}$
(1,4)

Figure 2.8: Truncated Markov model

2.3 Overview of Hierarchical Modeling Process

It has been shown that Markov model state space grows exponentially with the number of components in the system. Breaking the system into smaller segments and evaluating these segments takes dramatically less time than the evaluation of a complete system. For example, a sixteen component system has 2^{16} states under simple assumptions. Dividing it into four subsystems of four components gives $4 \cdot 2^4$ states. This enables the system's analyst to produce substantially more accurate Markov models by relying less heavily on model truncation. Consequently, there is a clear advantage to decomposing one large system into subsystems which can be evaluated separately then recombined to obtain the reliability for the overall system.

The hierarchical approach proposed here relies on a very special dependence between different segments of a system. The overall approach is outlined in figure 2.9.

Often, dependence on a particular subsystem within some larger system is restricted to a small set of operational conditions. This set of operational conditions can be represented by some small set of *effective components*. For example, consider the system represented in figure 2.10.

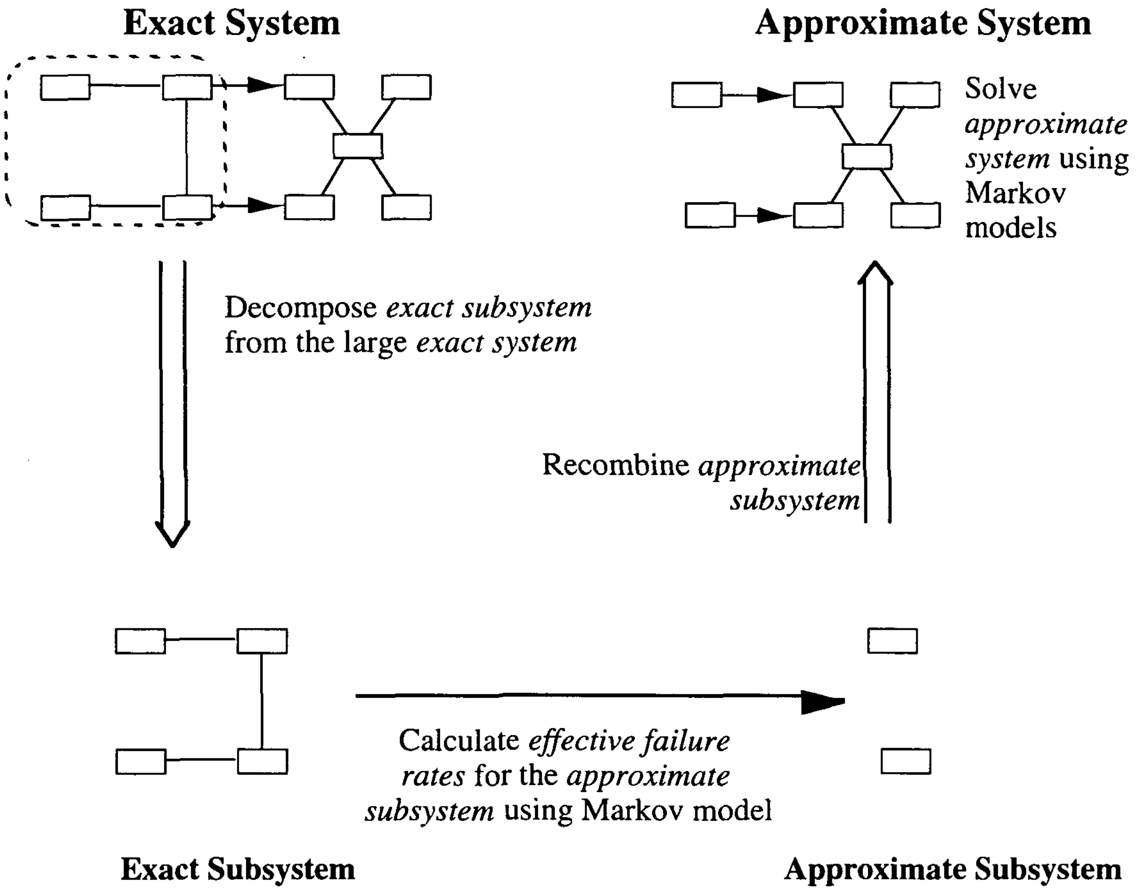

Exact System
Approximate System
Solve *approximate system* using Markov models
Decompose *exact subsystem* from the large *exact system*
Recombine *approximate subsystem*
Calculate *effective failure rates* for the *approximate subsystem* using Markov model
Exact Subsystem
Approximate Subsystem

Figure 2.9: Overview of Hierarchical Modeling Process

In order for decoder 1 to be functional it must receive input from processor 1. In order for decoder 2 to be functional it must receive input from processor 2. Both decoders can drive either actuator through the cross-link. This system uses the same processor core shown in figure 2.5. In order for the system to be considered operational at least one actuator must be functional.

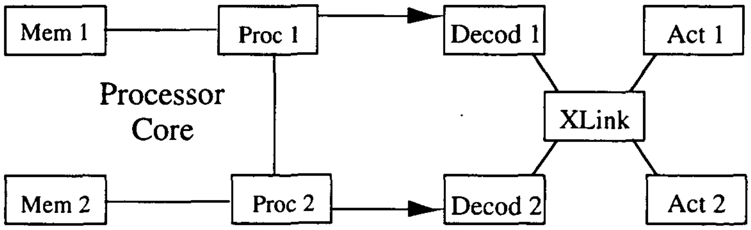


Figure 2.10: Exact System

From the description of the system in figure 2.10 we surmise that the effect of the processor core on the decoders, actuators and cross-link is completely captured by the operational state of processors 1 and 2. Consequently, the decoders, cross-link and actuators in figure 2.10 exhibit what we will call a *simple dependence* on the processor core. Notice that arrows pointed away from the processor core are used to emphasize the direction of this dependence.

Selecting a candidate decomposition often requires looking for 'focal points' for system behavior. For example, the utility of a specific decoder depends on its associated processor being operational. Determining processor operability requires examination of the memory units. Thus, the concept of processor 'channels' that mimic the behavior of each processor and its memory dependencies may provide the basis for simplification to be captured in an approximate subsystem. These channels represent a simple dependence on a particular subsystem. This sort of dependence is not always obvious, but once it has been identified it can be used to produce a candidate decomposition. For example one candidate decomposition of the system in figure 2.10 would be to isolate the processor core as indicated in figure 2.11.

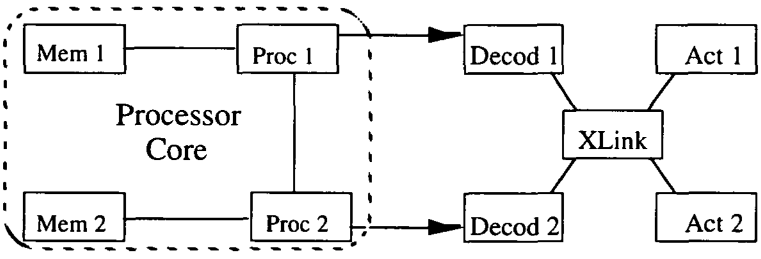


Figure 2.11: Candidate Decomposition

Once a subsystem has been identified for decoupling from the exact system, the next step is to produce a set of effective components which mimic the functionality of the subsystem, as well as its reliability. For example, in order to determine the reliability of the system in figure 2.11 we could replace the processor core with two effective processors, as shown in figure 2.12.

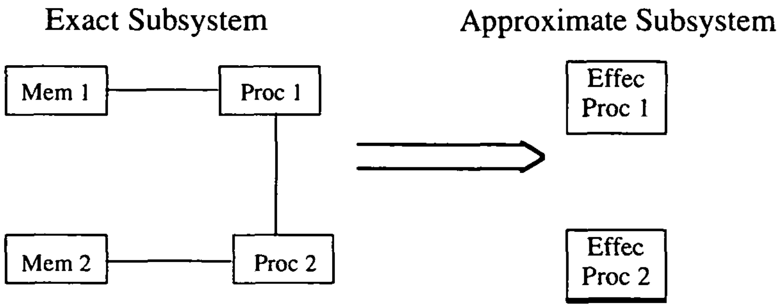


Figure 2.12: Exact Subsystem and Corresponding Approximate Subsystem

The approximate system which results from the decomposition of figures 2.11 and 2.12 is shown in figure 2.13. This approximate system can be used to estimate the reliability of the exact system shown in figure 2.11. In this approximate system, effective processor 1 has the same probability of being functional as the corresponding processor channel in the exact subsystem.

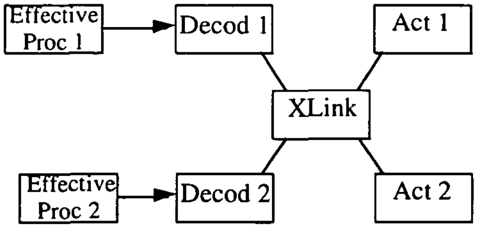


Figure 2.13: Approximate System

In order to evaluate the system shown in figure 2.13, we must be able to summarize the reliability of both effective processors in terms of their failure rates. In order to determine the *effective component failure rates* we must derive a system of equations which incorporate these failure rates as unknowns. One way to develop such a system of equations is to examine the Markov model for the approximate subsystem.

2.4 Determination of the Effective Component Failure Rates

A discrete-state continuous-transition Markov model can be produced which describes the reliability of the *approximate subsystem.* This Markov model is described by a system of state equations. The resulting system of equations relates the state probabilities at different points in time, to the effective failure rates in the approximate subsystem. If the state probability time histories are known, this system of equations can be solved for the set of effective failure rates.

Each of the states in the Markov model for the approximate subsystem maps to some set of states within the Markov model for the exact subsystem. By evaluating the Markov model for the exact subsystem, numerical values can be obtained for the state probabilities in the Markov model for the approximate subsystem. These values can be used in the system of state equations for the approximate subsystem model to solve for the effective failure rates. Once these rates have been determined, they can be used in a Markov model for the approximate system to determine system reliability.

Consider the system of figure 2.13. For this system we would like to determine failure rates for effective processors 1 and 2. In order to do this the Markov model of figure 2.14 is used to describe the approximate processor core. The transitions in this Markov model are labeled with effective failure rates, denoted by a superscript *e*. Subscripts are used to denote the effective component. In general these rates will be time-varying. These rates will also vary as a function of the failure condition of the subsystem, i.e., the states. In figure 2.14 dependence on state is captured within the parentheses. For example, $\lambda_1^e(2)$ is the effective failure rate of processor 1 when processor 2 has failed. Thus, state dependence in the effective failure rate is captured by using a different, time-varying, *effective transition rate* at each state.

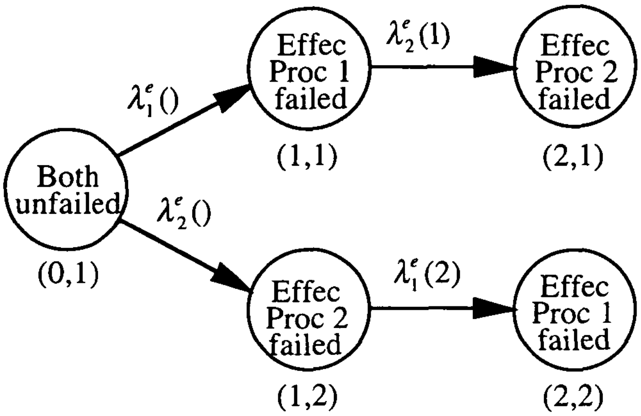


Figure 2.14: Markov Model for Approximate 'Processor Core'

The Markov model in figure 2.14 can be described by equations 2.5.

$$\frac{dP_{(0,1)}}{dt} = -[\lambda_1^e() + \lambda_2^e()]P_{(0,1)} \tag{2.5a}$$

$$\frac{dP_{(1,1)}}{dt} = \lambda_1^e()P_{(0,1)} - \lambda_2^e(1)P_{(1,1)} \tag{2.5b}$$

$$\frac{dP_{(1,2)}}{dt} = \lambda_2^e()P_{(0,1)} - \lambda_1^e(2)P_{(1,2)} \tag{2.5c}$$

$$\frac{dP_{(2,1)}}{dt} = \lambda_2^e(1)P_{(1,1)} \tag{2.5d}$$

$$\frac{dP_{(2,2)}}{dt} = \lambda_1^e(2)P_{(1,2)} \tag{2.5e}$$

This system of equations can be solved for $\lambda_1^e()$, $\lambda_2^e()$, $\lambda_1^e(2)$, and $\lambda_2^e(1)$. In order to solve such a system of equations we need to determine the appropriate state probabilities for the Markov model in figure 2.14. These state probabilities can be determined by evaluating a Markov model for the exact subsystem. If the Markov model of figure 2.7 is used, then the probability of being in states (0,1), (1,1) and (1,3) sum to produce the probability of being in state (0,1) of the approximate subsystem model. Similar mappings can be produced for all of the remaining states of the Markov model in figure 2.14.

Evaluating a model that includes time-varying, state dependent transition rates proceeds in a fashion similar to that of a more traditional time-invariant, state-independent transition rate model. The unique aspect is that the time-varying character is captured as piece-wise constant and the appropriate transition matrix is called at each time step of the numerical integration.

For now, we postpone discussion of how equations 2.5 are solved for the effective failure rates. We also postpone discussion of other reliability parameters such as fault coverage and repair rates until chapter 7. Until then we treat all systems as non-repairable and we assume that all components have perfect coverage.

Since a Markov model state space depends on the number of components in the system being evaluated, evaluation of the approximate system produced by this technique (figure 2.13) produces a smaller Markov model than that of the exact system (figure 2.11), which is easier to evaluate. More importantly, this means we do not have to rely as heavily on model truncation and its introduction of bounded solutions, to control the state space. This allows us to produce meaningful reliability estimates for a broader class of systems. However, the full benefits of this hierarchical modeling technique depend heavily on many of the details involved in determining the effective component failure rates; a subject treated more fully in chapters 4 and 5.

Chapter 3

Decomposition and Recombination

We have seen that by replacing a set of components from a large system with some smaller set of components, we can reduce the amount of work necessary to evaluate the reliability of a given system. Unfortunately, a decomposition which leads to a tractable solution is not always obvious. Clearly, if we can identify independent subsystems, these can be replaced by single components. But when can we replace a subsystem with some set of effective components? In the example used in chapter 2, the exact system depended on the functional status of different processor 'channels'. We use this notion of channelized dependence in developing guidelines for system decomposition.

We have concentrated on using Markov models to evaluate system reliability. However, it may be easier to understand what this decomposition accomplishes in terms of conditioning events which can be used to compute system reliability. For example, consider the system of figure 2.10, reproduced here in figure 3.1. The reliability for this system can be expressed in the form of equation 3.1.

$$\begin{aligned} R(Sys) = {} & R(Sys|Channel1Channel2)\Pr(Channel1Channel2) \\ & + R(Sys|\overline{Channel1}Channel2)\Pr(\overline{Channel1}Channel2) \\ & + R(Sys|Channel1\overline{Channel2})\Pr(Channel1\overline{Channel2}) \\ & + R(Sys|\overline{Channel1}\overline{Channel2})\Pr(\overline{Channel1}\overline{Channel2}) \end{aligned} \tag{3.1}$$

This equation can be obtained by application of the law of total probability. In this set of equations channels 1 and 2 refer to processor channels 1 and 2, where an overbar indicates a failed channel and no overbar indicates an unfailed channel. The joint probability that channel 1 has not failed and channel 2 has not failed is denoted by the term $\Pr(Channel1Channel2)$. The probabilities of the remaining three combinations of failed and unfailed processor channels are denoted similarly. The notation $R(Sys|\cdot)$ denotes the system reliability conditioned on a given event. In this case the conditioning events are all of the failed and unfailed combinations of the various processor channels.

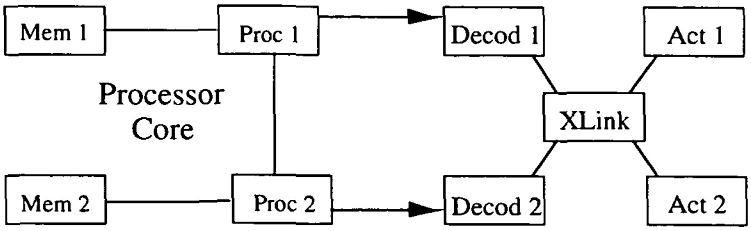


Figure 3.1: Exact System

Failure rates for the effective components of the approximate subsystem (figure 2.12) are derived to match the state probabilities of the Markov model in figure 2.14. These states correspond to the requisite conditioning events in equation 3.1. For example, the probability of being in state (0,1) equals the probability of having both channels 1 and 2 available, i.e., $P_{(0,1)}$=Pr(*Channel1Channel2*). Also, the Markov model for the approximate system of figure 2.13 combines the probabilities of these conditioning events to obtain the system reliability. Notice, that equation 3.1 is an alternative, but equivalent method for combining these joint probabilities (assuming that the requisite conditional reliabilities can be calculated).

Although equation 3.1 was derived for a particular system, the same equation holds for any system which contains two subsystem channels. In fact, a similar expression can be derived for any system which exhibits a channelized dependence on some subsystem. Therefore, a sufficient condition for the validity of a given decomposition to hold is that the Markov model for the approximate subsystem must accurately reflect all the requisite combinations of failed and unfailed subsystem channels. For example, if the probability of being in state (1,1) of the Markov model in figure 2.13 does not equal the probability of channel 1 being failed and channel 2 being unfailed, then the approximate system of figure 2.12 will not produce the reliability for the corresponding exact system. This concept of decomposition is captured in the guidelines of the following section.

3.1 Guidelines for Decomposition

In order to determine an acceptable decomposition for a given system, the analyst must rely on a detailed knowledge of the system in question. The first guideline outlines a basic approach for finding a subsystem which can be decomposed. Once a candidate decomposition has been selected, guidelines two and three help determine the validity of this decomposition. The following sections explain these guidelines in greater detail, and provide supporting arguments.

1. In selecting a candidate decomposition, the analyst should look for a subsystem which contains a substantial amount of complexity, but which has a very simple interface with the remainder of the exact system.

2. Only information summarized by the effective components will be available in the approximate system. No detailed information of the status of the subsystem beyond these effective components is available to the approximate system.

3. Components outside of the subsystem may depend on the effective components in any way; however, effective components may only have a "global" dependence on components outside of the subsystem.

3.2 Exploring Decomposition

Consider the system of figure 3.2. The processors of this system need access to at least one memory unit in order to be functional. The memory units do not rely on any supporting hardware. The sensors require input from at least one processor. Actuators 1 and 2 are driven by processors 1 and 2 respectively. In order for the system to function, at least one sensor/actuator pair must be functional.

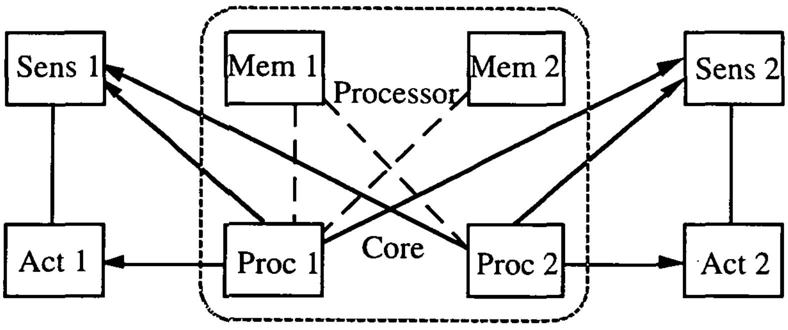


Figure 3.2: Exact System - Case I

Notice that the sensors interact with both processors. Thus, a single effective component, representing the whole processor core would be sufficient to characterize the interaction between the processor core and the sensors. However, the functionality of actuators 1 and 2 depend on the availability of specific processor channels, not the availability of the processor core as a whole. Replacing the processor core with a single effective component would not give sufficient information to characterize the interaction between the processor core and actuators; the functionality of the processor core as a whole does not indicate explicitly the functionality of either actuator. By replacing the processor core with two effective components, each representing a processor channel, we do have sufficient information to characterize all interaction between the processor core and the remainder of the overall system. Such an approximate system is shown in figure 3.3.

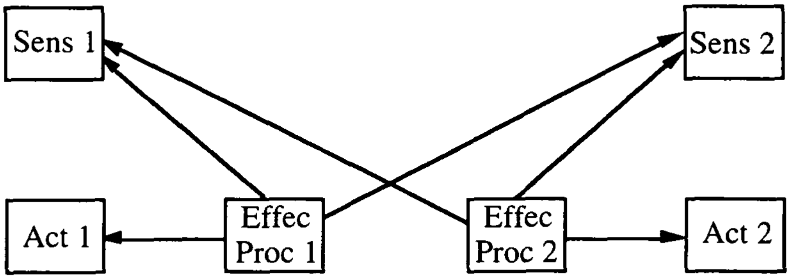


Figure 3.3: Approximate System - Case I

A crucial point in determining the number of effective components in the above example is an understanding of the level of detail necessary in order to fully capture all interaction between some subset of components and the remainder of a given system. In the example presented in figure 3.2 information between the processor core and the remainder of the system is completely characterized by knowing which processor channel is operational. This makes the decomposition in figure 3.3 possible.

Consider now a system where the cross strapping is such that the sensors and actuators need detailed information about the status internal to each processor channel. For this case the above reduction would be insufficient. Figure 3.4 shows such a system.

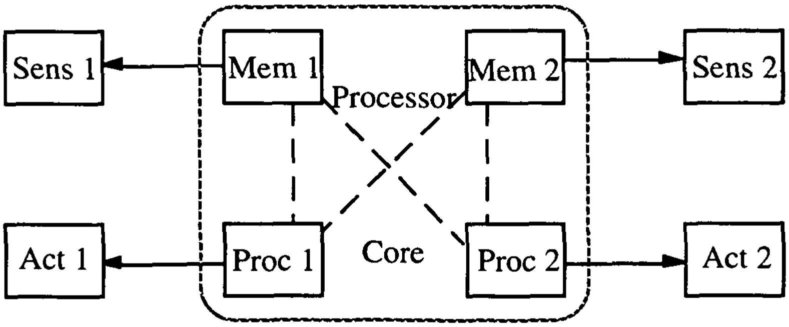


Figure 3.4: Exact system - Case II

In this configuration, in addition to the actuators depending on processor functionality, information about the status of the memory is necessary in determining the functional status of the sensors. Here, the channelized decomposition of figure 3.3 is insufficient in characterizing the interaction between the processor core and the remainder of the exact system. In fact, as long as the memory units are shared by both processors it is not clear that the processor core can be replaced by a smaller set of components at all.

We have already explored system decomposition in terms of the component space of the system, but now consider a closed form expression for system reliability. Notice that the reliability for both the system of figure 3.2 and the system of figure 3.4 can be expressed by equation 3.1. However, for the system of figure 3.4, this decomposition does not lead to a tractable model, i.e., the conditional reliabilities of this equation are not easily calculated since they do not correspond to a simple component description of the system.

We conclude that for the proposed system decomposition to work, the subsystem to be removed must only interact with the overall system through a set of 'channels' which will be replaced in the approximate system by an appropriate set of effective components. This ensures that all interaction between the exact subsystem and the remainder of the exact system will be completely captured by the approximate subsystem, as suggested in the second guideline for system decomposition (§3.1).

It is up to the systems analyst to determine a natural way in which a particular system may be decomposed. Often a system can be broken up with effective components representing power channels, processor channels, etc. Once a decomposition has been chosen, it is desirable to know whether or not a given set of effective components will accurately characterize the relationship between the subsystem and the remainder of the system. The following section discusses some of the conditions governing this decision.

3.3 Global vs. Channelized Dependence

The motivation behind the first two guidelines is clear from the examples in §3.2, but the third guideline is not immediately obvious. In order to understand this guideline we distinguish between two types of dependence. If a subsystem has a "global" dependence on some component in the remainder of the system, then the loss of this component makes all of the subsystem channels become non-functional. If a subsystem has a "channelized" dependence on some component in the remainder of the exact system, then the loss of this component causes specific subsystem channels to become unavailable, but not necessarily all subsystem channels.

Now, to explain the need for the third guideline, recall that the analysis technique as presented in chapter 2 represents a decomposition of the system architecture in which all of the information is derived from the underlying Markov models. In particular, the effective components of the approximate subsystem are an accurate representation of the exact subsystem in only the following way: the effective failure rates of these components are derived in such a way that the probability of being in specific failed or operational states is the same for the approximate and the exact subsystems.

The effective failure rates for the approximate subsystem are derived independently of any components outside of the subsystem in question. If, in fact, these effective components depend on the remainder of the system, then the states of the Markov model for the approximate subsystem no longer match the probability of the necessary conditioning events (i.e., joint probabilities of failed and unfailed effective channels). Consequently, the approximate system will no longer yield an accurate measure of the exact system's reliability.

Consider the two systems of figure 3.5. The processor core of the second system has a channelized dependence on the two power units in the remainder of the exact system, while the first system has no such dependence. According to the third guideline just proposed, only the processor core of figure 3.5a can accurately be replaced with two effective processors. We test this guideline by examining the reliability of both systems.

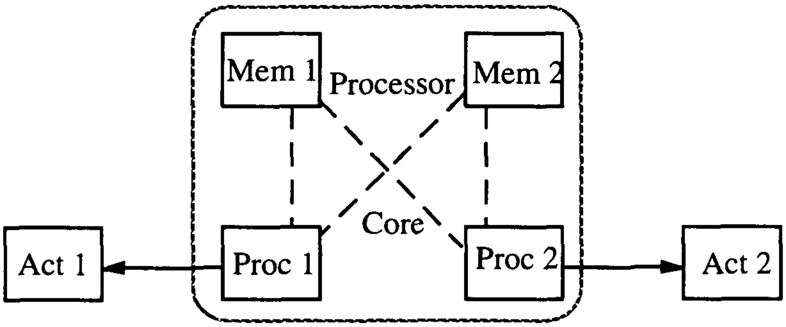


Figure 3.5a: No Channelized Dependence

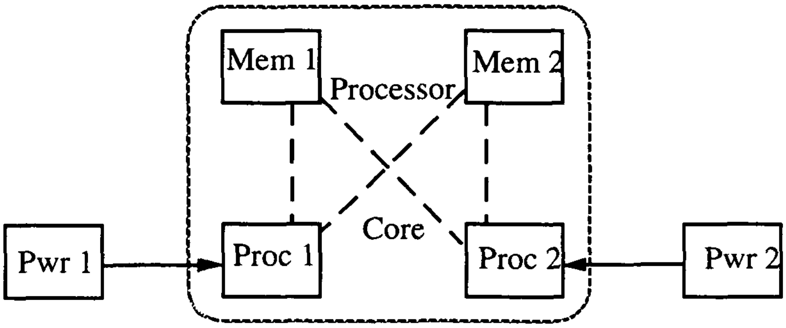


Figure 3.5b: Channelized Dependence

The processor core of either system in figure 3.5 can be replaced with two effective processors. In chapter 4 we will find that the failure rates for these components are derived independently of the remainder of the exact system. Consequently, expressions for the reliabilities and conditional reliabilities of these effective components can be derived by repeated application of the law of total probability on the processor core of figure 3.5. Such a set of expressions is shown in equations 3.2. In this set of expressions, the reliability of processors 1 and 2 are denoted by *R(Proc1)* and *R(Proc2)* respectively. The reliability for the two memory units operating in parallel is denoted *R(Mem1||Mem2)* as a shorthand for the component expression *1 - Q(Mem1)Q(Mem2)*, i.e., *R(Mem1 or Mem2).*

$$R(EffectProc1) = R(Proc1)R(Mem1 \| Mem2) \tag{3.2a}$$

$$R(EffectProc2) = R(Proc2)R(Mem1 \| Mem2) \tag{3.2b}$$

$$R(EffectProc1 | EffectProc2) = R(Proc1) \tag{3.2c}$$

$$R(EffectProc2 | EffectProc1) = R(Proc2) \tag{3.2d}$$

$$R(EffectProc1 | \overline{EffectProc2}) = \frac{R(Proc1)Q(Proc2)R(Mem1 \| Mem2)}{Q(Proc2) + R(Proc2)Q(Mem1)Q(Mem2)} \tag{3.2e}$$

$$R(EffectProc2 | \overline{EffectProc1}) = \frac{R(Proc2)Q(Proc1)R(Mem1 \| Mem2)}{Q(Proc1) + R(Proc1)Q(Mem1)Q(Mem2)} \tag{3.2f}$$

The law of total probability can be applied to the approximate system which results from the decomposition of the system in figure 3.5a, to produce a closed form expression for system reliability. This expression is listed in equation 3.3. Also, terms in equation 3.3 which represent effective components can be replaced by the corresponding term in equation 3.2. This yields equation 3.4 which expresses the system reliability solely in terms of components within the exact system. It can be determined by inspection that this expression does indeed accurately express reliability for the system of figure 3.5a. Thus for this sample system, hierarchical modeling can accurately be used to estimate reliability.

$$\begin{aligned} R(Sys) = {} & R(Act1 \| Act2)R(EffecProc2 | EffecProc1)R(EffecProc1) \\ & + R(Act1)R(EffecProc1 | \overline{EffecProc2})Q(EffecProc2) \\ & + R(Act2)R(EffecProc2 | \overline{EffecProc1})Q(EffecProc1) \end{aligned} \tag{3.3}$$

$$\begin{aligned} R(\mathit{Sys}) = {} & R(\mathit{Act1} \| \mathit{Act2}) R(\mathit{Proc1}) R(\mathit{Proc2}) R(\mathit{Mem1} \| \mathit{Mem2}) \\ & + R(\mathit{Act1}) Q(\mathit{Proc1}) R(\mathit{Proc2}) R(\mathit{Mem1} \| \mathit{Mem2}) \\ & + R(\mathit{Act2}) R(\mathit{EffecProc2} \mid \overline{\mathit{EffecProc1}}) Q(\mathit{EffecProc1}) \end{aligned} \tag{3.4}$$

Notice that hierarchical modeling of the system in figure 3.5b will yield the same effective component reliabilities of equation 3.2. In order to illustrate how this leads to inaccurate results we must derive two separate expressions for system reliability. First we use only components within the exact system, then we use effective component reliabilities.

Equation 3.5 can be derived by repeated application of the law of total probability on the system of figure 3.5b. This equation is simply the sum of the probabilities of all mutually exclusive operational states of the system in figure 3.5b.

$$\begin{aligned} R(\mathit{Sys}) = {} & R(\mathit{Mem1} \| \mathit{Mem2}) [R(\mathit{Pwr1}) R(\mathit{Pwr2}) R(\mathit{Proc1}) \\ & + R(\mathit{Pwr1}) Q(\mathit{Pwr2}) R(\mathit{Proc1}) \\ & + Q(\mathit{Pwr1}) R(\mathit{Pwr2}) R(\mathit{Proc2}) \\ & + R(\mathit{Pwr1}) R(\mathit{Pwr2}) Q(\mathit{Proc1}) R(\mathit{Proc2})] \end{aligned} \tag{3.5}$$

Equation 3.6 corresponds to the reliability of an approximate system in which the processor core of figure 3.5b is replaced by two effective processor components. This equation can be simplified by replacing all of the effective component reliabilities with the corresponding expression of equation 3.2. Equation 3.7 results from such an simplification.

$$\begin{aligned} R(\mathit{Sys}) = {} & R(\mathit{Pwr1}) R(\mathit{Pwr2}) R(\mathit{EffecProc1}) \\ & + R(\mathit{Pwr1}) Q(\mathit{Pwr2}) R(\mathit{EffecProc1}) \\ & + Q(\mathit{Pwr1}) R(\mathit{Pwr2}) R(\mathit{EffecProc2}) \\ & + R(\mathit{Pwr1}) R(\mathit{Pwr2}) R(\mathit{EffecProc2} \mid \overline{\mathit{EffecProc1}}) Q(\mathit{EffecProc1}) \end{aligned} \tag{3.6}$$

$$\begin{aligned} R(\mathit{Sys}) = {} & R(\mathit{Mem1} \| \mathit{Mem2}) [R(\mathit{Pwr1}) R(\mathit{Pwr2}) R(\mathit{Proc1}) \\ & + R(\mathit{Pwr1}) Q(\mathit{Pwr2}) R(\mathit{Proc1}) \\ & + Q(\mathit{Pwr1}) R(\mathit{Pwr2}) R(\mathit{Proc2}) \\ & + R(\mathit{Pwr1}) R(\mathit{Pwr2}) \frac{R(\mathit{Proc2}) Q(\mathit{Proc1})}{Q(\mathit{Proc1}) + R(\mathit{Proc1}) Q(\mathit{Mem1}) Q(\mathit{Mem2})}] \end{aligned} \tag{3.7}$$

Notice that the last terms in equations 3.5 and 3.7 differ by a normalization factor. In equation 3.7 this last term is conditioned on the event that effective processor one has failed. However, equation 3.5 indicates that no such conditioning is necessary. Since the reliability of the effective components in the hierarchical analysis is determined independently of the two power units, their dependence on these two power units is not accurately captured by such an analysis. For example, the joint probability that effective processor one will be functional while effective processor two has failed depends on the state of the second power unit. However, no such dependence is included in the hierarchical analysis. In short, if some subsystem exhibits a channelized dependence on the remainder of the exact system, then effective components of the approximate subsystem no longer accurately match the conditional reliabilities of the exact subsystem.

Finally, we would like to demonstrate that the problems exhibited by channelized dependence do not arise when the approximate subsystem exhibits a global dependence on some outside component or set of components (i.e., if all effective components within an approximate subsystem depend on the same external component or set of components). In such a situation the conditional reliabilities for the effective components of the approximate subsystem are the product of the conditional reliabilities obtained by analyzing the exact subsystem independently and the reliability of any external components necessary for the subsystem to operate.

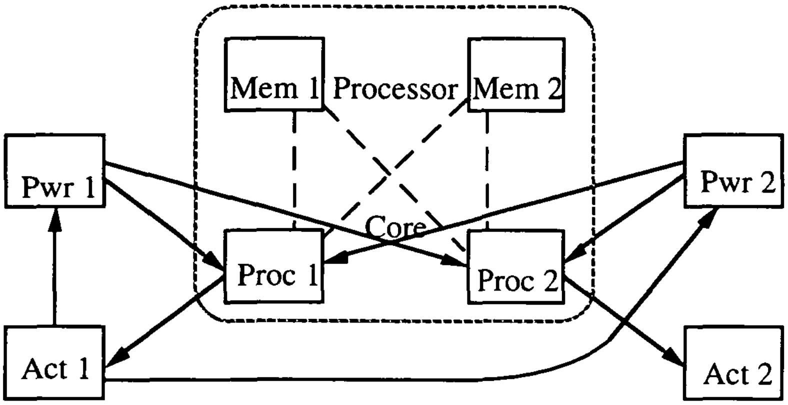


Figure 3.6: Global Dependence

Consider the system of figure 3.6. The processor core for this system is the same one used in all of the previous examples in this chapter. In this case the processor core requires power from at least one of the two power units operating in parallel. These power units depend on the availability of actuator 1. Actuators 1 and 2 rely on the availability of processor channels 1 and 2. In order for the system to be considered functional, at least one of the two actuators must be functional.

In order to explore the effect of this cross-strapping, we break up the expression for system reliability in terms of actuator 1, and the two power units operating in parallel. The resulting expression is listed in equation 3.8. Notice, that *R(Pwr1||Pwr2)* denotes the reliability of the two power units operating in parallel. Also notice that if either actuator 1 fails or the parallel power unit configuration fails, the system fails.

$$R(Sys) = R(Sys|Pwr1||Pwr2Act1)R(Pwr1||Pwr2)R(Act1) \qquad (3.8)$$

Clearly the addition of the external dependencies of figure 3.6 change the reliability of the exact system as well as the reliability of the processor core. However, the conditional reliability of equation 3.8 can now be considered independently of actuator 1, and the two power units. Expanding this term yields equation 3.9.

$$R(Sys) = R(Act1)R(Pwr1||Pwr2)R(EffectProc1) \qquad (3.9)$$

The external dependencies of the system in figure 3.6 alter the subsystem reliability. However, the resulting expression for system reliability depends on the reliability for the same effective processors of figure 3.3.

In general, it will be the case that either some external component is available, or the necessary external component/s has/have failed. In the former case the subsystem can be considered independently of the remainder of the exact system. In the latter case the system can be considered in the absence the subsystem altogether. We conclude that if a subsystem exhibits only a global dependence on some external component or set of components, the reliability of this subsystem may be considered independently of the remainder of the exact system.

3.3 Multiple Level Decomposition

Some systems will have subsystems which could themselves be analyzed hierarchically. The processor core presented above for example contains a parallel combination of memory units which could be regarded as an independent subsystem. This memory subsystem could be replaced by one effective memory unit, and this effective memory unit could in turn be used to evaluate the reliability of the processor core. Alternatively, some systems may be composed of several subsystems which may each be replaced with a different set of effective components.

In general, there may be several levels and several different components to any given hierarchical analysis. However, there may also be some more detailed interaction between different subsystems which can be captured hierarchically. Figure 2.9 gives an overall view of the hierarchical modeling process. Without loss of generality, we may assume that some of the components in the exact system of this diagram are themselves effective components. But, what happens when information from an approximate subsystem is needed at two distinct levels within a decomposition?

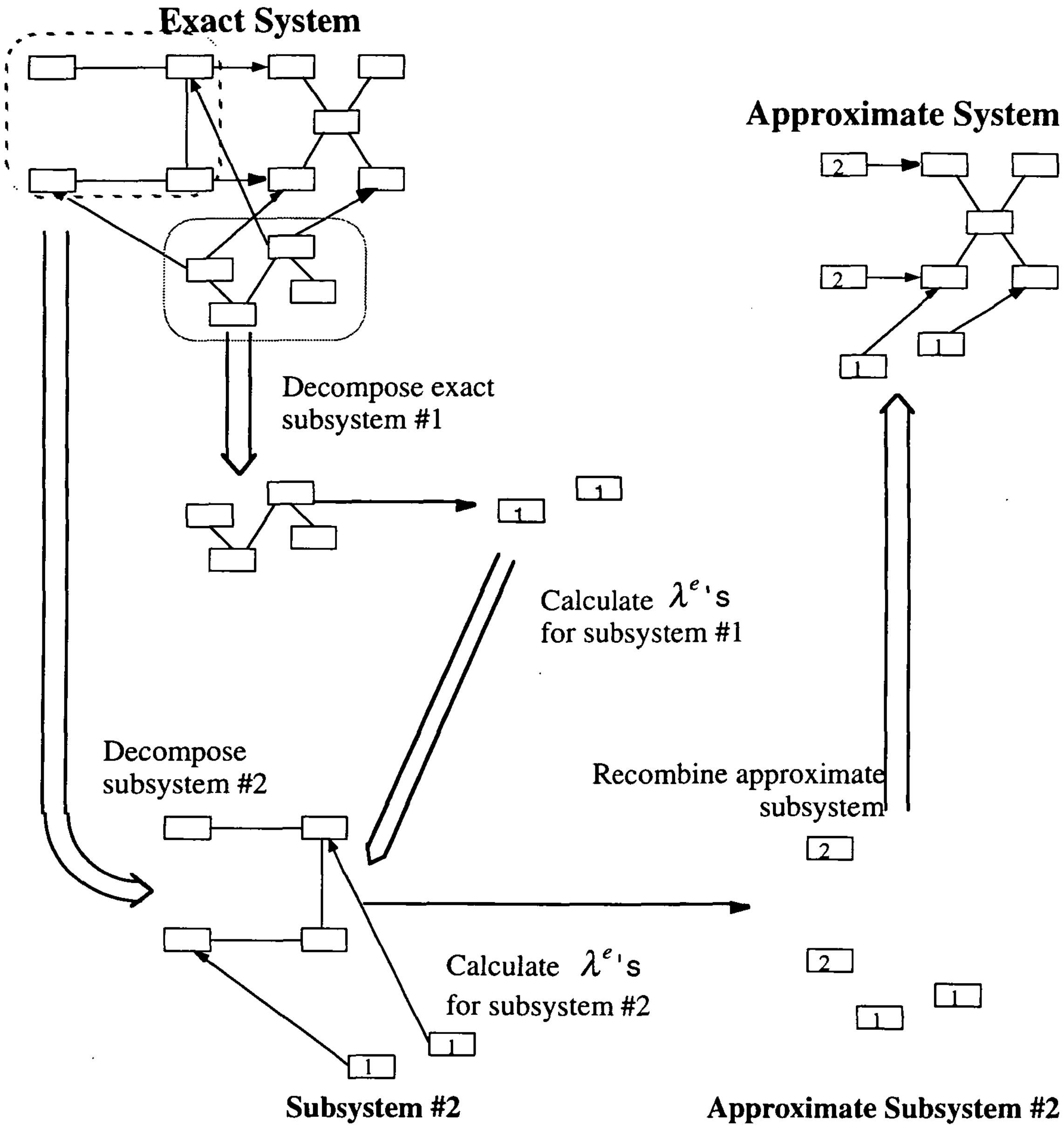

Exact System
Approximate System
2
2
1
1
Decompose exact subsystem #1
1
1
Calculate λ^e's for subsystem #1
Decompose subsystem #2
Recombine approximate subsystem
2
Calculate λ^e's for subsystem #2
2
1
1
1
1
Subsystem #2
Approximate Subsystem #2

Figure 3.7: Overview of Multiple Level Hierarchical Modeling

Figure 3.7 represents an overall view of a more intricate hierarchical analysis . In the situation represented here there are two subsystems which may be decomposed from the exact system. Information about the functional status of one approximate subsystem is needed by both the second subsystem, and the remainder of the exact system.

In the situation depicted in figure 3.7, effective failure rates for subsystem 1 are calculated as though subsystem 2 were not present. However, the effective failure rates for the second subsystem need to have some dependence on the effective components of the first subsystem. In chapter 4 we will see that this can be accurately captured by including the components of the first subsystem in the state dependence of the second subsystem. For example, assume that the second subsystem is the processor core of figure 2.11, and that the first subsystem is the cooling unit of figure 3.8.

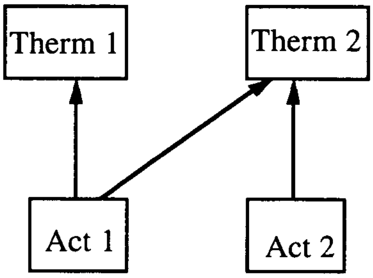


Figure 3.8: Exact Subsystem 1

Thermal units 1 and 2 of figure 3.8 provide cooling to certain components within the exact system. The amount of cooling is controlled by actuators 1 and 2. Notice that thermal unit 1 can only function properly if actuator 1 is available, whereas thermal unit 2 may be controlled by either actuator.

The thermal subsystem indicated in figure 3.8 can be replaced by two effective thermal units within the exact system of figure 3.7. Also, assume that the second subsystem of 3.7 is the processor core in figure 2.11. This processor subsystem may also be replaced by two effective processor components.

Let the terms $\lambda^e_{T1}(\cdot)$, and $\lambda^e_{T2}(\cdot)$ denote the effective failure rates for the resulting approximate thermal subsystem, and let $\lambda^e_{P1}(\cdot)$, and $\lambda^e_{P2}(\cdot)$ denote the effective processor failure rates. We know from the discussion of chapter 2 that $\lambda^e_{T1}(\cdot)$ will be a function of whether or not the second thermal unit has failed, and similarly for $\lambda^e_{T2}(\cdot)$. However, since the thermal subsystem does not depend on the processor subsystem, neither of these failure rates depends on the functional status of the two effective processors. These two processors on the other hand do depend on the functional status of the two effective thermal units. This dependence is captured by a dependence of $\lambda^{e}_{P1}(\cdot)$, and $\lambda^{e}_{P2}(\cdot)$ on the state of the approximate thermal subsystem.

3.5 Conclusions

In this chapter certain guidelines and rules for system decomposition are presented. Arguments and examples are presented in this chapter which support the following guidelines:

1. In selecting a candidate decomposition, the analyst should look for a subsystem which contains a substantial amount of complexity, but which has a very simple interface with the remainder of the exact system.

2. Only information summarized by the effective components will be available in the approximate system. No detailed information of the status of the subsystem beyond these effective components is available to the approximate system.

3. Components outside of the subsystem may depend on the effective components in any way; however, effective components may only have a "global" dependence on components outside of the subsystem.

The arguments presented here do not provide rigorous proof of the accuracy of a given hierarchical analysis. Rather, these arguments are presented in order to develop some intuition about the nature of this problem. Further insight is provided through closed form expressions for system reliability, which are produced by application of the law of total probability.

Although only simple decomposition was explored in detail, more complicated hierarchical analyses are achieved through repeated application of these principles. The overview of figure 3.7 is presented as one example of a more complicated hierarchical analysis.

Chapter 4

Properties of Effective Component Failure Rates

In chapter 3, it was shown that the reliability for many systems can be expressed in terms of certain conditional reliabilities and the probabilities of their conditioning events (e.g., equation 3.1). The hierarchical technique presented in this thesis captures the probability of those events in the approximate subsystem. The effective component failure rates are used to summarize multiple aggregate state information, whereas more traditional hierarchical techniques only indicate availability of a given subsystem [Abraham].

Effective component failure rates capture state information for subsystems which may be very complex internally. In some sense, we use effective failure rates to hide this complexity from the higher level model. However, as mentioned in previous chapters, effective failure rates may exhibit complicated behavior as well (i.e., state dependence and time variation). This chapter explores how effective failure rates capture different aspects of the exact subsystem, and its interaction with the remainder of the exact system. We begin with a general method of solving for effective failure rates.

4.1 Calculating the Effective Failure Rate

Recall from chapter 2 that continuous transition Markov models can be described by the matrix equation 4.1a. Notice that if a discrete time approximation is made to the continuous time differential, then we can use equation 4.1a to derive an expression which relates the effective failure rates to the probability of being in different states of the approximate subsystem model. If Euler's method is used to make the discrete time approximation, the matrix expression of equation 4.1b results.

$$\frac{d\vec{P}(t)}{dt} = [A]\vec{P}(t) \tag{4.1a}$$

$$\frac{\Delta\vec{P}(t)}{T_{avg}} = \vec{A}\left[\vec{P}(nt)\right] \Rightarrow \vec{A} = \frac{\Delta\vec{P}(t)}{T_{avg}}\left[\vec{P}(nt)\right]^{-1} \tag{4.1b}$$

The matrix in equation 4.1b can contain probabilities from states several failure levels apart. Thus, the resulting matrix elements can easily differ by many orders of magnitude. This makes solving for effective failure rates a numerically sensitive procedure. This topic is more fully discussed in chapter 5. There we will see that a reasonable approach to solving for these effective failure rates is to obtain a closed form expression for each of the unknown effective transition rates which make up the effective failure rates.

An approximate subsystem with two effective components will generally be composed of four unknown effective transition rates (as in figure 2.14). An approximate subsystem with three effective components will generally be composed of twelve unknown effective transition rates. In order to produce reasonably simple closed form expressions for large numbers of effective transition rates a reasonably high degree of sparsity is required from the transition matrix for the approximate subsystem model. In particular, each state within the Markov model for the approximate subsystem may only have one unknown transition rate associated with all entering transitions. Thus, even for large Markov models we need only solve one equation for each unknown effective transition rate. This approach minimizes numerical evaluation problems (see chapter 5).

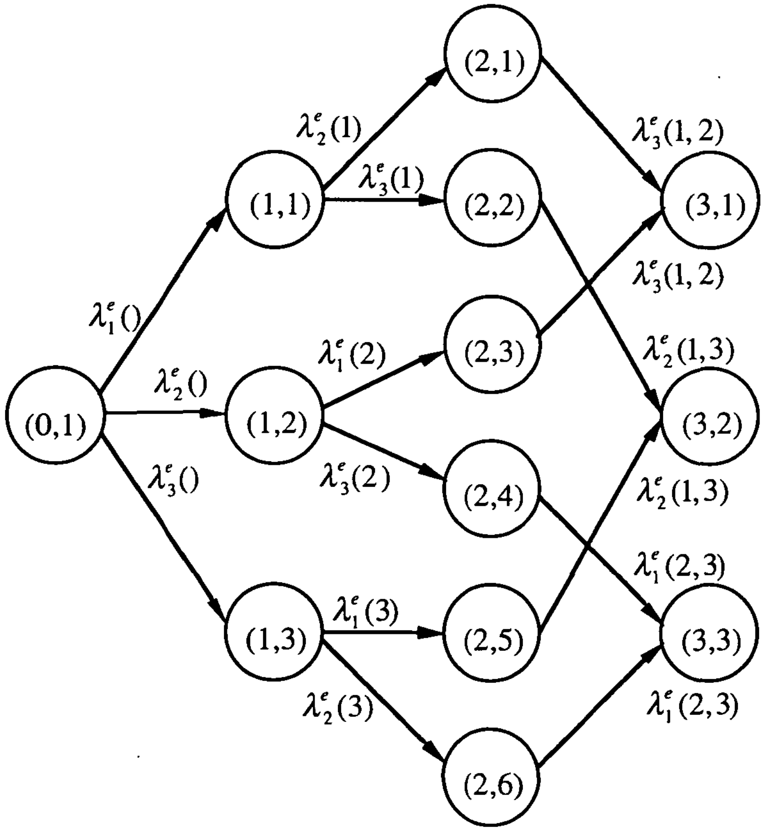


Figure 4.1: General 3 - Component Model

In general this approach to generating the Markov model for the approximate subsystem ends with a series of system loss states with one unknown entering transition rate. The equations for these states are solved for the unknown transition rate and the resulting solution can be used in the equation for the source state. Consider for example a general approximate subsystem with three effective components. Such an approximate subsystem could be described by the Markov model of figure 4.1.

Consider equations 4.2, which describe states (3, 1) and (2, 1). Although state (3, 1) has two entering transitions, both of these transitions have the same unknown effective rate, $\lambda_3^e(1,2)$. Also, the solution for this effective transition rate is independent of any of the other state equations. The effective rate associated with the transition into state (2, 1), $\lambda_2^e(1)$, depends only on states (1, 1), (2, 1) and $\lambda_3^e(1,2)$.

$$\frac{dP_{(3,1)}}{dt} = \lambda_3^e(1,2)[P_{(2,1)}(t) + P_{(2,3)}(t)] \rightarrow \frac{\Delta P_{(3,1)}}{T_{avg}} = \lambda_3^e(1,2)[P_{(2,1)}(nT_{avg}) + P_{(2,3)}(nT_{avg})]$$

$$\Rightarrow \lambda_3^e(1,2) = \frac{\Delta P_{(3,1)}}{T_{avg}[P_{(2,1)}(nT_{avg}) + P_{(2,3)}(nT_{avg})]} \tag{4.2a}$$

$$\frac{dP_{(2,1)}}{dt} = \lambda_2^e(1)P_{(1,1)}(t) - \lambda_3^e(1,2)P_{(2,1)}(t) \rightarrow \frac{\Delta P_{(2,1)}}{T_{avg}} = \lambda_2^e(1)P_{(1,1)}(nt) - \lambda_3^e(1,2)P_{(2,1)}(nt)$$

$$\Rightarrow \lambda_2^e(1) = \frac{\Delta P_{(2,1)}}{T_{avg}P_{(1,1)}(nT_{avg})} + \frac{\lambda_3^e(1,2)P_{(2,1)}(nT_{avg})}{P_{(1,1)}(nT_{avg})} \tag{4.2b}$$

In general, each effective transition rate depends on its source state probability history, destination state probability history, and all effective rates associated with transitions out of its destination state. Thus, all effective transition rates can be calculated from known state probabilities, and previously calculated effective rates. Thus, the solution for $\lambda_2^e(1)$ uses the previously calculated $\lambda_3^e(1,2)$ and known state probabilities. Finally, notice that although many of the states in figure 4.1 could be aggregated according to the rules for state aggregation (see §2.2), this would produce states with more than one unknown entering transition rate.

Now consider the system in figure 4.2. For this system a processor channel is considered functional if the corresponding processor is functional (e.g., processor channel 1 is functional if processor 1, and all of its supporting hardware are unfailed). Assume that the functionality of both sets of sensors and actuators depends on whether or not a given processor channel is functional. Also assume that the processor core does not depend on any components in the remainder of the exact system. For the time being no assumptions are made about the internal connectivity of the processor core itself.

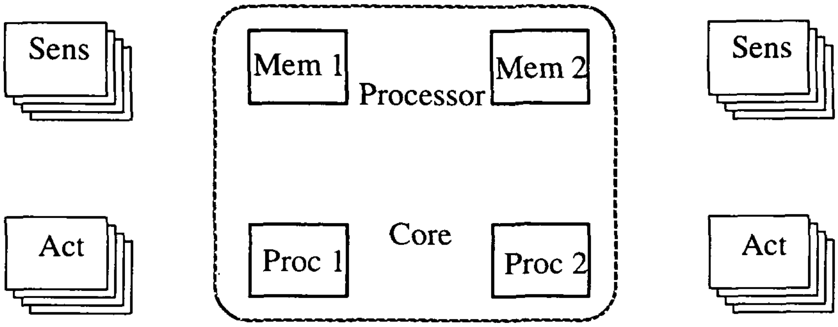


Figure 4.2: Sample System with Arbitrary Cross Strapping

Suppose we would like to replace the above processor core with two effective components which represent the functional status of processor channel 1 and 2 respectively. Since the reliability of the overall system depends only on the operational status of both processor channels, this set of effective components characterizes all of the information necessary to estimate system reliability.

The reliability of the approximate processor core can be described using the Markov model of figure 4.3. Effective failure rates in this model use the same notation of figure 2.13. The states for this model are also defined as in figure 2.13. For example state (0,1) corresponds to the state in which neither effective component has failed, and state (1,1) corresponds to the state in which effective processor 1 has failed, but effective processor 2 remains unfailed.

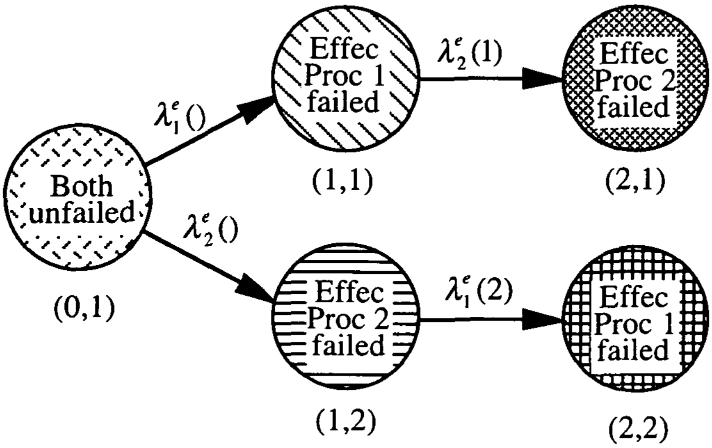


Figure 4.3: Approximate Subsystem Markov Model

The Markov model for the approximate processor core is described by a system of state equations. This system of state equations may be discretized according to Euler's method (see §2.1.1) These state equations as well as their resulting discretization are listed in equation 4.3.

$$\frac{dP_{(0,1)}}{dt} = -[\lambda_1^e() + \lambda_2^e()]P_{(0,1)}(t) \rightarrow \frac{\Delta P_{(0,1)}}{T_{avg}} = -[\lambda_1^e() + \lambda_2^e()]P_{(0,1)}(nT_{avg}) \qquad (4.3a)$$

$$\frac{dP_{(1,1)}}{dt} = \lambda_1^e()P_{(0,1)}(t) - \lambda_2^e(1)P_{(1,1)}(t)$$

$$\rightarrow \frac{\Delta P_{(1,1)}}{T_{avg}} = \lambda_1^e()P_{(0,1)}(nT_{avg}) - \lambda_2^e(1)P_{(1,1)}(nT_{avg}) \qquad (4.3b)$$

$$\frac{dP_{(1,2)}}{dt} = \lambda_2^e()P_{(0,1)}(t) - \lambda_1^e(2)P_{(1,2)}(t)$$

$$\rightarrow \frac{\Delta P_{(1,2)}}{T_{avg}} = \lambda_2^e()P_{(0,1)}(nT_{avg}) - \lambda_1^e(2)P_{(1,2)}(nT_{avg}) \qquad (4.3c)$$

$$\frac{dP_{(2,1)}}{dt} = \lambda_2^e(1)P_{(1,1)}(t) \rightarrow \frac{\Delta P_{(2,1)}}{T_{avg}} = \lambda_2^e(1)P_{(1,1)}(nT_{avg}) \qquad (4.3d)$$

$$\frac{dP_{(2,2)}}{dt} = \lambda_1^e(2)P_{(1,2)}(t) \rightarrow \frac{\Delta P_{(2,2)}}{T_{avg}} = \lambda_1^e(2)P_{(1,2)}(nT_{avg}) \qquad (4.3e)$$

In equations 4.3 $P_{(0,1)}$, $P_{(1,1)}$, $P_{(1,2)}$, $P_{(2,1)}$, and $P_{(2,2)}$ represent the probabilities of being in the corresponding states within the approximate subsystem's Markov model (figure 4.3). Notice that the discretization, or *averaging interval*, takes on special significance in this analysis. Recall that the effective failure rates are continuous functions of time which we are approximating as piece-wise constant. T_{avg} defines the length of the interval over which these failure rates are taken to be constant, i.e., this parameter defines the length of the averaging interval.

The discrete approximation in equations 4.3 can be used to derive a closed form expression for the effective component failure rates in terms of the state probabilities $P_{(0,1)}$, $P_{(1,1)}$, $P_{(1,2)}$, $P_{(2,1)}$, and $P_{(2,2)}$. In particular, we derive solutions for the effective transition rates to the second failure level, and then use these solutions to solve for transition rates to the first failure level. The resulting solutions are listed in equations 4.4.

$$\lambda_1^e(2) = \frac{\Delta P_{(2,2)}}{T_{avg}P_{(1,2)}(nT_{avg})} \qquad (4.4a)$$

$$\lambda_2^e(1) = \frac{\Delta P_{(2,1)}}{T_{avg}P_{(1,1)}(nT_{avg})} \qquad (4.4b)$$

$$\lambda_1^e() = \frac{\Delta P_{(1,1)}}{T_{avg}P_{(0,1)}(nT_{avg})} + \frac{\lambda_2^e(1)P_{(1,1)}(nT_{avg})}{P_{(0,1)}(nT_{avg})} \qquad (4.4c)$$

$$\lambda_2^e() = \frac{\Delta P_{(1,2)}}{T_{avg} P_{(0,1)}(nT_{avg})} + \frac{\lambda_1^e(2) P_{(1,2)}(nT_{avg})}{P_{(0,1)}(nT_{avg})} \tag{4.4d}$$

Thus, the solutions of the higher failure level transition rates are used to solve for lower failure transition rates.

Notice that equations 4.4 do not explicitly depend on the exact processor core. This means that these expressions for the effective failure rates are independent of the details of the exact subsystem. But, in order to determine numerical values for the state probabilities of equations 4.3, we need to evaluate the exact subsystem and match its states to those in figure 4.3. Consequently, the internal details of the exact processor core will affect the numerical behavior of the effective failure rates.

4.2 Affect of Exact Subsystem Architecture on Effective Rates

4.2.1 No Shared Resources

Given the same system of figure 4.2 and the same proposed decomposition, assume the resulting processor channels share no resources. This is the case in which processor 1 depends only on the availability of memory 1, processor 2 depends only on the availability of memory 2, and the two memory units do not depend on each other or any other components. The Markov model for such an exact subsystem is shown in figure 4.4. The time-invariant transition rates of this Markov model have subscripts that indicate specific components of the exact subsystem (e.g., λ_{M2} is the failure rate for memory unit 2).

The states in the Markov model for the exact subsystem (figure 4.4) are shaded to indicate which state they correspond to in the Markov model for the approximate subsystem (figure 4.3). In this case, all of the states in the model for the exact processor core can be mapped into the states of the Markov model in figure 4.3. Thus, the probability of being in any given state of the Markov model for the approximate processor core is the sum of the appropriate state probabilities in the Markov model for the exact processor core (e.g., the probability of being in state (1,2) in figure 4.3 is the sum of the probabilities of being in states (1,3), (1,4), (2,9), and (2,12) in figure 4.4).

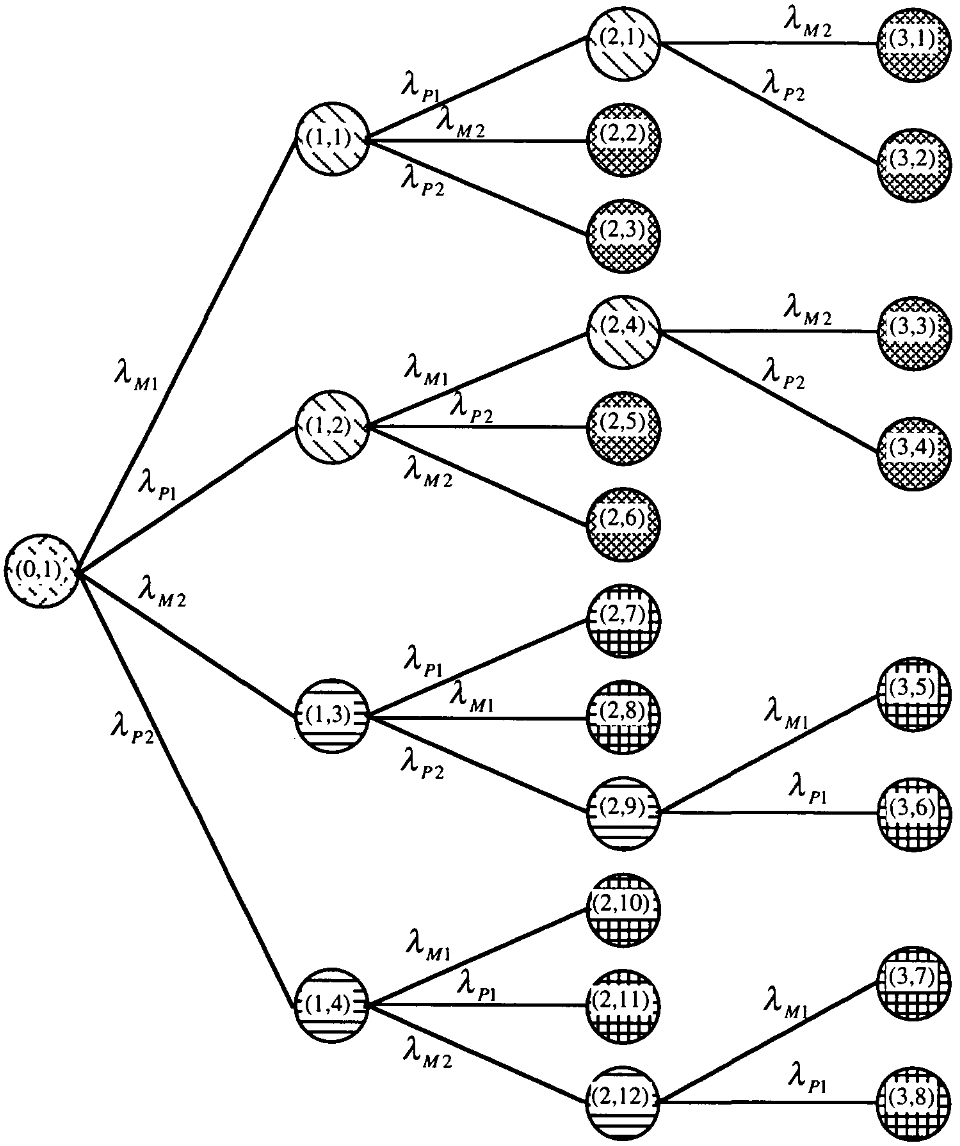


Figure 4.4: Markov Model for 'Processor Core' - No Shared Resources

Since the effective components in this sample system share no resources they are completely independent of one another. Consequently, we expect the failure rate of either effective component to be independent of the operational status of the other effective component. Notice that both processor channels represent a simple series configuration composed of a memory and processor unit. This leads us to expect the failure rate for each of these effective processors to be the sum of the failure rates for the memory and processor. For example, $R(\mathit{EffectProc1}) = e^{-\lambda_{P1}t}e^{-\lambda_{M1}t} = e^{-(\lambda_{P1}+\lambda_{M1})t} = e^{-\lambda_1^e t}$, i.e., the failure rate for effective processor 1 is the sum of the failure rates for processor 1 and memory 1. Thus, $\lambda_1^e() = \lambda_1^e(2) = \lambda_{P1} + \lambda_{M1}$.

Now examine the Markov model for the exact processor core. Close examination of this Markov model reveals that all the states which correspond to the same state in the Markov model of figure 4.3 have the same exit transitions. For example, all of the states in figure 4.4 which correspond to state (1, 1) in the effective Markov model have exactly two exit transitions, and these exit transitions are associated with a memory failure and a processor failure in every case. Therefore, the Markov model of figure 4.4 can be reduced exactly into the form of figure 4.3 [Babcock1]. The associated transition rates for this aggregated Markov model would be the sum of the processor and memory unit failure rates.

For the case in which no resources are shared between channels of the system in figure 4.3, the effective failure rate should have a very simple form. This expectation is confirmed with the following numerical example. The exact processor core was evaluated with a failure rate of 10^{-4} (failures per hour) assigned to each of the memory units and 10^{-5} assigned to each of the processor units. State probabilities were determined every 100 hours from 0 to 1000 hours, and the results were used to determine the state probabilities for the Markov model of the approximate subsystem. Using the data obtained, the effective failure rates were determined over successive 100 hour intervals. This corresponds to selecting an averaging interval, T_{avg} of 100 hours. Values produced for the effective failure rates are listed in table 4.1.

Time(hr's)	$\lambda_1^e()$	$\lambda_2^e()$	$\lambda_1^e(2)$	$\lambda_2^e(1)$
0.0 - 100.0	1.1000e-04	1.1000e-04	1.1050e-04	1.1050e-04
100.0 - 200.0	1.1000e-04	1.1000e-04	1.1016e-04	1.1016e-04
200.0 - 300.0	1.1000e-04	1.1000e-04	1.1010e-04	1.1010e-04
300.0 - 400.0	1.1000e-04	1.1000e-04	1.1007e-04	1.1007e-04
400.0 - 500.0	1.1000e-04	1.1000e-04	1.1005e-04	1.1005e-04
500.0 - 600.0	1.1000e-04	1.1000e-04	1.1004e-04	1.1004e-04
600.0 - 700.0	1.1000e-04	1.1000e-04	1.1004e-04	1.1004e-04
700.0 - 800.0	1.1000e-04	1.1000e-04	1.1003e-04	1.1003e-04
800.0 - 900.0	1.1000e-04	1.1000e-04	1.1003e-04	1.1003e-04
900.0 - 1000.0	1.1000e-04	1.1000e-04	1.1002e-04	1.1002e-04

Table 4.1: Test Results - No Shared Resources

Since $\lambda_1^e(2) \approx \lambda_1^e()$ and $\lambda_2^e(1) \approx \lambda_2^e()$ to within the level of accuracy of our test data, this supports the argument that the effective component failure rates are independent of state. Since all of the above failure rates are constant from one time interval to another, we also conclude that effective component failure rates are time-invariant. Finally, notice that the resulting effective failure rate is indeed the sum of the processor and memory failure rates.

4.2.2 Shared Resources

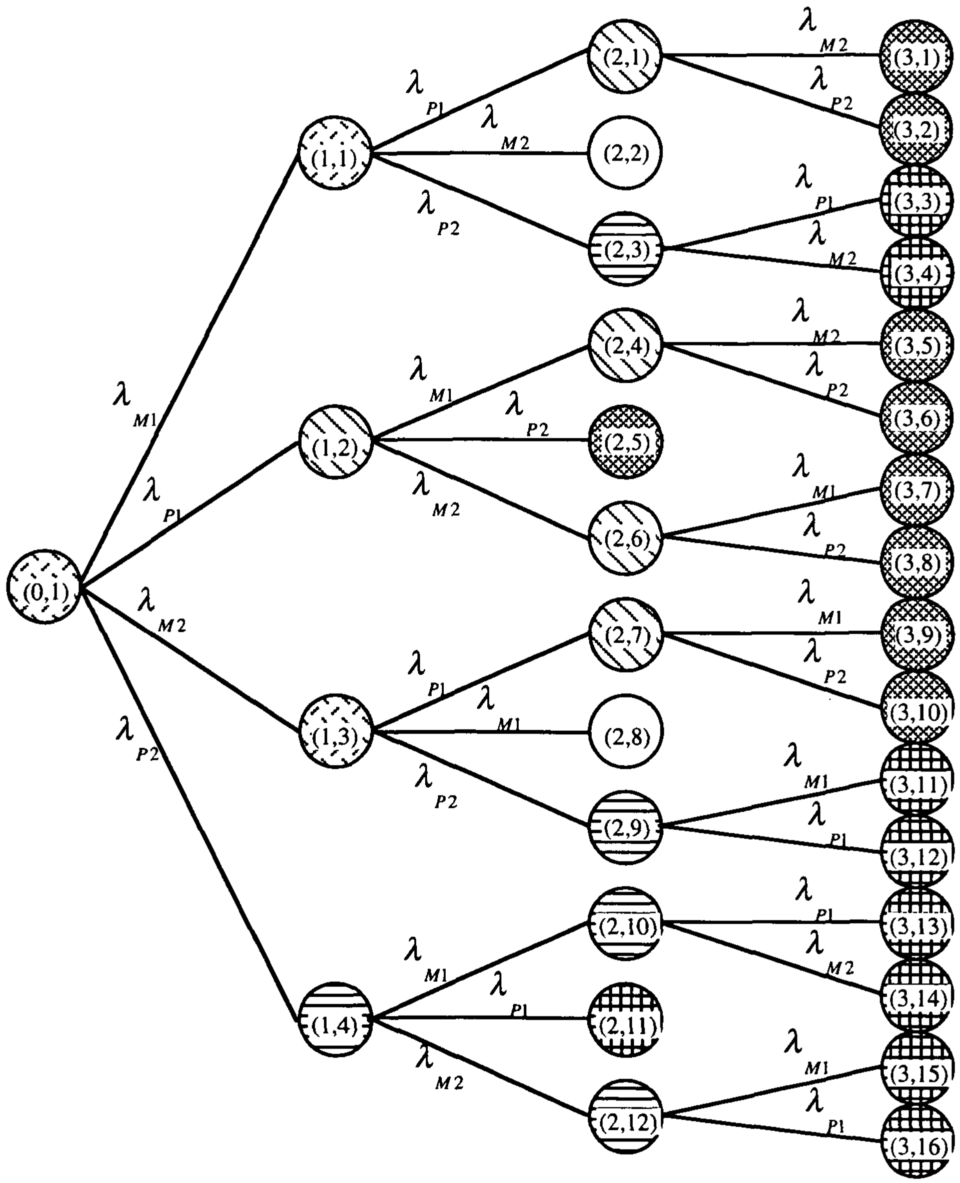


Figure 4.5: Markov Model for 'Processor Core' - Shared Resources

In the previous section the two effective components were independent of one another because they did not share resources. Consider now the case where the effective components share memory. In particular, what happens if processor 1 remains functional so long as either memory 1 or memory 2 is operational, and similarly for processor 2? If we again try to represent this new processor core with the two effective processors, we may still use the same Markov model for the approximate subsystem (figure 4.3). However, the exact processor core is now described by the Markov model of figure 4.5.

The Markov model of figure 4.5 contains states which do not correspond to any of the states in the Markov model for the approximate subsystem. In particular, states (2, 2) and (2, 8) correspond to a simultaneous failure of both effective components (i.e., a common mode failure). Recall however, that the goal of evaluating the exact subsystem is to determine specific state probabilities and to capture that information in the effective component failure rates. If the remainder of the approximate system depends on which effective components have failed but not on the order of their failure, then the probability of states (2, 2) and (2,8) can be mapped arbitrarily into either system loss state within the Markov model of figure 4.3. In order to retain the symmetry of the original system, the probability of being in states (2, 2) and (2, 8) are divided evenly between the two system loss states in the Markov model for the approximate subsystem.

In, §4.2.1, the effective component failure rates for the case where there is no resource sharing were neither state nor time dependent. By looking at the mapping of states between the Markov models for the exact and approximate subsystem, we saw that the effective component transition rates corresponded to a specific set of transitions in the Markov model for the exact subsystem. For example, all of the states in figure 4.4 which correspond to state (1, 1) in the approximate subsystem model have exactly two exit transitions, and these exit transitions are associated with a memory failure and a processor failure in every case. However, it can be seen from the Markov model of figure 4.5 that this is not the case when resources are shared.

Transitions out of state (0, 1) in the Markov model for the approximate subsystem (figure 4.3) correspond either to a processor failure in the exact subsystem or a common mode failure attributable to the consecutive loss of both memory units (i.e., before any other component failures). Transitions out of the first failure level correspond to a mixture of either a processor or memory failure. For example, the transition from state (1, 1) in the Markov model of figure 4.3 to state (2, 1) corresponds to either the loss of processor 2 at the first failure level of the Markov model in figure 4.5 or to the loss of processor 2 or a memory unit at failure level 2. Since the probability flows from failure level 0 to the system loss states, at different times either one of these failure modes will dominate $\lambda_2^e(1)$. This suggests that for short mission times we would expect both effective failure rates to look like a processor failure, and for longer times we expect these failure rates to look like the sum λ_M and λ_P. Although, it is hard to characterize the effect of the common mode failure on the resulting effective component failure rates, it seems clear that the resulting failure rates will depend on the state of the Markov model in figure 4.3, as well as with time.

We now see that effective component failure rates can capture many different failure modes for a given subsystem channel. Differences attributable to the loss of one or more channels are captured by the state dependence, and differences at different failure levels are captured by time dependence. In order to illustrate this effect, the numerical study of table 4.1 was conducted again using state probabilities obtained from the Markov model of figure 4.5, and the same component failure rates for the exact subsystem. The results of this study are listed in table 4.2.

Time(hr's)	$\lambda_1^e()$	$\lambda_2^e()$	$\lambda_1^e(2)$	$\lambda_2^e(1)$
0.0 - 100.0	1.0495e-05	1.0495e-05	1.1050e-04	1.1050e-04
100.0 - 200.0	1.1465e-05	1.1465e-05	1.1016e-04	1.1016e-04
200.0 - 300.0	1.2408e-05	1.2408e-05	1.1010e-04	1.1010e-04
300.0 - 400.0	1.3324e-05	1.3324e-05	1.1007e-04	1.1007e-04
400.0 - 500.0	1.4213e-05	1.4213e-05	1.1005e-04	1.1005e-04
500.0 - 600.0	1.5078e-05	1.5078e-05	1.1004e-04	1.1004e-04
600.0 - 700.0	1.5919e-05	1.5919e-05	1.1004e-04	1.1004e-04
700.0 - 800.0	1.6738e-05	1.6738e-05	1.1003e-04	1.1003e-04
800.0 - 900.0	1.7534e-05	1.7534e-05	1.1003e-04	1.1003e-04
900.0 - 1000.0	1.8309e-05	1.8309e-05	1.1002e-04	1.1002e-04

Table 4.2: Test Results - Shared Resources

The effective component failure rate changes with state by a full order of magnitude. However, for the range of times in table 4.2, the effective transition rates remain on the same order of magnitude. This suggests that dependence on state may be significantly stronger than dependence on time. As would be expected due to the symmetry of the exact subsystem, the failure rates for effective processors 1 and 2 are equal for all time (i.e., $\lambda_1^e = \lambda_2^e$). Surprisingly, the effective transition rates, $\lambda_1^e(2)$ and $\lambda_2^e(1)$, did not reflect the failure modes as we expected. This indicates that the presence of a common mode failure complicates our intuition about the effective failure rates. This also suggests the need for a more detailed analysis of the relationship between the

Markov model for the exact system and the effective component failure rates. In the next section we examine this problem more closely by explicitly calculating a common mode failure rate.

4.2.3 Decomposition with Common Mode Failure

We noted in §4.2.2 that the Markov model for the exact processor core contains two states which correspond to a simultaneous failure of the two effective components. It was also noted that the addition of this common mode failure complicates our intuition about the effective component failure rates. In order to clarify the effect of the common mode failure, we explicitly include a common mode failure transition in the Markov model for the approximate subsystem. The resulting Markov model for the approximate subsystem is shown in figure 4.6.

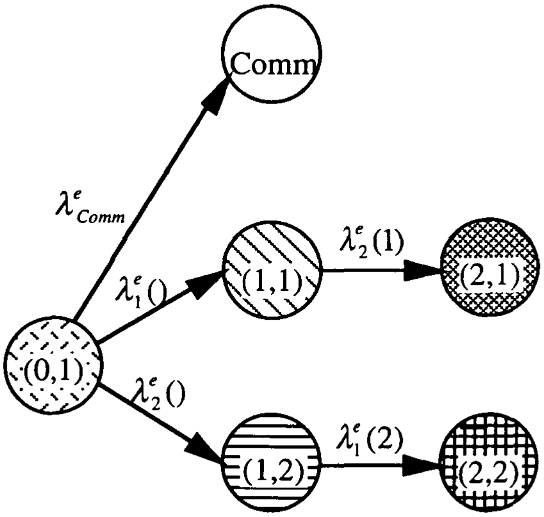


Figure 4.6: Approximate Subsystem Model with Common Mode Failure

The transition rate associated with the common mode failure is denoted in the Markov model of figure 4.6 by λ^e_{Comm}. Consequently, the state equations of the Markov model for the approximate subsystem need to be revised to incorporate equations 4.5. Equation 4.5a replaces equation 4.3a, and equation 4.5b is an additional equation which describes the common mode failure state.

$$\frac{dP_{(0,1)}}{dt} = -[\lambda^e_1() + \lambda^e_2() + \lambda^e_{Comm}]P_{(0,1)}(t)$$

$$\rightarrow \frac{\Delta P_{(0,1)}}{T_{avg}} = -[\lambda^e_1() + \lambda^e_2() + \lambda^e_{Comm}]P_{(0,1)}(nT_{avg}) \qquad (4.5a)$$

$$\frac{dP_{Comm}}{dt} = \lambda^e_{Comm} P_{(0,1)}(t) \rightarrow \frac{\Delta P_{Comm}}{T_{avg}} = \lambda^e_{Comm} P_{(0,1)}(nT_{avg}) \qquad (4.5b)$$

For this revised decomposition we can again produce a direct mapping between states in the Markov model of figure 4.5 and the states in the Markov model for the approximate subsystem. We would like to add equation 4.6 to the set of closed form solutions in equations 4.4.

$$\lambda^e_{Comm} = \frac{\Delta P_{Comm}}{T_{avg} P_{(0,1)}(nT_{avg})} \qquad (4.6)$$

For this decomposition, all single effective processor transitions from the full up state correspond to a processor failure within the exact subsystem. Consequently, we expect the rates $\lambda^e_1()$ and $\lambda^e_2()$ to equal λ_P for all times. Also, in the previous section it was suggested that $\lambda^e_1(2)$ and $\lambda^e_2(1)$ would vary with time to match two distinct failure modes. Because of the influence of the common mode failure, these transition rates did vary in time, but not exactly as expected. Since the common mode failure transition no longer has any affect on these transition rates, we now expect $\lambda^e_1(2)$ and $\lambda^e_2(1)$ to approximate λ_P for short times and the sum of λ_P and λ_M for longer times.

Our intuition about the behavior of $\lambda^e_1(2)$ and $\lambda^e_2(1)$ is supported by examining the relation between these effective failure rates and failure rates within the Markov model for the exact subsystem. In particular consider the equation for state (1, 2) in the Markov model for the approximate subsystem (equation 4.3c). The flow of probability out of this state is defined by the product $P_{(1,2)}\lambda^e_1(2)$. State (1, 2) in the Markov model for the approximate subsystem maps to the sum of states (1, 4), (2, 3), (2, 9), (2, 10), and (2, 12) in the Markov model for the exact subsystem. If we assume that both memory units share the same failure rate, then net flow out of these states is defined by the sum $(\lambda_{M1} + \lambda_{P1})(P_{(2,3)} + P_{(2,9)} + P_{(2,10)} + P_{(2,12)})$. These two probability flows can be equated to derive an expression for $\lambda^e_1(2)$ in terms of parameters of the exact subsystem. We will denote the probability of being in state (i, j) in the Markov model for the approximate subsystem by $P_{(i,j)}$ and the probability of being in state (i, j) of the Markov model for the exact subsystem by $P'_{(i,j)}$. Under this convention, equation 4.7 yields the aforementioned expression, where $P_{(1,2)}$ has been replaced by $P'_{(1,4)} + P'_{(2,3)} + P'_{(2,9)} + P'_{(2,10)} + P'_{(2,12)}$ in equation 4.7b.

$$P_{(1,2)}\lambda_1^e(2) = \lambda_{P1}P'_{(1,4)} + (\lambda_{P1} + \lambda_{M1})(P'_{(2,3)} + P'_{(2,9)} + P'_{(2,10)} + P'_{(2,12)}) \tag{4.7a}$$

$$\lambda_1^e(2) = \frac{\lambda_{P1}P'_{(1,4)} + (\lambda_{P1} + \lambda_{M1})(P'_{(2,3)} + P'_{(2,9)} + P'_{(2,10)} + P'_{(2,12)})}{P'_{(1,4)} + P'_{(2,3)} + P'_{(2,9)} + P'_{(2,10)} + P'_{(2,12)}} \tag{4.7b}$$

Notice from the symmetry of the Markov model for the exact subsystem that $P'_{(2,3)} = P'_{(2,9)} = P'_{(2,10)} = P'_{(2,12)}$. Also it can be shown that for short times $P'_{(1,4)} \cong \lambda_{P2}t$ and $P'_{(2,10)} \cong \lambda_{P2}\lambda_{M1}\, t^2\!/_2$ [Babcock2]. By applying these approximations to equation 4.7b we get the short time approximation of equation 4.8.

$$\lambda_1^e(2) = \frac{\lambda_{P1} + (\lambda_{P1} + \lambda_{M1})2\lambda_{M1}t}{1 + 2\lambda_{M1}t} \tag{4.8}$$

Equation 4.8 verifies our intuition about the short time behavior of $\lambda_1^e(2)$ and, by symmetry, $\lambda_2^e(1)$. Notice however that if the common mode failure transition were not explicitly included in the Markov model for the approximate subsystem, then equation 4.7a would not hold and these effective transition rates would not exhibit the suggested behavior.

The common mode transition in the approximate subsystem corresponds to a total memory failure in the exact subsystem. This might lead us to believe that the common mode failure rate equals the failure rate for the memory unit. In order to see why this is not the case, we relate the effective transition rate for the common mode failure to values from the Markov model for the exact subsystem (figure 4.5).

State (0, 1) in the Markov model for the approximate subsystem (figure 4.6) corresponds to the sum of states (0, 1), (1, 1) and (1, 3) in the Markov model for the exact subsystem. Equation 4.9a results from equating the time derivative of the above state probabilities.

$$\begin{aligned}
\frac{dP_{(0,1)}}{dt} &= -\lambda_1^e()P_{(0,1)} - \lambda_2^e()P_{(0,1)} - \lambda_{Comm}^e P_{(0,1)} \\
&= -\lambda_1^e()[P'_{(0,1)} + P'_{(1,1)} + P'_{(1,3)}] - \lambda_2^e()[P'_{(0,1)} + P'_{(1,1)} + P'_{(1,3)}] \\
&\quad -\lambda_{Comm}^e[P'_{(0,1)} + P'_{(1,1)} + P'_{(1,3)}] \\
&= -\lambda_{P1}[P'_{(0,1)} + P'_{(1,1)} + P'_{(1,3)}] - \lambda_{P2}[P'_{(0,1)} + P'_{(1,1)} + P'_{(1,3)}] \\
&\quad -\lambda_{M1}P'_{(1,1)} - \lambda_{M2}P'_{(1,3)} \\
&= \frac{dP'_{(0,1)}}{dt} + \frac{dP'_{(1,1)}}{dt} + \frac{dP'_{(1,3)}}{dt}
\end{aligned} \tag{4.9a}$$

Probability flows in the exact subsystem model which correspond to the failure of effective processor 1 or effective processor 2 in the approximate subsystem can be identified. By equating these probability flows with the corresponding probability flows of the approximate subsystem in equation 4.9a, we obtain equation 4.9b.

$$(4.9a) \Rightarrow \begin{cases} \lambda_1^e() = \lambda_{P1} \\ \lambda_2^e() = \lambda_{P2} \end{cases} \qquad (4.9b)$$

This approach can be used to obtain an expression relating the effective common mode failure rate with values from the exact subsystem. If both memory units have the same failure rate, then 4.9c relates the common mode failure rate to the failure rate of the memory unit and a ratio of state probabilities.

$$\lambda_{M1} = \lambda_{M2} \Rightarrow \lambda_{comm}^e = \frac{P'_{(1,1)} + P'_{(1,3)}}{P'_{(0,1)} + P'_{(1,1)} + P'_{(1,3)}} \lambda_{M1} \qquad (4.9c)$$

Clearly, the resulting common mode failure rate will exhibit a dependence on time which is governed by this ratio of state probabilities. This behavior, as well as the behavior of the other effective failure rates, can be seen empirically.

The original test of §4.2.1 was run again using the new approximate subsystem model (figure 4.6) and the following results were obtained.

Time(hr's)	$\lambda_1^e()$	$\lambda_2^e()$	λ_{Comm}^e	$\lambda_1^e(2)$	$\lambda_2^e(1)$
0.0 - 100.0	1.000e-05	1.000e-05	9.888e-07	1.131e-05	1.131e-05
100.0 - 200.0	1.000e-05	1.000e-05	4.930e-06	1.303e-05	1.303e-05
200.0 - 300.0	1.000e-05	1.000e-05	4.816e-06	1.488e-05	1.488e-05
300.0 - 400.0	1.000e-05	1.000e-05	6.647e-06	1.669e-05	1.669e-05
400.0 - 500.0	1.000e-05	1.000e-05	8.427e-06	1.846e-05	1.846e-05
500.0 - 600.0	1.000e-05	1.000e-05	1.016e-05	2.018e-05	2.018e-05
600.0 - 700.0	1.000e-05	1.000e-05	1.184e-05	2.086e-05	2.086e-05
700.0 - 800.0	1.000e-05	1.000e-05	1.347e-05	2.349e-05	2.349e-05
800.0 - 900.0	1.000e-05	1.000e-05	1.507e-05	2.508e-05	2.508e-05
900.0 - 1000.0	1.000e-05	1.000e-05	1.662e-05	2.663e-05	2.663e-05

Table 4.3: Test Results - Common Mode Failure

Notice that the effective component failure rate at the first failure level is indeed independent of time. The common mode failure rate varies with time as expected. Also, the effective component failure rates at the second failure level are still time varying. In order to see how closely the actual behavior of these transition rates matches the expected behavior consider the following comparisons. Since for the given test case $\lambda_{P1} = \lambda_{P2} = 10^{-5}$, $\lambda_{M1} = \lambda_{M2} = 10^{-4}$ and t = 900 hours then by the approximate equation 4.8 we would expect $\lambda_2^e(1)$ to be approximately $2.52\text{x}10^{-5}$, and this value is about what we see in table 4.3 in the interval from 900 to 1000 hours. For t = 100 hours and the same processor and memory failure rates, the approximation of equation 4.8 says that $\lambda_2^e(1)$ should be about $1.2\text{x}10^{-5}$, very close to λ_{P1} , which is what we get in table 4.3 in the range from 100 to 200 hours.

The above example illustrates how different aspects of the exact subsystem architecture are captured by the effective failure rates of the approximate subsystem. Subsystem channel failure modes which depend on the status of the remainder of the subsystem are mimicked by state dependence. Failure modes which depend on the failure level of the Markov model for the exact subsystem are captured by time dependence. The following section further examines the relationship between the exact subsystem and the approximate subsystem.

4.3 Defining States for the Approximate Subsystem

4.3.1 Sequence Dependencies

In the previous sections we discussed how different aspects of an exact subsystem affect the effective component failure rates of the corresponding approximate subsystem. However, most of these aspects are captured naturally by using Markov models to perform the reliability analysis of the exact subsystem. For example, Markov models naturally capture sequence dependencies within the exact subsystem [Babcock1]. Consequently, in performing a given hierarchical analysis, the user generally need not consider the internal structure of the exact subsystem. As noted in chapter 3, a far more important consideration is the interaction of this exact subsystem with the remainder of the exact system.

In the hierarchical technique proposed here we are seeking to capture states of the exact subsystem which fully characterize its effect on the remainder of the exact system. While the components of the approximate subsystem are used to replace specific channels of the exact subsystem, the probability of being in specific subsystem states is captured by effective component failure rates. Clearly, the states needed to fully characterize the dependence of the exact system on the subsystem must appear in the Markov model of the approximate subsystem. Otherwise this Markov model would not produce the necessary effective failure rates to properly represent the exact subsystem's influence on the remainder of the exact system.

Notice that the Markov models of figures 4.3 and 4.6 are used in conjunction with the same exact subsystem. However, the two models capture different states of the exact subsystem. The approximate subsystem model of figure 4.3 captures all of the possible combinations of operational and inoperational subsystem channels. It also distinguishes between two sequences of channel failures. For example, state (2, 1) of this Markov model represents the failure of effective component 1 followed by a failure of effective component 2. The model of figure 4.6 on the other hand captures all of the information in the model of figure 4.3 and explicitly tracks the probability of having a simultaneous fault of both channels. Deciding which model is appropriate depends on the interaction we are trying to capture between the exact subsystem and the remainder of the exact system.

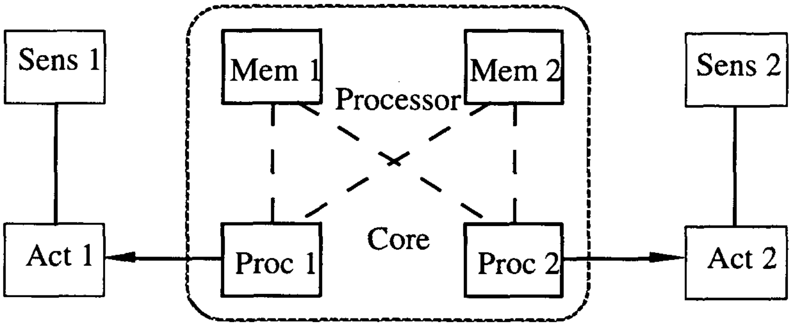


Figure 4.7: Example System

Consider the system of figure 4.7. Assume that the processor core for this system has the same connectivity described in §4.2.2. In order for the system to function properly there must be at least one functioning sensor/actuator pair. Each actuator receives data from its corresponding sensor and control signals from the processor core, as indicated. Initially the system uses the first sensor/actuator pair. If this pair becomes non-functional, then the system reconfigures to the active second sensor/actuator pair. Assume that if a sensor/actuator pair does not receive control input from the processor core that it may continue to operate in a degraded mode as long as the sensor and actuator are both unfailed. However, the system will always attempt to operate in the highest operational mode possible.

Consider the following possibilities. If the first processor channel becomes unavailable, the system immediately attempts to reconfigure to the second sensor/actuator pair. If this pair is unavailable and the first sensor/actuator pair is unfailed, the system continues to operate in the degraded mode mentioned earlier. If the processor core is not needed to perform the actual reconfiguration, the reliability of the system depends only on the availability of either processor channel. For this case, the approximate subsystem need only capture the probability of having both channels operating, channel 1 failed, channel 2 failed, or both channels failed. No information about the sequence of channel failures is required. For this case the Markov model of figure 4.3 is sufficient to describe the dependence of the sensor/actuator suite on the processor core.

Consider now the case where the system operates as before, except that now processor channel 2 is responsible for the reconfiguration from the first sensor/actuator pair to the second. Assume that if this processor channel is unavailable when a reconfiguration is attempted the system waits indefinitely and a system failure occurs. Here the reliability of the system depends on the order in which the two processor channels fail. For example, if processor channel 1 fails first, the system will reconfigure to the second sensor/actuator pair and continue to function even after processor channel 2 becomes unavailable. However, if processor channel 2 fails followed by a failure of processor channel 1, a system failure will occur even though one or the other sensor/actuator pair could continue to operate in a degraded mode.

When the reliability of the system depends on the order in which subsystem channels become unavailable, the approximate subsystem must correctly capture the sequence of effective component failures. Since the Markov model of figure 4.3 differentiates between different failure sequences we might be led to believe that we can use this model to capture failure sequence dependencies on the subsystem channels. However, notice that the processor core of figure 4.7 has a common mode failure which is not captured in the Markov model of figure 4.3, i.e., the failure of both memory units before either processor fails. The probability of this state is distributed between the two effective component failure sequences, and thus does not accurately reflect the operation of the system. For example, if the system is using the first sensor/actuator pair and a common processor channel failure occurs the system would try to reconfigure to the second sensor/actuator pair and fail. However, if the Markov model of figure 4.3 is used to calculate effective component failure rates, this situation would sometimes allow the system to continue to operate and sometimes not. We conclude that the effective component failure rates must be calculated with the Markov model of figure 4.6 in order to correctly capture the effect of the common mode failure.

It seems that order dependencies on effective component failures could be captured by explicitly including common mode failures in the approximate subsystem model. For the purposes of this thesis we simply note that order dependencies are difficult to handle unless they occur completely within the exact subsystem.

From this example we conclude the following.

1. The states of the approximate subsystem model must be generated to correctly capture the effect of the exact subsystem on the remainder of the system.

2. In order to correctly capture these states we may need to pay close attention to odd states of the exact subsystem. In particular, common mode failures require care in capturing the sequence dependencies of the exact subsystem in the states of the model of the approximate subsystem.

4.3.2 States with Zero Probability

In many situations, states of the approximate subsystem model have no probability because of the unreachability of certain states within the exact subsystem. Consider for example the system of figure 4.8. In this system the memory units represent a redundant pair but processor 1 must be unfailed in order for processor 2 to have access to either of them.

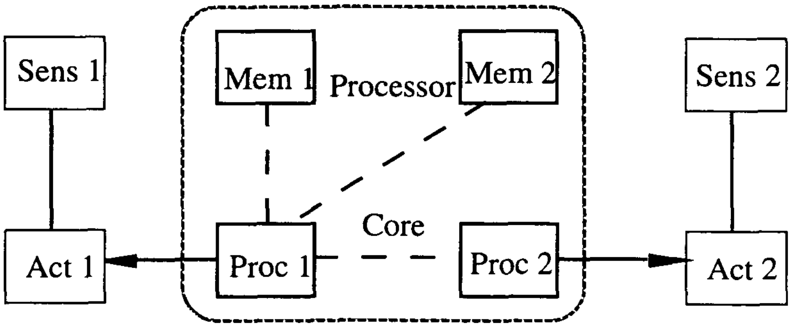


Figure 4.8: System with Unreachable State

We would like to replace the processor core of this system with two effective processors. Solving for the effective component failure rates of such an approximate subsystem by the Markov model of figure 4.3, results in equations 4.4. Also, the states of this model are defined in terms of which effective components are functional. Since in the processor core of figure 4.8 the loss of processor channel 1 causes the loss of processor channel 2, there will be no probability in state (1, 1) of the approximate subsystem model.

Numerically, this zero state probability produces a division by zero in equation 4.4b. Physically, this zero state probability corresponds to the unreachability of a specific state within the exact subsystem. How can this physical interpretation be reflected in the Markov model for the approximate subsystem?

One simple approach is to use the same Markov model but to redefine the states in terms of which effective components have *failed* as opposed to simply becoming *non-functional.* A *failure* of the first processor within the exact subsystem of figure 4.8 might correspond to a failure of effective processor 1, while effective processor 2 might be considered to be *unfailed* but *non-functional.* Thus, if the states of the Markov model in figure 4.3 are defined in terms of which effective components have failed, then state (1, 1) still contains some probability, but state (2, 1) has zero probability. Numerically, this results in a zero transition rate from state (1, 1) to state (2, 1), i.e., we have eliminated the division by zero error encountered previously. In general, this definition of the states only permits states with zero probability to exist 'downstream' from states with non-zero probability. This avoids division by zero and it sets all transitions into an unreachable state of the exact subsystem to zero.

The limitation of this technique is that for large exact subsystems, with complex interdependencies among subsystem channels, it becomes difficult to define when a channel has *failed* and when it has simply become *non-functional.* For example, in the system of figure 4.8, the consecutive failure of both memory units causes both processors to become *non-functional.* Do we interpret this as the *failure* of effective processor 1 or effective processor 2? If the states of the approximate subsystem model are defined as before, then the loss of this shared resource corresponds to a common mode transition to system loss. However, under this revised state definition we are trying to avoid common mode transitions in order to avoid having states with non-zero probability follow states with zero probability.

An alternative approach to reflecting the unreachability of certain subsystem states in the Markov model for the approximate subsystem is the following:

1. Build a Markov model for the approximate subsystem under the assumption that all of the states which need to be captured will have finite probability.

2. Determine the actual state probabilities using a Markov model for the exact subsystem. States within the approximate subsystem model with zero state probability correspond to unreachable subsystem states.

3. Eliminate all unreachable states from the approximate subsystem model. Appropriate transitions must be created between the remaining states in the approximate subsystem model.

4. Develop state equations for the resulting approximate subsystem model and use this system of equations to determine numerical values for the remaining effective transition rates.

In terms of the processor core of figure 4.8, the previous multi-step procedure would call for the elimination of state (1, 1). The transition emanating from state (0, 1) due to the loss of effective processor 1 would go directly to state (2, 1). This is demonstrated in the revised Markov model of figure 4.9.

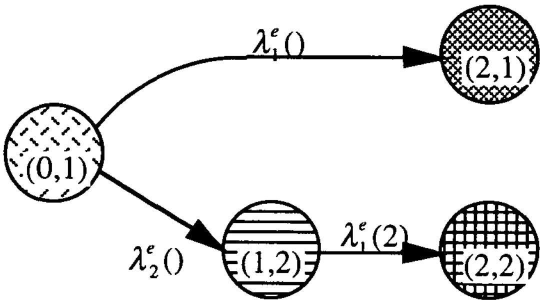


Figure 4.9: Revised Approximate Subsystem Model

The transition rates present in the Markov model of figure 4.9 can be calculated by deriving a closed form solution from the state equations, as in §4.1. Such a closed form solution can be found in equations 4.10.

$$\lambda_1^e() = \frac{\Delta P_{(2,1)}}{T_{avg} P_{(0,1)}(nT_{avg})} \tag{4.10a}$$

$$\lambda_1^e(2) = \frac{\Delta P_{(2,2)}}{T_{avg} P_{(1,2)}(nT_{avg})} \tag{4.10b}$$

$$\lambda_2^e() = \frac{\Delta P_{(1,2)}}{T_{avg} P_{(0,1)}(nT_{avg})} + \frac{\lambda_1^e(2) P_{(1,2)}(nT_{avg})}{P_{(0,1)}(nT_{avg})} \tag{4.10c}$$

Thus, a closed form solution for the effective component failure rates is not derived until an approximate subsystem model has been derived which only contains states with finite probability. Once such a Markov model has been generated, the remainder of the analysis proceeds as usual.

4.3.3 State Space Reduction Techniques

From the discussion of §4.1 it can be seen that the number of effective transition rates which must be solved for grows rapidly with the number of effective components within the approximate subsystem. For example, consider a subsystem which contains four channels to be modeled by four effective components. If the effective components need only capture the probability of having any combination of failed and unfailed effective components available without regard to the failure sequences, then the resulting effective transition rates will simply depend on which effective components have already failed. This results in $4 \cdot 2^3$ or thirty two unknown effective transition rates.

Clearly, even for approximate subsystems of only moderate size the number of parameters (i.e., effective transition rates) to be solved for becomes unreasonable. In order to mitigate this problem we need to simplify the effective failure rate, or develop a simple approximation to the effective failure rate.

State Aggregation

For many systems the order of failures which cause a subsystem to fail completely has no bearing on system reliability. For such systems, we would like to reduce the size of the approximate subsystem model by aggregating all of the system loss states. However, it appears that this would violate the assumption that only one unknown effective transition flow into any given state of the approximate subsystem model of §4.1. In order to get around this problem we assume that all transitions leading to a system loss state occur at the same rate. Although this may not accurately model the failure rate for a particular effective component, it will accurately model the net flow of probability into system loss. This assumption allows us to use the same approach for calculating effective transition rates. In fact it is this assumption which reduces the number of unknown parameters to be calculated.

Consider the Markov model of figure 4.3. We can simplify this model by aggregating states (2, 1) and (2, 2) as in figure 4.10. The transition from states (1, 1) and (1, 2) are both labeled with the same failure rate $\lambda^e(1\ Comp)$. The solution to the set of unknown effective transition rates is in equations 4.11. Notice that in this system we now have only three unknown transition rates to solve for as opposed to four. Also λ^e*(1 Comp)* is used to replace the transition rates which were a function of state, i.e., which effective components had already failed.

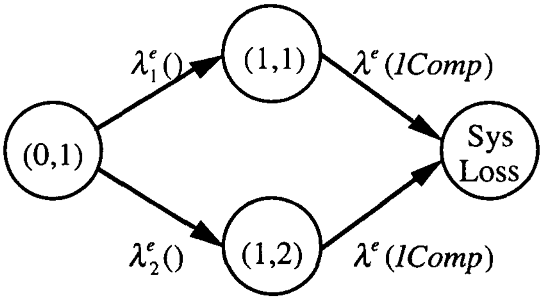


Figure 4.10: Approximate Subsystem Model with State Aggregation

$$\lambda^e(1Comp) = \frac{\Delta P_{Sys-Loss}}{T_{avg}[P_{(1,1)}(nT_{avg}) + P_{(1,2)}(nT_{avg})]} \tag{4.11a}$$

$$\lambda_1^e() = \frac{\Delta P_{(1,1)}}{T_{avg}P_{(0,1)}(nT_{avg})} + \frac{\lambda^e(1Comp)P_{(1,1)}(nT_{avg})}{P_{(0,1)}(nT_{avg})} \tag{4.11b}$$

$$\lambda_2^e() = \frac{\Delta P_{(1,2)}}{T_{avg}P_{(0,1)}(nT_{avg})} + \frac{\lambda^e(1Comp)P_{(1,2)}(nT_{avg})}{P_{(0,1)}(nT_{avg})} \tag{4.11c}$$

An additional benefit to state space reduction of the approximate subsystem model is that it permits additional state aggregation of the exact subsystem model. For example, consider the Markov model of figure 4.5. Since we need to capture specific state probabilities within the approximate subsystem model, only the states which have the same shading can be considered for aggregation. Although all of the states in the final failure level of the exact subsystem model satisfy the rules for state aggregation of §2.2.1, they cannot all be aggregated if the final failure level states of the approximate subsystem model have not already been aggregated.

Model Truncation

Most systems of interest will exhibit progressively degraded reliability as more subsystem channels become unavailable. Therefore we may assume that a given subsystem has failed entirely at a certain failure level without increasing our estimate of system reliability. This intuition is captured in the following model truncation technique.

Consider an approximate subsystem composed of three effective components. Such a subsystem can be described by the Markov model of figure 4.1. What happens if we assume that the loss of any two effective components will cause the subsystem to fail? This corresponds to eliminating all of the states at failure level three. If we also aggregate all of the states at the second failure level, this results in the Markov model of figure 4.11. Notice that this Markov model contains substantially less unknown transition rates than the Markov model of figure 4.1.

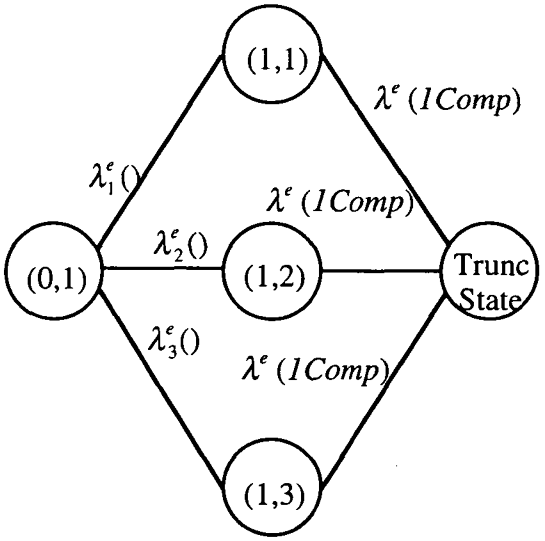


Figure 4.11: Approximate Subsystem Model with Truncation

In theory the Markov model of figure 4.1 could capture the essential characteristics of the corresponding exact subsystem perfectly. However, the truncated model can at best provide an approximation to these characteristics. The quality of this approximation depends on the difference between the probability in the truncation state and the true probability of system loss. Once the probability in the truncation state is known, the systems analyst can decide whether or not to expand the subsystem model.

When the probability of the truncation state is viewed as a system loss probability, the resulting effective failure rates provide a pessimistic estimate of subsystem reliability. If we add the probability of being in the truncation state to the first state, then the resulting set of state probabilities produces an optimistic estimate of subsystem reliability. However, this estimate seems unrealistic since it yields no probability of losing the entire subsystem. We conclude that model truncation yields a simple over bound to subsystem unreliability, and consequently system unreliability. However, a good corresponding lower bound on subsystem unreliability is not obvious.

4.4 Conclusion

The relationship between the approximate subsystem and its corresponding exact subsystem reflects in many ways the relationship between the exact subsystem and the remainder of the exact system. Certain aspects of the subsystem behavior have a direct impact on system reliability and these must be captured in the Markov model for the approximate subsystem. Other aspects of the exact subsystem have no direct impact on the remainder of the exact subsystem, and these aspects are reflected indirectly in the values of the resulting effective failure rates.

This chapter explores the relationship between the approximate subsystem and its corresponding exact subsystem, developing some intuition about this relationship. From the general system of figure 4.2 we saw the following:

1. Resource sharing between subsystem channels provides the basis for dependence between effective components in the approximate subsystem.

2. Dependence between subsystem channels is captured in the effective component failure rates by state dependence. Also, failure modes which depend on the failure level of the Markov model for the exact subsystem are captured in the effective component failure rates by time dependence.

These two conclusions provide understanding of the nature of what we call an effective failure rate. Specifically, the conclusions of §4.3.1 can guide the system's analyst in developing appropriate approximate subsystem models:

1. The states of the approximate subsystem model must be generated to correctly capture the effect of the exact subsystem on the remainder of the system.

2. In order to correctly capture these states we may need to pay close attention to odd states of the exact subsystem. In particular, common mode failures require care in capturing the sequence dependencies of the exact subsystem in the states of the model of the approximate subsystem.

In §4.3.2 we note the potential problems presented by the unreachability of certain states within the exact subsystem, and one approach to dealing with this problem. Finally, we note that state space reduction techniques can be used in conjunction with the approximate subsystem model in order to simplify the hierarchical analysis.

Chapter 5

Numerical Issues

The previous chapter discussed one method for determining effective component failure rates, and highlighted some of the characteristics of this parameter. However, one of the greatest difficulties which arises in the actual implementation of this modeling technique is finding accurate values for the necessary effective component failure rates. Because the solution technique proposed in chapter 4 is based on the difference in state probabilities, the resulting effective failure rates are highly sensitive to roundoff error. Here, we look more closely at the numerical issues involved in determining effective component failure rates. We pay close attention to the effect of T_{avg} on the resulting effective failure rates.

5.1 Effects of Discretization

In the previous chapter we derived closed form solutions for effective component failure rates from the Markov model associated with the approximate subsystem. The system of linear differential equations associated with this Markov model was used to develop finite difference equations which in turn were used to solve for the effective failure rates. In order to develop a set of finite difference equations Euler's method of integration was applied.

Equation 5.1 is a matrix representation of the state equations for an arbitrary Markov model. A distinguishing characteristic of determining effective component failure rates is that we are interested in finding elements of the transition matrix, $[A]$ given values of the probability vector, $\vec{p}$. This means that the second form of equation 5.1 must be used to obtain a solution for the vector of transition rates, $\vec{a}$.

$$\frac{d\vec{p}}{dt} = [A]\vec{p} \Leftrightarrow [P]\vec{a} = \frac{d\vec{p}}{dt} \tag{5.1}$$

Equation 5.2 is produced by applying Euler's method of integration to the second form of equation 5.1. From linear algebra, we know that if the rank of the matrix $[P]$ is

$$[P]\vec{a} = \frac{\Delta\vec{p}}{T_{avg}} \tag{5.2}$$

equal to the length of the vector $\vec{a}$ the elements of this vector are uniquely determined by equation 5.2 [Strang]. If the Markov model for the approximate subsystem is generated as suggested in §4.1 the resulting system of effective failure rates will always be uniquely determined.

The solution of equation 5.2 necessarily involves a difference in state probabilities. In contrast, when equation 5.1 is used to solve for state probabilities, the probabilities are stepped forward in time and no such difference is used. Notice that subtraction is an operation which is particularly susceptible to roundoff error. Whereas roundoff error is a secondary concern in the forward integration of equation 5.1, we expect roundoff error to play a relatively significant role in solving equation 5.2 for the effective transition rates. This issue is incorporated in the following observations.

1. As T_{avg} decreases, the differences in state probabilities, $\Delta\vec{p}$, become small. Since these state probabilities are calculated using finite precision, as T_{avg} shrinks the differences in these state probabilities may introduce significant roundoff error.

2. As T_{avg} increases, the discretization of equation 5.2 becomes a poor approximation to the corresponding continuous differential equation. Such error is referred to as integration error.

5.2 Choosing an Appropriate T$_{avg}$

Clearly, our choice of T_{avg} involves a tradeoff between integration and roundoff error; large values for T_{avg} produce large integration error, while small values of T_{avg} produce very small $\Delta\vec{p}$ and large roundoff error. This may be conveyed by the following example.

Consider the set of components in figure 5.1. Solar arrays 1 through 4 provide energy to the four power supplies. The starboard radiator (Rad S) regulates the temperature for arrays 1 and 3. Similarly, the port radiator (Rad P) regulates the temperature for arrays 2 and 4. The power supplies 1 and 4 are cross-strapped to allow them to share solar arrays. Thus, in order for power supply 1 to function, solar array 1 must be functional or power supply 4 must be functional. Power supplies 2 and 3 are similarly cross-strapped. These dependencies are summarized graphically in figure 5.1.

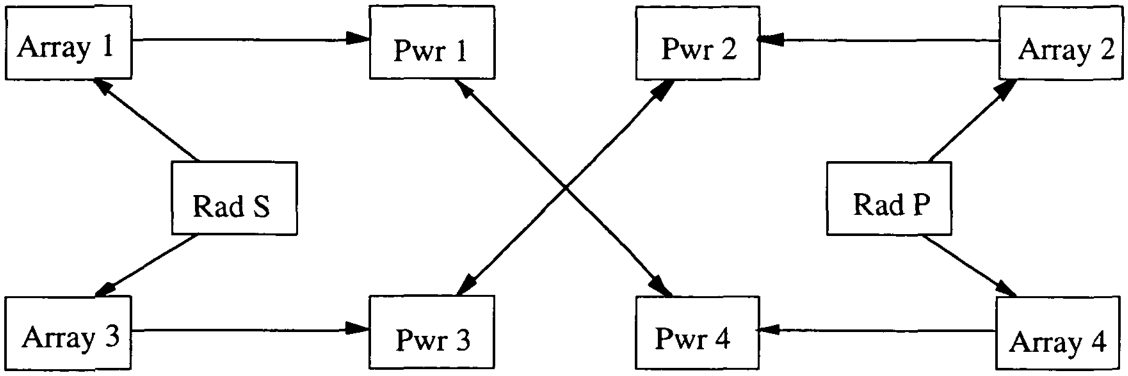


Figure 5.1: Exact Power Subsystem

Let the components in figure 5.1 represent a subsystem which we would like to replace, within some large system, by the four effective components in figure 5.2. Because of the symmetry of this particular subsystem, we can assume that all four effective components in the approximate subsystem will share the same failure rate. Assume also that the reliability of the resulting approximate system depends only on which effective power supplies have failed, but not on the order in which they fail.

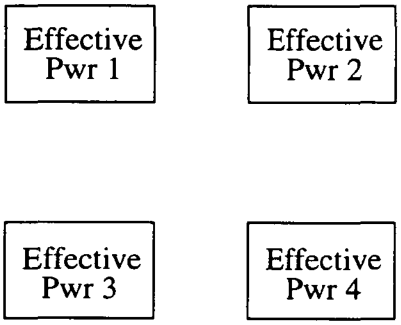


Figure 5.2: Approximate Power Subsystem

Under the stated assumptions, the 'chain' model in figure 5.3 can accurately be used to determine the correct state dependent failure rate for each of the effective power units in figure 5.2. The 'chain' model has at most one entering and one exit transition for each state. Because of this special structure, closed form solutions for the effective component failure rates can be derived by starting from either the transition rate leaving state (0, 1) or the transition rate entering state (4, 1). For this particular example, we will start with the transition rate leaving state (0, 1).

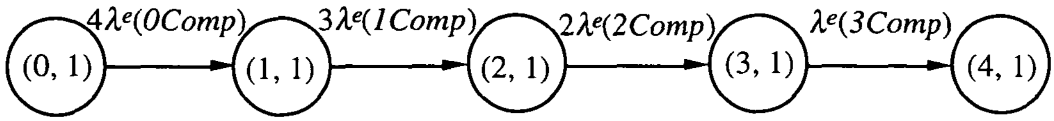


Figure 5.3: Markov Model for Approximate Power Subsystem

In the Markov model of figure 5.3, the transition rates $\lambda^e(0Comp)$, $\lambda^e(1Comp)$, $\lambda^e(2Comp)$ and $\lambda^e(3Comp)$ all make up the shared effective failure rate for the approximate subsystem of figure 5.2. This failure rate is a function of how many other effective components have failed as opposed to which effective components have failed. For this subsystem, $\lambda_1^e(2) = \lambda_1^e(3) = \lambda_1^e(4) = \lambda^e(1Comp)$. In order to compute this effective failure rate, we develop the following system of equations which describe the Markov model of figure 5.3.

$$\frac{dP_{(0,1)}}{dt} = -4\lambda^e(0Comp)P_{(0,1)}(t) \tag{5.3a}$$

$$\frac{dP_{(1,1)}}{dt} = 4\lambda^e(0Comp)P_{(0,1)}(t) - 3\lambda^e(1Comp)P_{(1,1)}(t) \tag{5.3b}$$

$$\frac{dP_{(2,1)}}{dt} = 3\lambda^e(1Comp)P_{(1,1)}(t) - 2\lambda^e(2Comp)P_{(2,1)}(t) \tag{5.3c}$$

$$\frac{dP_{(3,1)}}{dt} = 2\lambda^e(2Comp)P_{(2,1)}(t) - \lambda^e(3Comp)P_{(3,1)}(t) \tag{5.3d}$$

$$\frac{dP_{(4,1)}}{dt} = \lambda^e(3Comp)P_{(3,1)}(t) \tag{5.3e}$$

The Markov model in figure 5.3 produces only four independent state equations. In particular, equation 5.3e above results from the negative of the sum of equations 5.3a through 5.3d. Thus, the Markov model is fully described by equations 5.3a through 5.3d. Since there are four unknown effective transition rates, $\lambda^e(0Comp)$, $\lambda^e(1Comp)$, $\lambda^e(2Comp)$, $\lambda^e(3Comp)$ and four equations, the effective failure rate is uniquely determined by the equations 5.3a through 5.3d.

In order to solve the system of first order differential equations, 5.3, we need to integrate out the time derivatives on the left hand side. Application of Euler's method results in the system of difference equations, 5.4. Here we have excluded the discretized version of equation 5.3e, since it is not used in solving for the effective failure rates.

$$\frac{\Delta P_{(0,1)}}{T_{avg}} = -4\lambda^e(0Comp)P_{(0,1)}(nT_{avg}) \tag{5.4a}$$

$$\frac{\Delta P_{(1,1)}}{T_{avg}} = 4\lambda^e(0Comp)P_{(0,1)}(nT_{avg}) - 3\lambda^e(1Comp)P_{(1,1)}(nT_{avg}) \tag{5.4b}$$

$$\frac{\Delta P_{(2,1)}}{T_{avg}} = 3\lambda^e(1Comp)P_{(1,1)}(nT_{avg}) - 2\lambda^e(2Comp)P_{(2,1)}(nT_{avg}) \tag{5.4c}$$

$$\frac{\Delta P_{(3,1)}}{T_{avg}} = 2\lambda^e(2Comp)P_{(2,1)}(nT_{avg}) - \lambda^e(3Comp)P_{(3,1)}(nT_{avg}) \tag{5.4d}$$

The state probabilities on the right hand side of equations 5.4 are evaluated at the beginning of each time averaging interval. It would be equally valid to use the state probabilities at the end of the averaging interval, or some other point between the beginning and the end. For example, $P_{(0,1)}$ could be evaluated at nT_{avg}, $(n+1)T_{avg}$ or some point between the two. Also, it is unlikely that this choice will affect roundoff error, but it may affect integration error. When Euler's method is used to integrate forward in time, the state probabilities are typically taken at the beginning of each averaging interval. Thus, we assume for the time being that this is also the case in our solution of equations 5.4.

Recall from §4.1 that, in general, effective transition rates can most easily be solved for by starting from the system loss states and working backwards. Due to the particularly simple structure of the chain model, it is just as easy to begin solving for the effective transition rates by starting at the fully operational state, (0, 1). Equation 5.5 can be derived in such a manner.

$$4\lambda^e(0Comp) = -[\frac{\Delta P_{(0,1)}}{P_{(0,1)}T_{avg}}] \tag{5.5a}$$

$$3\lambda^e(1Comp) = -[\frac{\Delta P_{(1,1)} + \Delta P_{(0,1)}}{P_{(1,1)}T_{avg}}] \tag{5.5b}$$

$$2\lambda^e(2Comp) = -[\frac{\Delta P_{(2,1)} + \Delta P_{(1,1)} + \Delta P_{(0,1)}}{P_{(2,1)}T_{avg}}] \tag{5.5c}$$

$$\lambda^e(3Comp) = -[\frac{\Delta P_{(3,1)} + \Delta P_{(2,1)} + \Delta P_{(1,1)} + \Delta P_{(0,1)}}{P_{(3,1)}T_{avg}}] \tag{5.5d}$$

In order to obtain a numerical solution for the effective failure rates in a given decomposition, we must choose an appropriate value for T_{avg}. Our choice of T_{avg} may vary with time or it might be fixed for all times of interest. It is possible that the roundoff error may be more significant than integration error during some portions of the evaluation and less significant during other portions of the evaluation. Consequently, the above tradeoff would call for a different T_{avg} for different periods in the evaluation. For simplicity, we assume that T_{avg} will be fixed.

5.2.1 Sensitivity to Roundoff Error

For small times, the change in state probability, $\Delta P_{(0,i)}$ may be several orders of magnitude smaller than the probability of being in state *(0, i)*. The difference calculation to derive $\Delta P_{(0,i)}$ will result in a significant loss of accuracy for this case.

Suppose the full subsystem is evaluated using 4 digits of machine precision. If $P_{(0,1)}(t) = 9.004\text{x}10^{-1}$ and $P_{(0,1)}(t+T_{avg}) = 9.000\text{x}10^{-1}$ then $\Delta P_{(0,1)} = 4\text{x}10^{-4}$, but has only one digit of precision. The same loss of precision is also possible at any other state.

In order to mitigate the problem of roundoff error, we either need to increase the precision of our data, or increase T_{avg} until an acceptable level of roundoff error is achieved. In the above example this would mean either increasing the machine precision a few more digits, or increasing T_{avg} until all $\Delta P_{(0,i)}$ are greater than 10^{-3}. Since the precision of our data is always limited, unreasonably small values for T_{avg} should be avoided.

5.2.2 Sensitivity to Integration Error

All numerical integration techniques are by nature an approximation of the definite integral. Therefore, any numerical integration technique that is applied will result in some level of error based on whatever simplifying assumptions are used to perform the integration. For example, by applying Euler's method to equations 5.3, we have assumed that the state probabilities, $P_{(0,i)}$, and effective transition rates, $\lambda^e(i\ Comp)$ are constant over each averaging interval. The three equations below indicate how equation 5.3a was integrated using Euler's method.

$$\int_{t}^{t+T_{avg}} \frac{dP_{(0,1)}}{dt}dt = -\int_{t}^{t+T_{avg}} 4\lambda^e(0Comp)P_{(0,1)}dt$$

$$\Rightarrow \int_{P_{(0,1)}(t)}^{P_{(0,1)}(t+T_{avg})} dP_{(0,1)} = -4\lambda^e(0Comp)P_{(0,1)}(t)T_{avg}$$

$$\Rightarrow P_{(0,1)}(t+T_{avg}) - P_{(0,1)}(t) = -4\lambda^e(0Comp)P_{(0,1)}(t)T_{avg}$$

$$\Rightarrow \frac{\Delta P_{(0,1)}}{T_{avg}} = -4\lambda^e(0Comp)P_{(0,1)}(t)$$

In the second step indicated above, we have assumed a constant value for $\lambda^e(0Comp)$ and we have assumed that $P_{(0,1)} = P_{(0,1)}(t)$ across the interval from t to $t + T_{avg}$. Figure 5.4 gives a graphical interpretation of how Euler's method is used in numerical integration.

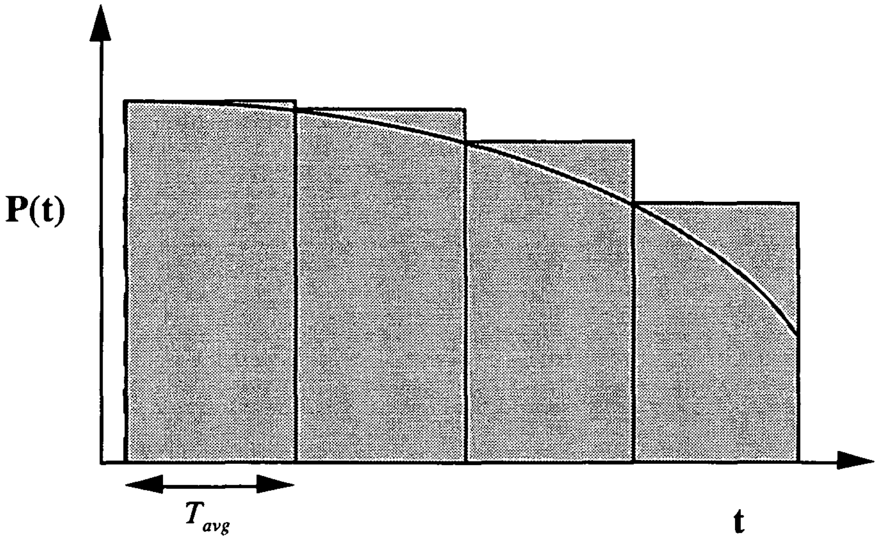


Figure 5.4: Euler's Method Applied to State Equations

By assuming constant state probabilities, and constant effective transition rates across an averaging interval, two distinct sources of integration error have been introduced. From the discussion of §5.1 and the graph in figure 5.4, we conclude that integration error will be small if the state probabilities and effective transition rates change very little across an averaging interval. This will only be the case for small enough T_{avg}.

5.3 A Numerical Example

5.3.1 Roundoff Error

A Markov model was built for the exact power subsystem shown in figure 5.1 using an automated Markov model construction tool. This tool is briefly presented in chapter 6 (for a more detailed discussion of this tool see Hutchins, Babcock, and Rosch). The failure rates for all components in this subsystem were set to 10^{-4} failures per hour. The resulting Markov model was evaluated using an integration time step, Δt of one hour and single precision arithmetic (i.e., eight digits of precision). State probabilities within the exact subsystem model were mapped to states within the approximate power subsystem model (figure 5.3). The length of the averaging interval, T_{avg} was also set to one hour.

A partial listing of the results taken at one hour time intervals is given in table 5.1. Notice that it takes several integration steps for any probability to flow into higher states, i.e., states (2, 1) and (3, 1). Because table 5.1 contains results from each time step in the original integration the first few entries for $P_{(2,1)}$ and $P_{(3,1)}$ are zero.

Time(hours)	$P_{(0,1)}(time)$	$P_{(1,1)}(time)$	$P_{(2,1)}(time)$	$P_{(3,1)}(time)$
0.0	1.0000e00	0.0000e00	0.0000e00	0.0000e00
1.0	9.9960e-01	5.0000e-04	0.0000e00	0.0000e00
2.0	9.9920e-01	7.9956e-04	5.0000e-07	0.0000e00
3.0	9.9880e-01	1.1987e-03	1.1993e-06	1.4400e-10
5.0	9.9840e-01	1.5974e-03	2.3972e-06	5.7565e-10
5.0	9.9800e-01	1.9956e-03	3.9929e-06	1.4383e-09

Table 5.1: Evaluation Results for Exact Subsystem (taken at 1 hour time intervals)

Since results have been taken from each integration time step, the above data gives us a lower bound on all possible T_{avg} for this evaluation. Values for the effective failure rate can be calculated over each averaging interval by substituting the above state probabilities into equations 5.5. Some of the results of this calculation are given in table 5.2.

T_{avg} (hrs)	$4\lambda^e(0Comp)$	$3\lambda^e(1Comp)$	$2\lambda^e(2Comp)$	$\lambda^e(3Comp)$
0.0 - 1.0	5.0000e-04	0.0000e00	0.0000e00	0.0000e00
1.0 - 2.0	5.0011e-04	9.7500e-04	0.0000e00	0.0000e00
2.0 - 3.0	5.0021e-04	9.6253e-04	-7.4230e-02	0.0000e00
3.0 - 4.0	5.0044e-04	1.0670e-03	6.7643e-02	5.6036e02
4.0 - 5.0	5.0058e-04	1.0631e-03	5.2694e-02	1.7629e02

Table 5.2: Effective Failure Rate (T_{avg} = 1.0 hour)

Initial inspection of the results in table 5.2 may lead the reader to believe that some of the entries have been calculated incorrectly. In particular, the negative failure rate shown for $\lambda^e(2Comp)$ has no physical meaning. However, close examination of the original data in table 5.1 reveals that the calculations which produce the results in table 5.2 are indeed correct.

The change in state probability $\Delta P_{(0,1)}$ from 2 to 3 hours is -3.9989x10^{-4}, and that $\Delta P_{(1,1)}$ across the same time interval is 3.9912x10^{-4}. Since $P_{(0,1)}$ is only accurate to the seventh decimal place, the sum, $\Delta P_{(0,1)} + \Delta P_{(1,1)} = -7.6960\text{x}10^{-7}$ has only one significant digit. When we add this to the value of $\Delta P_{(2,1)}$ (i.e., 7.9929x10^{-7}) the sum becomes -2.9692x10^{-8}, but this value has no digits of precision. In this case, finite precision arithmetic has resulted in *catastrophic cancellation* (i.e., a result which has no meaning because it has no digits of precision).

From the previous discussion we might be led to believe that roundoff error would be easy to detect because the resulting effective failure rate would be pure noise; however, the problem is more subtle than this. Notice that $\Delta P_{(0,1)}$ is on the order of 10^{-4} for all times with only 4 digits of precision. Since $\Delta P_{(0,1)}$ is the dominant term in equations 5.5, each component of the effective failure rate will have at most 4 digits of precision. Since $\Delta P_{(1,1)}$ is comparable in magnitude to $\Delta P_{(0,1)}$ but of opposite sign, there exists the potential for losing even more precision.

The effects of roundoff error can be explored by taking a detailed look at the roundoff error present in this particular case. We begin by estimating the number of digits of precision in each component of the effective failure rate. Assume that the only loss of precision occurs during the summation involved in equations 5.5. The number of digits of precision for a given summation is approximately the number of digits of precision of the largest term in that summation minus the loss in magnitude of the resulting summation. For example, assume that $P_{(0,1)}$ has 8 digits of precision and is on the order of 10^0. If $\Delta P_{(0,1)}$ is on the order of 10^{-3}, then this difference has about 5 digits of precision.

The indicated summations are calculated below (table 5.3) for each of the time intervals specified. This information is then used, as discussed above, to determine the approximate accuracy of the effective failure rate for each one of these time intervals.

Table 5.3 shows how the resulting effective failure rate from this set of data loses a substantial amount of accuracy due to roundoff error. We have not set a minimum on the number of significant digits in our effective failure rate. However, we presume that the level of error in the effective failure rate for this example would be intolerable for most applications.

	Summation			
	$\Delta P_{(0,1)}$	$\Delta P_{(0,1)} + \Delta P_{(1,1)}$	$\Delta P_{(0,1)} + \Delta P_{(1,1)} + \Delta P_{(2,1)}$	$\Delta P_{(0,1)} + \Delta P_{(1,1)} + \Delta P_{(2,1)} + \Delta P_{(3,1)}$
0.0 - 1.0	-5.0000e-04	5.4073e-17	5.4073e-17	5.4073e-17
1.0 - 2.0	-3.9995e-04	-3.9000e-07	9.9999e-09	9.9999e-09
2.0 - 3.0	-3.9996e-04	-7.6960e-07	2.9692e-08	2.9836e-08
3.0 - 4.0	-3.9996e-04	-1.2790e-06	-8.1123e-08	-8.0692e-08
4.0 - 5.0	-3.9994e-04	-1.6981e-06	-1.0235e-07	-1.0148e-07
	Digits of Precision			
	$\lambda^e(0Comp)$	$\lambda^e(1Comp)$	$\lambda^e(2Comp)$	$\lambda^e(3Comp)$
0.0 - 1.0	4			
1.0 - 2.0	4	1	0	0
2.0 - 3.0	4	1	0	0
3.0 - 4.0	4	2	1	1
4.0 - 5.0	4	2	1	1

Table 5.3: Approximate Accuracy of λ^e
(taken at 1 hour time intervals)

The level of accuracy in the first averaging interval cannot be determined by the same method applied to subsequent intervals. Because we have used the state probabilities, $P_{(0,1)}$ through $P_{(3,1)}$, evaluated at the beginning of each averaging interval, calculation of the effective failure rate leads to division by 0 at the first averaging interval. It is unclear what the effective failure rate should be for averaging intervals in which the initial state probability is zero (e.g., $\lambda^e(0Comp)$ from 0 to 1 hour). This topic will be examined further in §5.4.

This example highlights two aspects of roundoff error involved in determining the effective failure rates for an approximate subsystem:

1. Taking the difference in probabilities across averaging intervals leads to a significant though not necessarily catastrophic loss of accuracy.

2. Determination of an effective failure rate involves the summation of several terms which may be of opposite sign but comparable magnitude. This summation may lead to catastrophic cancellations which result in a numerically unstable solution.

The first issue is simple enough to detect by examining the data from an evaluation of the exact subsystem; the difference in state probability across an averaging interval can be compared to the state probability at a given time to determine its relative accuracy. The second problem is easy enough to detect in the small example discussed above, but may not be so clear in general. Still, for this example, an averaging interval of 1 hour is clearly too small.

By increasing T_{avg} we hope to increase the change in state probabilities used to calculate the effective failure rates and thereby reduce the amount of roundoff error mentioned in item 1. We also assume for large enough T_{avg} that the catastrophic cancellations discussed in item 2 will not take place.

Driven by the assumption that a larger averaging interval will reduce roundoff error, we examine the results of our evaluation of the exact subsystem in figure 5.1 at 50 hour time intervals (i.e., every fiftieth step of Δt = 1 hour). A partial listing of this data is displayed in table 5.4.

Time (hrs)	$P_{(0,1)}(time)$	$P_{(1,1)}(time)$	$P_{(2,1)}(time)$	$P_{(3,1)}(time)$
0.0	1.000000e00	0.000000e00	0.000000e00	0.000000e00
50.0	9.800286e-01	1.946484e-02	5.783169e-04	2.780214e-06
100.0	9.601314e-01	3.786809e-02	1.872372e-03	2.210736e-05
150.0	9.403313e-01	5.524800e-02	5.101050e-03	7.298605e-05
200.0	9.206495e-01	7.164195e-02	7.087962e-03	1.685688e-04
250.0	9.011052e-01	8.708646e-02	1.076122e-02	3.202990e-04
300.0	8.817162e-01	1.016171e-01	1.505324e-02	5.380446e-04
350.0	8.624988e-01	1.152688e-01	1.990048e-02	8.302236e-04
400.0	8.434676e-01	1.280752e-01	2.524331e-02	1.203921e-03
450.0	8.246362e-01	1.400695e-01	3.102578e-02	1.664999e-03
500.0	8.060167e-01	1.512836e-01	3.719546e-02	2.218200e-03

Table 5.4: Evaluation Results for Exact Subsystem (taken at 50 hour time intervals)

The number of digits of precision available after roundoff error can be estimated for each effective transition rate as was done for table 5.3. The results of this estimation are listed in table 5.5.

Digits of Precision				
Time (hrs)	*$4\lambda^e$(0Comp)*	*$3\lambda^e$(1Comp)*	*$2\lambda^e$(2Comp)*	*λ^e(3Comp)*
0 - 50	6	4	3	3
50 - 100	6	5	3	3
100 - 150	6	5	4	4
150 - 200	6	5	4	4
200 - 250	6	5	4	4
250 - 300	6	5	4	4
300 - 350	6	5	4	4
350 - 400	6	5	4	4
400 - 450	6	5	5	4
450 - 500	6	5	5	4

Table 5.5: Approximate Accuracy of λ^e (T_{avg} = 50.0 hour)

For this example a considerable increase in the averaging interval has reduced roundoff error substantially. We presume that the level of error in the effective failure rate for this example would be tolerable for many applications. However, we still have no estimate of the integration error for this example.

5.3.2 Integration Error

As we continue to increase T_{avg}, we expect roundoff error in λ^e to decrease. Integration error, which we assumed to be negligible at first, however, will increase. We assume roundoff error to be monotonically decreasing and integration error to be monotonically increasing. If this is the case, then for some value of T_{avg}, roundoff will cease to be the dominant form of error and integration error will take over. Given the above assumption of monotonicity, this will take place at the point where total error is at a minimum. The graph of figure 5.5 illustrates this behavior.

The important point is that the total error has some global minimum, and that this point can be determined by varying T_{avg}. Also, for the range of T_{avg} for which total error increases, integration error must be greater than roundoff error. Therefore, we would like to estimate the total error of the effective failure rate for this choice of averaging interval.

In order to estimate the total error, we will have to compare an evaluation of the approximate system to an evaluation of the corresponding exact system. The difference in state probabilities between these two systems would be the total error associated with the given hierarchical analysis. Assume that the effective components of the approximate subsystem accurately capture the relationship between the subsystem and the remainder of the exact system. If this assumption holds, then the total error for the given analysis can be completely attributed to the effective component failure rates. In practice, once we have an evaluation for the exact system there is no longer any need to perform the hierarchical modeling, so this ability to measure total error is not usually available to the system's analyst.

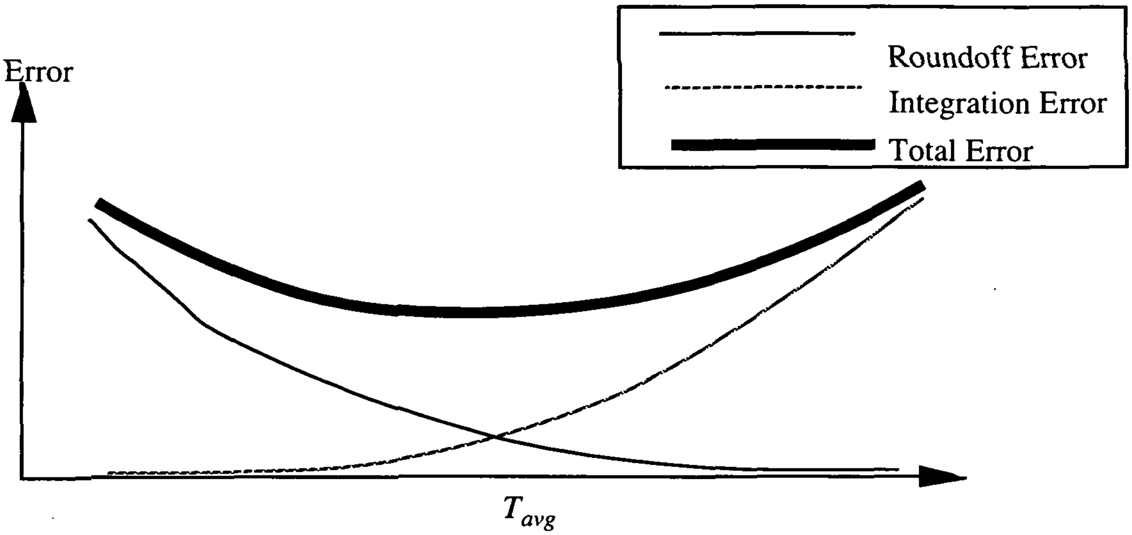


Figure 5.5: Nature of Total Error

A more useful measure of the total error associated with the effective failure rates can be obtained by using the resulting effective failure to determine state probabilities for the approximate subsystem model. The piece-wise approximation to the effective component failure rates can be used in a Markov model evaluator. The difference between the state probabilities as determined from the exact subsystem model and the state probabilities as determined from the approximate subsystem model can be used as an estimate of the total error associated with the effective failure rates. This measure of error is demonstrated in the following numerical example.

The state probabilities listed in table 5.4 can be used to calculate an effective failure rate for the approximate power subsystem of figure 5.1. The piece-wise constant approximation to this effective failure is listed in table 5.6 for several consecutive averaging intervals. Notice however that calculation of $3\lambda^e(1Comp)$, $2\lambda^e(2Comp)$, $\lambda^e(3Comp)$ at the first averaging interval involves division by zero and is therefore undefined. In order to evaluate the approximate subsystem we set them to 0 with the understanding that there may exist some set of values which would better characterize the reliability of the exact subsystem across this averaging interval.

Time (hrs)	$4\lambda^e(0Comp)$	$3\lambda^e(1Comp)$	$2\lambda^e(2Comp)$	$\lambda^e(3Comp)$
0 - 50	3.9943e-04	0.0000e00	0.0000e00	0.0000e00
50 - 100	5.0605e-04	1.5350e-03	5.1755e-03	5.7934e-01
100 - 150	5.1244e-04	1.2782e-03	2.0453e-03	1.2720e-01
150 - 200	5.1862e-04	1.1902e-03	1.4680e-03	5.6292e-02
200 - 250	5.2458e-04	1.1445e-03	1.2034e-03	3.2600e-02
250 - 300	5.3034e-04	1.1157e-03	1.0525e-03	2.1765e-02
300 - 350	5.3591e-04	1.0954e-03	9.5470e-04	1.5850e-02
350 - 400	5.4130e-04	1.0800e-03	8.8630e-04	1.2242e-02
400 - 450	5.4652e-04	1.0677e-03	8.3560e-04	9.8610e-03
450 - 500	5.5158e-04	1.0574e-03	7.9653e-04	8.1976e-03

Table 5.6: λ^e Using Euler Approximation (T_{avg} = 50.0 hours)

The effective failure rate shown in table 5.6 can now be used in the numerical evaluation of the approximate subsystem model of figure 5.3. The results of this evaluation are listed in table 5.7.

Time (hrs)	$P_{(0,1)}(time)$	$P_{(1,1)}(time)$	$P_{(2,1)}(time)$	$P_{(3,1)}(time)$
0.0	1.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
50.0	9.8023E-01	1.9773E-02	0.0000E+00	0.0000E+00
100.0	9.6053E-01	3.7274E-02	2.0068E-03	1.3863E-05
150.0	9.4092E-01	5.3956E-02	5.5989E-03	6.6691E-05
200.0	9.2143E-01	6.9758E-02	7.8346E-03	1.7168E-04
250.0	9.0208E-01	8.4688E-02	1.1676E-02	3.3656E-04
300.0	8.8287E-01	9.8768E-02	1.6071E-02	5.6937E-04
350.0	8.6384E-01	1.1202E-01	2.0966E-02	8.7744E-04
400.0	8.4499E-01	1.2449E-01	2.6311E-02	1.2670E-03
450.0	8.2633E-01	1.3618E-01	3.2056E-02	1.7431E-03
500.0	8.0788E-01	1.4713E-01	3.8152E-02	2.3098E-03

Table 5.7: Approximate State Probabilities (T_{avg} = 50.0 hours)

The results of table 5.7 can now be compared to the exact subsystem data (as in table 5.4) and the difference between these two sets of results can be used to estimate the total error in our effective failure. This estimate of the total (absolute) error is listed in table 5.8.

Time (hrs)	$P_{(0,1)}(time)$	$P_{(1,1)}(time)$	$P_{(2,1)}(time)$	$P_{(3,1)}(time)$
50.0	-1.9803E-04	-3.0857E-04	5.7832E-04	2.7802E-06
100.0	-3.9470E-04	5.9451E-04	-1.3446E-04	8.2443E-06
150.0	-5.8960E-04	1.2923E-03	-5.9786E-04	6.2955E-06
200.0	-7.8173E-04	1.8842E-03	-7.4667E-04	-3.1062E-06
250.0	-9.7104E-04	2.3988E-03	-9.1463E-04	-1.6260E-05
300.0	-1.1573E-03	2.8494E-03	-1.0175E-03	-3.1322E-05
350.0	-1.3403E-03	3.2442E-03	-1.0660E-03	-5.7216E-05
400.0	-1.5195E-03	3.5894E-03	-1.0678E-03	-6.3086E-05
450.0	-1.6948E-03	3.8899E-03	-1.0298E-03	-7.8107E-05
500.0	-1.8655E-03	5.1491E-03	-9.5699E-04	-9.1617E-05

Table 5.8: Total (absolute) Error in Approximate Subsystem Model Probabilities (T_{avg} = 50.0 hours)

By comparing tables 5.8 and 5.7, we conclude that the effective failure rate for this choice of averaging interval yields approximate state probabilities with about 2 to 3 digits of precision. Given that the reliability parameters (e.g., failure rates) of the exact subsystem are only accurate to within 1 to 2 digits of precision for most real world applications, we might conclude that λ^e calculated above is accurate enough. However, it may still be possible to gain even more accuracy by increasing T_{avg} even further. Consequently, we repeat the above experiment with T_{avg} equal to 100 hours. The resulting effective failure rate is listed in table 5.9.

Notice that the values of table 5.9 are close to the values for the effective failure rate in table 5.6. Since the two effective failure rates are equivalent to within an order of magnitude, we expect both effective failure rates to yield approximate state probabilities which are consistent to within one significant digit. We already know that the results have about two digits of precision when T_{avg} equals 50 hours. The same analysis which led to the results of table 5.8 is repeated for T_{avg} equal to 100 hours. The results are listed in table 5.10.

Time (hrs)	$4\lambda^e(0Comp)$	$3\lambda^e(1Comp)$	$2\lambda^e(2Comp)$	$\lambda^e(3Comp)$
0 - 50	4.9879E-04	0.0000e00	0.0000e00	0.0000e00
50 - 100	4.9879E-04	0.0000e00	0.0000e00	0.0000e00
100 - 150	5.1142e-04	1.4827e-03	2.6683e-03	1.8010e-01
150 - 200	5.1142e-04	1.4827e-03	2.6683e-03	1.8010e-01
200 - 250	5.2353e-04	1.2205e-03	1.3589e-03	3.8419e-02
250 - 300	5.2353e-04	1.2205e-03	1.3589e-03	3.8419e-02
300 - 350	5.3383e-04	1.1648e-03	1.0878e-03	1.8944e-02
350 - 400	5.3383e-04	1.1648e-03	1.0878e-03	1.8944e-02
400 - 450	5.4488e-04	1.1010e-03	8.9455e-04	1.0915e-02
450 - 500	5.4488e-04	1.1010e-03	8.9455e-04	1.0915e-02

Table 5.9: λ^e (T_{avg} = 100.0 hours)

Time (hrs)	$P_{(0,1)}(time)$	$P_{(1,1)}(time)$	$P_{(2,1)}(time)$	$P_{(3,1)}(time)$
50.0	-2.2634e-04	-2.8021e-04	5.7832e-04	2.7802e-06
100.0	-7.6840e-04	-1.2321e-03	1.8724e-03	2.2107e-05
150.0	-1.0007e-03	7.4950e-05	8.1355e-04	3.0165e-05
200.0	-1.5132e-03	1.9294e-03	-1.3829e-04	6.8335e-05
250.0	-1.7314e-03	2.7478e-03	-5.4116e-04	8.4798e-06
300.0	-2.1993e-03	3.9116e-03	-8.8328e-04	6.1261e-05
350.0	-2.4463e-03	5.6587e-03	-1.1015e-03	-1.6765e-05
400.0	-2.9143e-03	5.6866e-03	-1.2556e-03	2.6554e-05
450.0	-3.1236e-03	6.1143e-03	-1.2184e-03	-3.7680e-05
500.0	-3.5307e-03	6.7803e-03	-1.1380e-03	6.5780e-06

Table 5.10: Total (absolute) Error in Approximate Subsystem Model Probabilities (T_{avg} = 100.0 hours)

Comparison of the results in table 5.10 with table 5.7 confirms that the approximate state probabilities in this case have approximately one digit of precision. This is consistent with the results of the previous case (T_{avg} = 50 hours) to one significant digit as expected. A comparison of tables 5.10 and 5.8 shows that the approximate state probabilities of table 5.10 are noticeably less accurate. This suggests that the effective failure rates of table 5.9 (T_{avg} = 100 hours) are less accurate than the effective failure rates of table 5.6 (T_{avg} = 50 hours). Presumably, we have passed the optimum value of T_{avg} and our results are now affected primarily by integration error.

In order to gain some intuition about the integration error for this particular example, we derive an exact solution for $4\lambda^e(0Comp)$. The probability contained in the first state of the Markov model in figure 5.3 is described by equation 5.6a where $P_{(0,1)}(t)$ is the probability of being in state 0 at time t. This equation can be solved for $4\lambda^e(0Comp)$ as indicated by equation 5.6b. Note that if $4\lambda^e(0Comp)$ is time-invariant then equation 5.6a is the exact solution to equation 5.3a. Equation 5.6b can be used to find exactly what time-invariant transition rate will make the probability in state 0 go from $P_{(0,1)}(t_i)$ to $P_{(0,1)}(t_i + T_{avg})$. The value of $\lambda^e(0Comp)$ derived from equation 5.6b does not assume a constant value for $P_{(0,1)}$. Consequently, the value derived from this equation represents the best constant value that can be achieved for $\lambda^e(0Comp)$.

$$P_{(0,1)}(t_i + T_{avg}) = P_{(0,1)}(t_i)e^{-4\lambda^e(0Comp)T_{avg}} \tag{5.6a}$$

$$4\lambda^e(0Comp) = -\ln\left(\frac{P_{(0,1)}(t_i + T_{avg})}{P_{(0,1)}(t_i)}\right)\frac{1}{T_{avg}} \tag{5.6b}$$

In order to gain some appreciation for how much our current solution (table 5.9) varies from this ideal solution, we calculate the values of $4\lambda^e(0Comp)$ using equation 5.6b and compare them to the values listed in table 5.9. This comparison can be found in table 5.11.

Averaging Interval	$4\lambda^e(0Comp)$ Exact	$4\lambda^e(0Comp)$ Euler	Difference
0 - 100	4.0696E-04	3.9879E-04	8.1696E-06
100 - 200	4.2012E-04	4.1142E-04	8.7028E-06
200 - 300	4.3276E-04	4.2353E-04	9.2305E-06
300 - 400	4.4353E-04	4.3383E-04	9.6919E-06
400 - 500	4.5508E-04	4.4488E-04	1.0199E-05

Table 5.11: Comparison of Euler Failure Rate to Exact (T_{avg} = 100 hours)

The above analysis suggests that the effective failure rate for this example at the zeroth state has approximately 1 to 2 digits of precision across the entire interval from 0 to 500 hours. In order to see how this error depends on time, we concentrate on the first and last averaging intervals. The first is the most accurate value of $4\lambda^e(0Comp)$, and the last is the least accurate. Values for $4\lambda^e(0Comp)$ calculated by Euler's method are compared to the exact solution, as a function of T_{avg} (figure 5.6). Since this transition rate is the least susceptible to roundoff error, we use the comparison of figure 5.6 to examine integration error only.

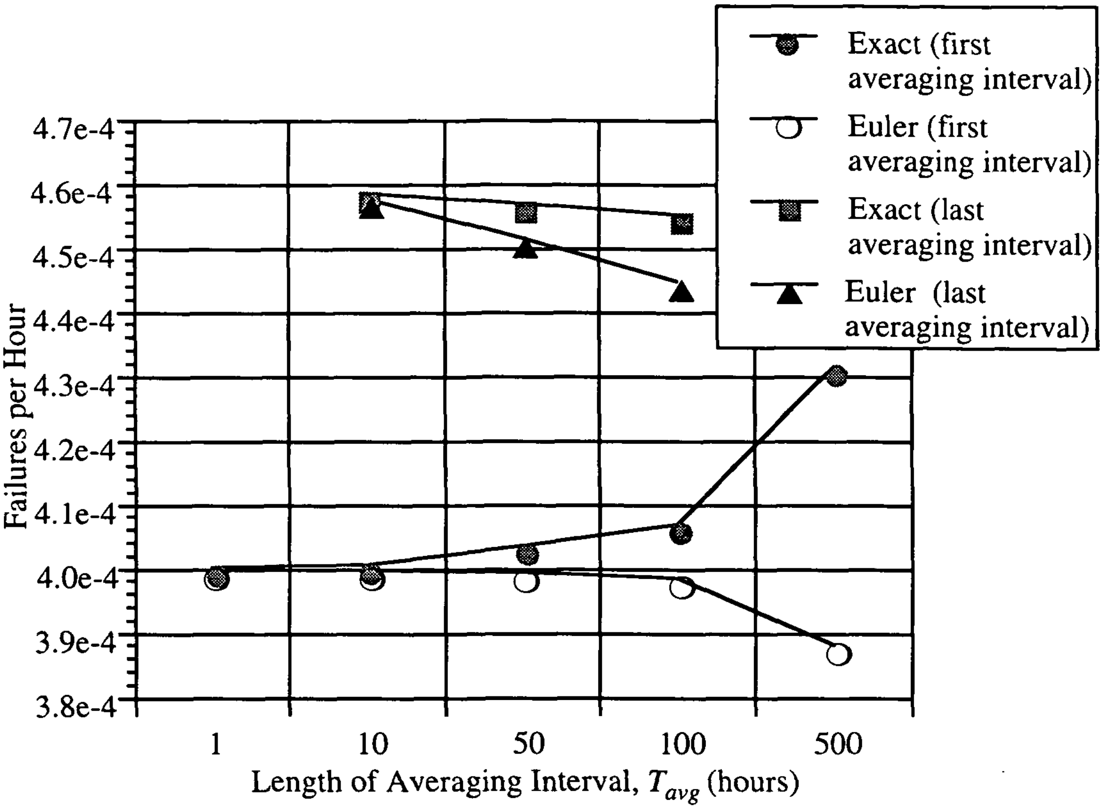


Figure 5.6: Euler vs. Exact Value for 4λ^e(0 Comp)

The trend in the above data is clear: as the length of the averaging interval increases, the integration error in our averaging process becomes unacceptably large. Integration error is almost zero when the length of the averaging interval is 1 hour. Of course, for such a small T_{avg} we get large roundoff error in the other effective rates. Also, integration error starts to become noticeable even with an averaging interval of 50 to 100 hours. We conclude that there exists only a narrow band of values for T_{avg} which will produce an accurate set of effective failure rates (i.e., from about 50 to 100 hours).

5.4 Improved Integration

Only a narrow set of values for T_{avg} produce an accurate effective failure rate for the power subsystem. This represents a limitation on the hierarchical modeling process. It seems likely that this range of values will change from one system to the next; that the range will grow or shrink, and most certainly change location. Finding a potentially small range of values for T_{avg} may prove cumbersome. We would like to alleviate this problem by increasing the size of this range, by reducing either roundoff or integration error.

Roundoff error is a direct result of the method chosen to solve for the effective failure rate. In particular, an expression for the effective failure rate has been derived by solving a differential equation backwards. This solution necessarily involves taking a finite difference. Since this is the source of roundoff error it seems unlikely that any other choice of integration technique will alleviate this problem. It is also unclear how to solve for the effective failure rates without numerical integration.

Integration error, unlike roundoff error, depends strongly on the choice of integration technique. One simple improvement to the Euler's method is based on the trapezoidal rule of integration [Stewart, pp. 440 - 445]. Instead of assuming a constant state probability over a given averaging interval, we assume a linear state probability. This assumption is highlighted in the sample integration presented below, and the corresponding graphical interpretation of figure 5.7.

$$\int_{t}^{t+T_{avg}} \frac{dP_{(0,1)}}{dt}dt = -\int_{t}^{t+T_{avg}} 4\lambda^{e}(OComp)P_{(0,1)}dt$$

$$\Rightarrow \int_{P_{(0,1)}(t)}^{P_{(0,1)}(t+T_{avg})} dP_{(0,1)} = -4\lambda^{e}(OComp)\frac{P_{(0,1)}(t)+P_{(0,1)}(t+T_{avg})}{2}T_{avg}$$

$$\Rightarrow P_{(0,1)}(t+T_{avg}) - P_{(0,1)}(t) = -4\lambda^{e}(OComp)\overline{P}_{(0,1)}T_{avg}$$

$$\Rightarrow \frac{\Delta P_{(0,1)}}{T_{avg}} = -4\lambda^{e}\overline{P}_{(0,1)}$$

In the above system of equations $\overline{P}_{(0,1)}$ denotes the average probability in state 0 for a given averaging interval. Notice that we obtain the same result if we use the average probability with Euler's method, as opposed to the initial state probability.

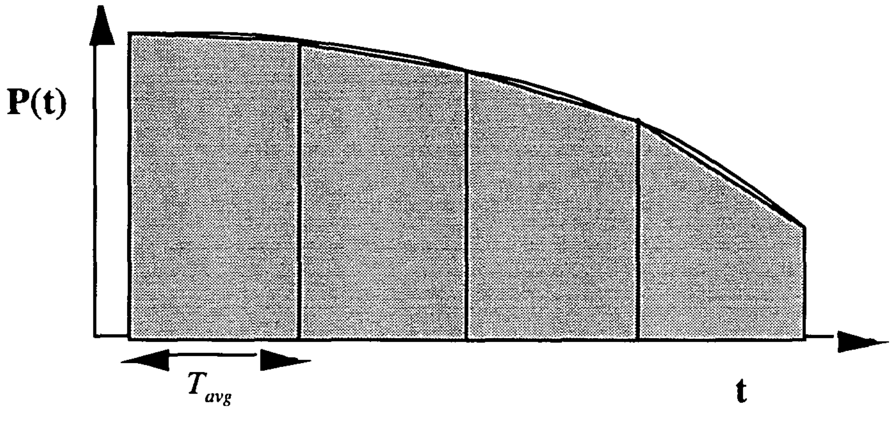


Figure 5.7: Trapezoidal Rule of Integration Applied to State Equations

Since state probabilities are approximately linear for small enough T_{avg}, the above method of integration should be more accurate than Euler's method. In order to verify this empirically, we calculate the effective failure rate for the power subsystem presented in §5.3. The resulting effective failure rate is presented in table 5.12.

Time (hrs)	$4\lambda^e(0Comp)$	$3\lambda^e(1Comp)$	$2\lambda^e(2Comp)$	$\lambda^e(3Comp)$
0 - 50	4.0346e-04	1.0410e-03	2.3643e-03	3.6676e-01
50 - 100	4.1022e-04	1.0423e-03	1.6993e-03	1.2944e-01
100 - 150	4.1674e-04	1.0396e-03	1.2822e-03	5.9141e-02
150 - 200	4.2304e-04	1.0365e-03	1.0761e-03	3.4017e-02
200 - 250	4.2913e-04	1.0332e-03	9.5578e-04	2.2482e-02
250 - 300	4.3502e-04	1.0298e-03	8.7750e-04	1.6243e-02
300 - 350	4.4071e-04	1.0265e-03	8.2231e-04	1.2465e-02
350 - 400	4.4623e-04	1.0232e-03	7.8140e-04	9.9932e-03
400 - 450	4.5156e-04	1.0199e-03	7.4973e-04	8.2762e-03
450 - 500	4.5674e-04	1.0167e-03	7.2450e-04	7.0298e-03

Table 5.12: λ^e Using Trapezoidal Rule (T_{avg} = 50.0 hours)

By using the average state probability, we have eliminated the division by zero problem present when using an Euler approximation (table 5.6). We can estimate the accuracy of the transition rates in table 5.12 by calculating approximate state probabilities as explained in §5.3. The resulting error is shown in table 5.13.

Time (hrs)	$P_{(0,1)}(time)$	$P_{(1,1)}(time)$	$P_{(2,1)}(time)$	$P_{(3,1)}(time)$
50.0	2.5700e-06	5.2520e-06	7.7375e-06	3.3808e-08
100.0	5.1800e-06	8.0060e-06	1.2558e-05	7.9523e-07
150.0	7.5200e-06	1.1212e-05	1.5904e-05	1.4107e-06
200.0	9.7700e-06	1.3820e-05	1.8904e-05	1.3400e-06
250.0	1.2000e-05	1.6102e-05	2.1760e-05	1.0580e-06
300.0	1.4300e-05	1.8490e-05	2.4192e-05	6.9436e-07
350.0	1.6360e-05	2.0226e-05	2.6775e-05	2.9625e-07
400.0	1.8700e-05	2.1990e-05	2.9347e-05	1.2710e-07
450.0	2.0600e-05	2.3230e-05	3.1796e-05	5.7770e-07
500.0	2.2790e-05	2.4260e-05	3.4487e-05	1.0677e-06

Table 5.13: Total (absolute) Error Associated with Trapezoidal Rule (T_{avg} = 50.0 hours)

By comparing tables 5.13 and 5.8, we conclude that the trapezoidal approximation results in substantially less integration error. The effective failure rate as calculated by the trapezoidal approximation can be compared to the exact failure rate as done in table 5.11. The graphical comparison presented in figure 5.6 is repeated in the graph of figure 5.8.

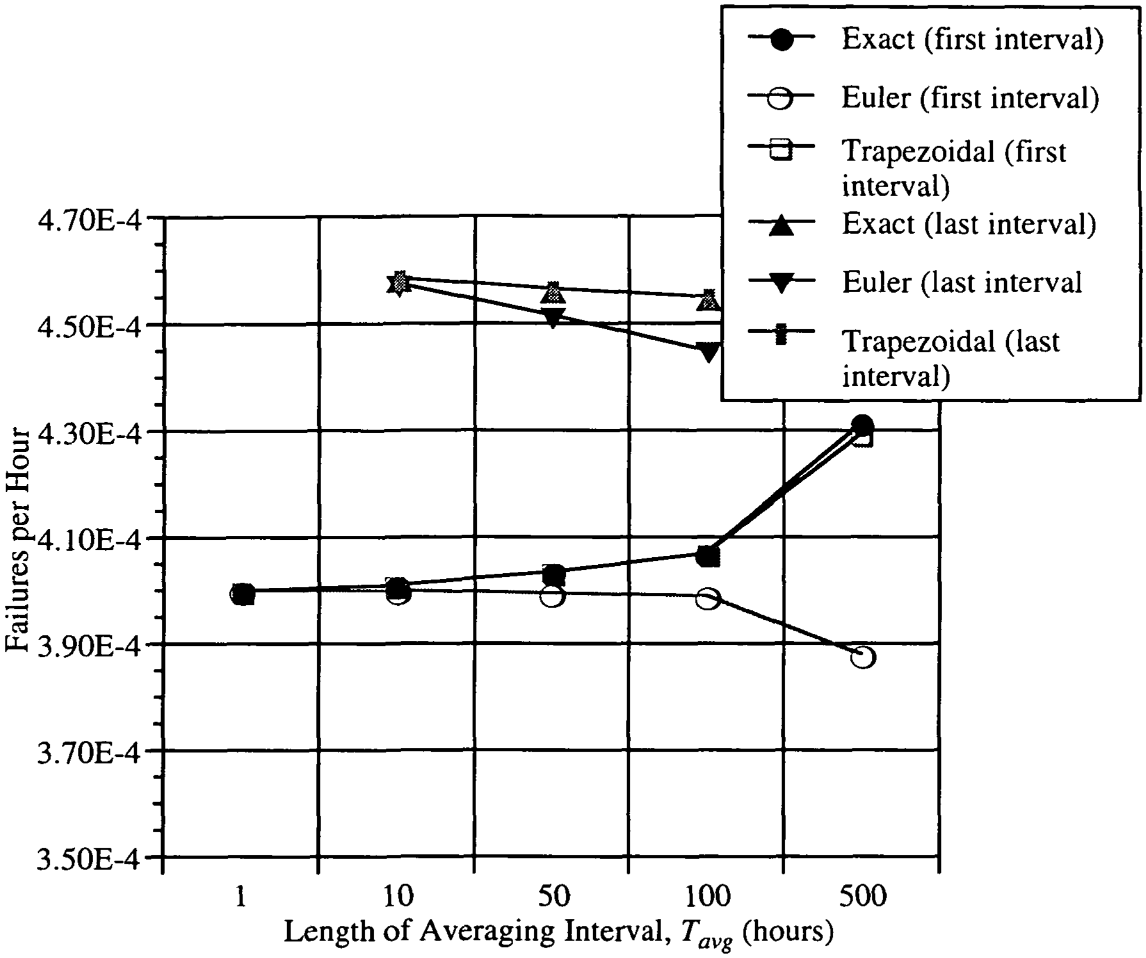


Figure 5.8: Trapezoidal vs. Exact Value for $4\lambda^e(0 \text{ Comp})$

We can conclude from figure 5.8 that the trapezoidal approximation provides a better estimate of the actual effective failure rate than a simple application of Euler's method. Also, by using the trapezoidal approximation we have increased the range of values of T_{avg} over which we may obtain a valid effective failure rate.

Numerical tests similar to the one presented in tables 5.12 and 5.13 can be repeated with an averaging interval of 500 hours. Such a test shows that for the given sample system, even with such a large averaging interval, the effective failure rate calculated by a trapezoidal approximation still produces results with at least one digit of precision. With an averaging interval of 1000 hours, the level of error becomes unacceptably large for this power subsystem example.

This suggests that a higher order approximation to the behavior of state probabilities across an averaging interval might further reduce the integration error (e.g., a quadratic approximation such as Simpson's rule [Stewart, pp. 440 - 445]). However, for the representative systems explored so far, the trapezoidal approximation has been sufficient. Finally, notice that by applying a trapezoidal approximation we have not eliminated the dependence of λ^e on the change in state probabilities. This means that we have not affected roundoff error at all.

5.5 Dependence of T_{avg} on Component Failure Rates

Both the integration and roundoff error depend on how quickly the state probabilities $P_{(0,1)}$ through $P_{(3,1)}$ change over a given averaging interval. Consequently, the accuracy of the resulting effective failure rate depends on our choice of averaging interval. The accuracy of our results must also depend on any other parameters which affect the change in state probabilities. The failure rate of components within the exact subsystem have a strong affect on the resulting state probabilities $P_{(0,1)}$ through $P_{(3,1)}$. Consequently, the accuracy of the effective component failure rate, for a given averaging interval, must also depend on the component failure rates of the exact subsystem.

In order to gauge what effect the component failure rates of the exact subsystem have on our choice of an appropriate T_{avg}, we assume that the choice of an appropriate T_{avg} will be inversely proportional to the sum of the original component failure rates. For the power subsystem considered in §5.3, the appropriate T_{avg} was approximately 50 hours. In this case each component shared the same constant failure rate, λ_{Share}, of 10^{-4}. The sum of these component failure rates was 10^{-3}. For this sample system, the shared component failure rate, λ_{Share} was varied several times, and an appropriate T_{avg} was determined empirically. The results of this test are summarized in table 5.14.

λ_{Share}	1e-3 Failures/hr	1e-4 Failures/hr	1e-5 Failures/hr	1e-6 Failures/hr
$\sum_{10Comps}\lambda_{Share}$	1e-2 Failures/hr	1e-3 Failures/hr	1e-4 Failures/hr	1e-5 Failures/hr
T_{avg}	5 hrs	50 hrs	500 hrs	5000 hrs

Table 5.14: Effect of Component Failure Rates on $\mathbf{T_{avg}}$

In table 5.14, λ_{Share} represents the shared component failure rate. The data of table 5.14 indicates the following relationship for the given example.

$$T_{avg} \approx \frac{1}{20 \sum_{comps} \lambda} \qquad (5.7)$$

Equation 5.7 represents a rough estimate of the appropriate averaging interval, and in practice an averaging interval 5 to 10 times greater may still be sufficient. Initially, we assume that the approximate result of equation 5.7 holds generally.

In practice, we can validate our choice of T_{avg} by varying this parameter and observing what effect this has on the resulting effective failure rates. If changes in the T_{avg} produce comparable changes in the effective failure rates, we may conclude that the resulting effective failure rates are still a function of T_{avg}. If this is the case we must continue to vary T_{avg} until the effective failure rates cease to change substantially with the choice of T_{avg}.

Alternatively, the validity of T_{avg} can be tested by another means. The effective transition rates which result from a particular choice of T_{avg} can be used in a numerical evaluation of the approximate subsystem model. If the resulting state probabilities closely approximate the state probabilities from the exact subsystem model, we conclude that the given T_{avg} is appropriate.

5.6 Conclusions

In this chapter we have explored some important numerical issues which arise in calculating the effective component failure rates. The system of difference equations which results from numerically integrating the state equations for the approximate subsystem model is especially sensitive to roundoff error. This makes traditional matrix based solution techniques difficult to apply. It is hoped that by writing separate equations for each effective transition rate in terms of known quantities that roundoff error is reduced substantially. Also, these finite differences arise from the use of numerical integration of the approximate subsystem model's state equations. Integration error is reduced by using integration based on the trapezoidal rule.

We have seen that determining a value for the length of the averaging interval represents a tradeoff between roundoff and integration error. This relationship has been explored in detail for a small sample system. For this sample system, an empirical relationship between T_{avg} and the sum of exact subsystem component failure rates was developed (equation 5.7). This relationship was proposed as a method for producing an initial guess for T_{avg}, with two methods provided to validate the value of T_{avg} selected.

Chapter 6

Hierarchical Analysis of Some Sample Systems

One objective of this thesis is to show that there exist some systems which may accurately be analyzed with the hierarchical technique proposed. Until now a hierarchical technique has been presented in some detail, but its validity has only been argued heuristically. Here we use empirical data to support the accuracy of the hierarchical modeling process.

Another objective of this thesis is to show that the hierarchical modeling technique is useful in making many intractable systems tractable. In the latter half of this chapter, a real world example of an intractable system is presented. The hierarchical modeling technique is then used to obtain an accurate reliability estimate for this system.

6.1 Two Sample Systems

6.1.1 No External Dependence

Consider the system of figure 6.1. The processor core in this system has no dependence on any components in the remainder of the exact system. Thus, we label this system as one having "no external dependence", in order to distinguish it from the next example. This system can also be found in figure 3.5a, and a description of how this system behaves can be found in §3.3.

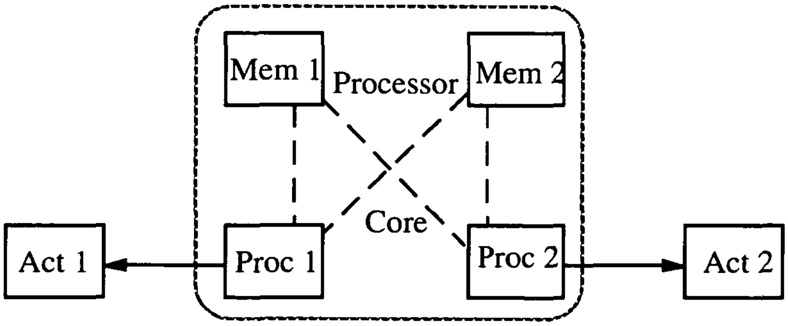


Figure 6.1: Exact System - No External Dependence

According to the guidelines of chapter 3, we should accurately be able to replace the exact processor core with two effective processors. The resulting approximate system can be found in figure 6.2. The exact processor core was evaluated with a failure rate of

10^{-4} (failures per hour) assigned to each of the memory units and 10^{-5} assigned to each of the processor units.

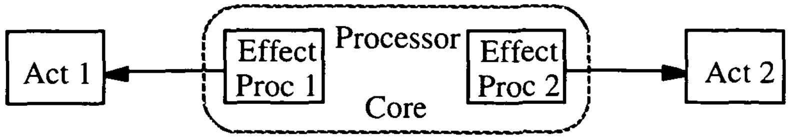


Figure 6.2: Approximate System - No External Dependence

In order to evaluate the approximate system of figure 6.2 we first need to calculate effective failure rates for the approximate processor core. We start with the approximate subsystem model of figure 4.3. The state space of this approximate subsystem model is used to help generate the exact subsystem model of figure 4.5. As suggested in §4.2.2, the probability of the two common mode failure states is evenly split between the two system loss states of the approximate subsystem model. It should be noted however, that this probability assignment is arbitrary. Since the remainder of the exact system depends only on the availability of specific processor channels, and not on the order in which they fail, the resulting approximate subsystem need only capture the aggregate system loss probability. How the common mode failure probability is assigned to either of the system loss states has no bearing on the accuracy of the resulting approximate system.

For this system, the approximate relation of equation 5.7 suggests an averaging interval of about 200 hours. Empirically we find that an averaging interval of 100 is more appropriate for this system. Also, all of the states in the approximate subsystem model for this example will contain finite probability. Thus we would like to use the closed form solution for the effective failure rates of this system, derived in §4.1 (equations 4.4). In order to gain the additional accuracy afforded by the trapezoidal approximation we simply replace the state probabilities evaluated at the beginning of each averaging interval with the corresponding average state probabilities (e.g. $P_{(0,1)}(nT_{avg}) \rightarrow \overline{P_{(0,1)}}$). Although we do not list the resulting numerical values, we note that the effective failure rates for this system are very similar to the rates in table 4.2. The reliability of the approximate system of figure 6.2 can be determined by evaluating the Markov model of figure 6.3.

Assume that a failure rate of 5.0x10^{-5} (failures per hour) is assigned to both actuators. These failure rates, along with the numerical values determined for the effective transition rates can be used to evaluate the Markov model of figure 6.3. Because this example is small enough, a Markov model for the exact system can also be generated and evaluated. These reliability estimates can be compared to determine the actual error associated with this hierarchical analysis. For this example, the absolute error associated with both the reliability and unreliability estimates are almost identical. The error associated with the reliability estimate is plotted in figure 6.4.

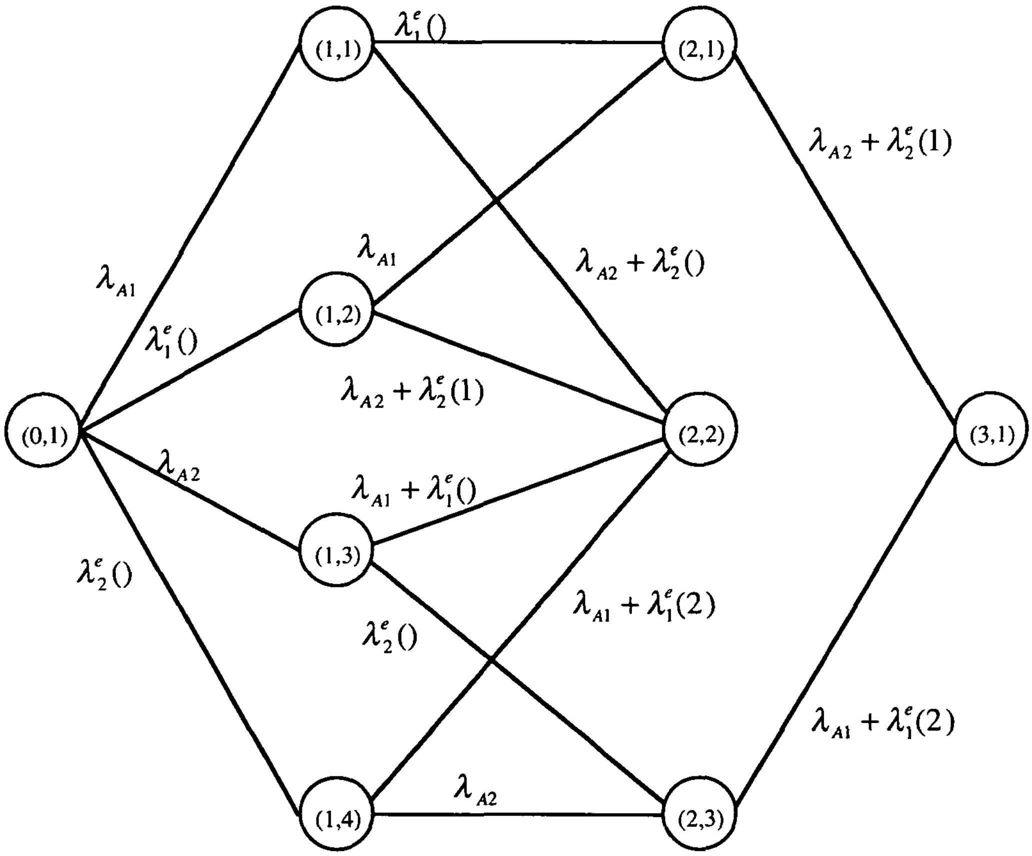

(0,1)
(1,1)
(1,2)
(1,3)
(1,4)
(2,1)
(2,2)
(2,3)
(3,1)
$\lambda_1^e()$
λ_{A1}
$\lambda_1^e()$
λ_{A2}
$\lambda_2^e()$
λ_{A1}
$\lambda_{A2}+\lambda_2^e(1)$
$\lambda_{A2}+\lambda_2^e()$
$\lambda_{A1}+\lambda_1^e()$
$\lambda_2^e()$
$\lambda_{A1}+\lambda_1^e(2)$
λ_{A2}
$\lambda_{A2}+\lambda_2^e(1)$
$\lambda_{A1}+\lambda_1^e(2)$

Figure 6.3: Markov Model for Approximate System - No External Dependence

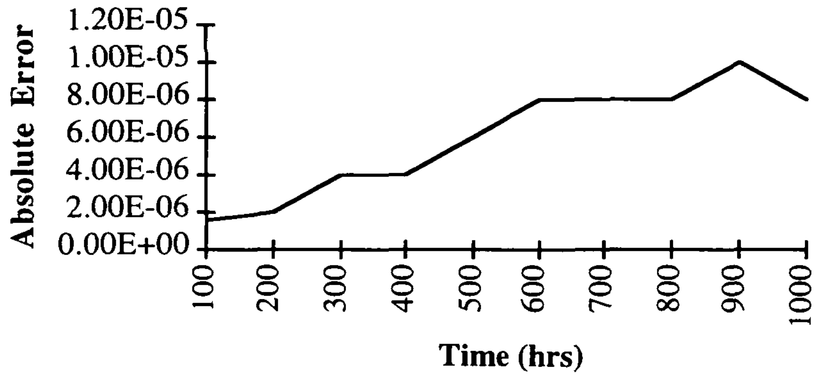


Figure 6.4: Absolute Error in Reliability/Unreliability - No External Dependence

The reliability for this system is on the order of 10^0 for the times listed in figure 6.4. On the other hand, the unreliability ranges in value from about 10^{-4} to 10^{-3} for this range of time values. This leads to substantially different amounts of relative error. The relative error associated with the reliability estimate is plotted in figure 6.5. Compare this to the relative error associated with the unreliability estimate plotted in figure 6.6.

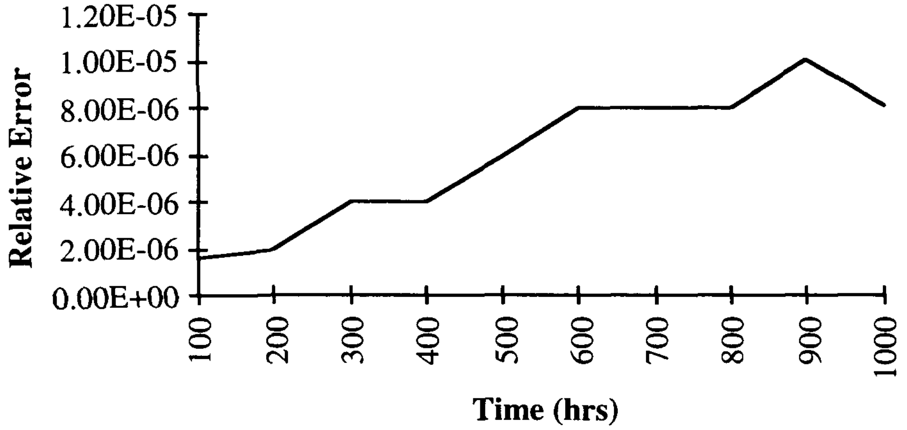


Figure 6.5: Relative Error in Reliability - No External Dependence

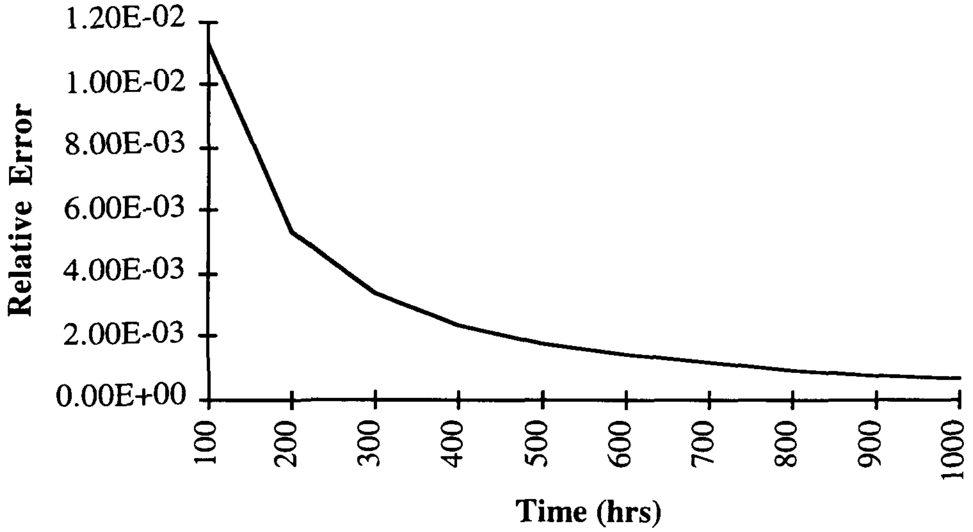


Figure 6.6: Relative Error in Unreliability - No External Dependence

Because the reliability for this system is very close to one for the range of time indicated, the relative error for this example closely matches the absolute error of figure 6.4. However, the relative error for the unreliability estimate decreases substantially with increasing unreliability. The unreliability estimate is substantially less accurate: the reliability for this system has about six digits of precision, the unreliability has only two or three.

Traditional techniques for reliability analysis generally produce unreliability results which are only accurate to within one to two digits of precision due to uncertainty in component reliability parameters (e.g., failure rate). We conclude that for this example, the hierarchical analysis technique produces a reasonably accurate approximation to the reliability/unreliability estimate obtainable by a direct analysis of the corresponding exact system.

6.1.2 Global Dependence

The sample system considered in §6.1.1 contains a subsystem which exhibits no dependence on the remainder of the exact system. Consider now the system of figure 6.7. This system contains a processor which exhibits a global dependence on the power supply. However, according to the guidelines of chapter 3 we should still be able to replace the processor core of this system with the approximate subsystem implicit in figure 6.2. The resulting approximate system is shown in figure 6.8.

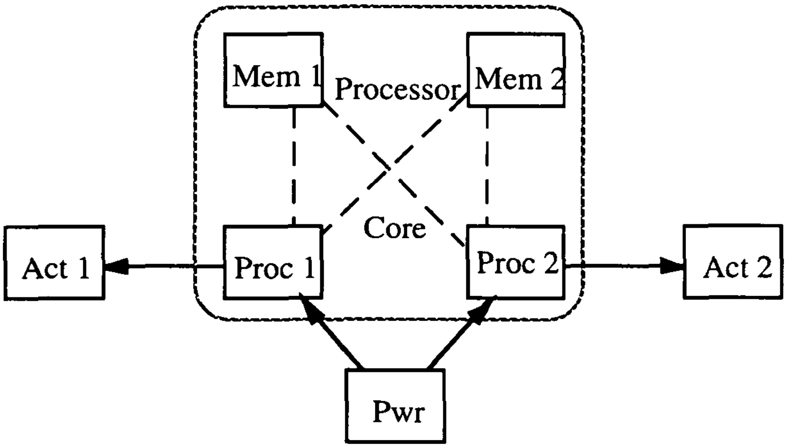


Figure 6.7: Exact System - Global Dependence

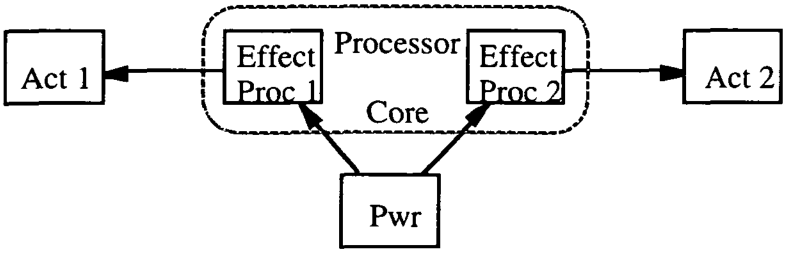
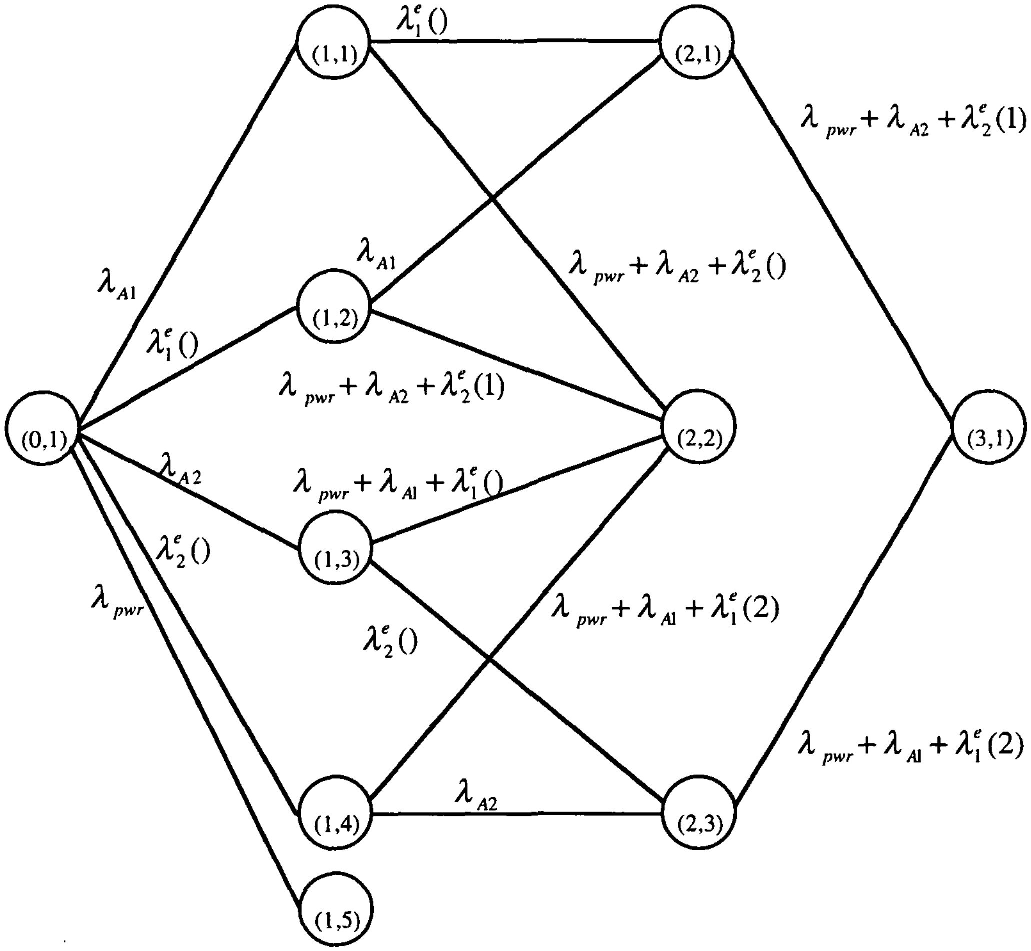


Figure 6.8: Approximate System - Global Dependence

The approximate system of figure 6.8 uses the same effective failure rates as the approximate system of figure 6.2. The effect of the global power supply is captured in the Markov model for the approximate system, shown in figure 6.9.



$\lambda_1^e()$
(1,1)
(2,1)
$\lambda_{pwr} + \lambda_{A2} + \lambda_2^e(1)$
λ_{A1}
λ_{A1}
$\lambda_{pwr} + \lambda_{A2} + \lambda_2^e()$
$\lambda_1^e()$
(1,2)
$\lambda_{pwr} + \lambda_{A2} + \lambda_2^e(1)$
(0,1)
(2,2)
(3,1)
λ_{A2}
$\lambda_{pwr} + \lambda_{A1} + \lambda_1^e()$
$\lambda_2^e()$
(1,3)
λ_{pwr}
$\lambda_{pwr} + \lambda_{A1} + \lambda_1^e(2)$
$\lambda_2^e()$
$\lambda_{pwr} + \lambda_{A1} + \lambda_1^e(2)$
λ_{A2}
(1,4)
(2,3)
(1,5)

Figure 6.9: Markov Model for Approximate System - Global Dependence

Assume that a failure rate of 1.0x10^{-5} failures per hour is assigned to the power unit and the actuators for this system have the same failure rate as the actuators for the system of §6.1.1. The reliability and unreliability of the exact system may be evaluated directly. The results of this evaluation may be compared to an evaluation of the approximate system of figure 6.9.

Absolute error between estimates for system reliability and unreliability for this system are similar (i.e., they differ by no more than a factor of 2). The absolute error associated with the estimate for system unreliability, obtained through a hierarchical analysis is plotted in figure 6.10. Since system reliability continues to be on the order of one, the relative error for this estimate closely matches the absolute error of figure 6.10. System unreliability varies from 10^{-3} to 10^{-2} for the times of interest. This results in a relative error which decreases from about 10^{-3} to 10^{-4}.

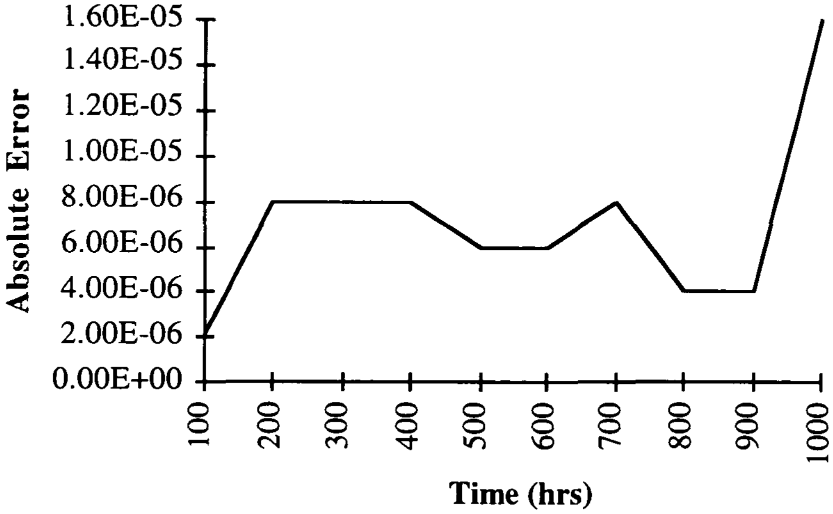


Figure 6.10: Absolute Error, System Unreliability- Global Dependence

The relative error associated with this hierarchical analysis suggests that the reliability and unreliability estimates obtained have about five and three digits of precision, respectively. Based on this information and the discussion at the end of §6.1.1, we conclude that the hierarchical analysis also produces acceptably accurate estimates for system reliability and unreliability when the subsystem has a global dependence on the remainder of the system.

The two examples of §6.1 do not prove that the hierarchical modeling technique proposed in this report can accurately be applied to all systems or even to a certain class of systems. However, the analysis of these two sample systems does demonstrate that this technique can accurately be applied to some systems. This technique was proposed for the analysis of large systems which could not otherwise be analyzed with the aid of Markov models.

6.2 An Automated Markov Model Construction Tool

The construction of even moderate sized Markov models is an error prone and tedious process. To deal with this, several software tools have been developed to assist in the model construction process [Nicole, Goyal]. A unique example of this class of tools is the CAME program.

The Computer Aided Markov Evaluator (CAME) [Hutchins] has been produced at the Charles Stark Draper Laboratory to assist in the reliability analysis of large fault-tolerant systems with known constant failure rates. An important aspect of this software package is that it automatically constructs a Markov reliability model based on a user defined system description. The tool is currently implemented on a Symbolics MacIvory system.

CAME input is based on the Symbolics graphical user interface. A system can be described in CAME through four input windows. The *architecture* window is used to input components and component parameters (i.e., failure rate, repair rate, coverage values, etc.). The *reconfigurations* window can be used to define how the system reconfigures in response to a specific component failure. The *further specifications* window can be used to abstract certain definitions (e.g., specific Boolean conditions can be defined and labeled in this window). Finally, the *performance level* window is used to define different levels in which the system may be considered operational. These operating or *performance levels* are evaluated sequentially. For example, if the system meets the requirements for performance levels 1 and 2 then the system is considered to be in performance level one.

CAME also has two output windows. The *model builder* window is where the user commands CAME to generate a Markov model based on specific model building parameters (e.g., model truncation, state aggregation). This window can also be used to invoke the state inspector which allows detailed examination of the resulting Markov model. The *model evaluator* can be used to obtain numerical reliability estimates based on known constant failure rates.

6.3 Space Station Freedom

The work presented in this report was motivated by the reliability analysis of Space Station Freedom (SSF). In particular, attitude control for SSF is considered critical for station survival. Although this represents a highly reliable system, such reliability is achieved through the use of a high level of redundancy with only moderately reliable components.

The combination of high redundancy and moderately reliable components, combined with long mission (or evaluation) times, is not amenable to traditional Markovian analysis. Since component failures are common for mission times of interest, the Markov model used to evaluate system reliability must be built several failure levels deep in order to obtain satisfactory bounds on system unreliability. This may be very time consuming and it may not even be possible to produce such a large model given constraints on processor speed and memory.

Space Station Freedom is scheduled to be built in several stages. The reliability of the station at each stage, or mission build (MB), is evaluated independently. Here we examine the reliability of attitude control at MB-2. Although this represents only the second stage of space station construction it is still a fairly complicated system.

6.3.1 Traditional Analysis of SSF

Figure 6.11 represents the architecture of SSF at MB-2. Layered components represent duplicates of the given component. For example, there are two top propulsion modules.

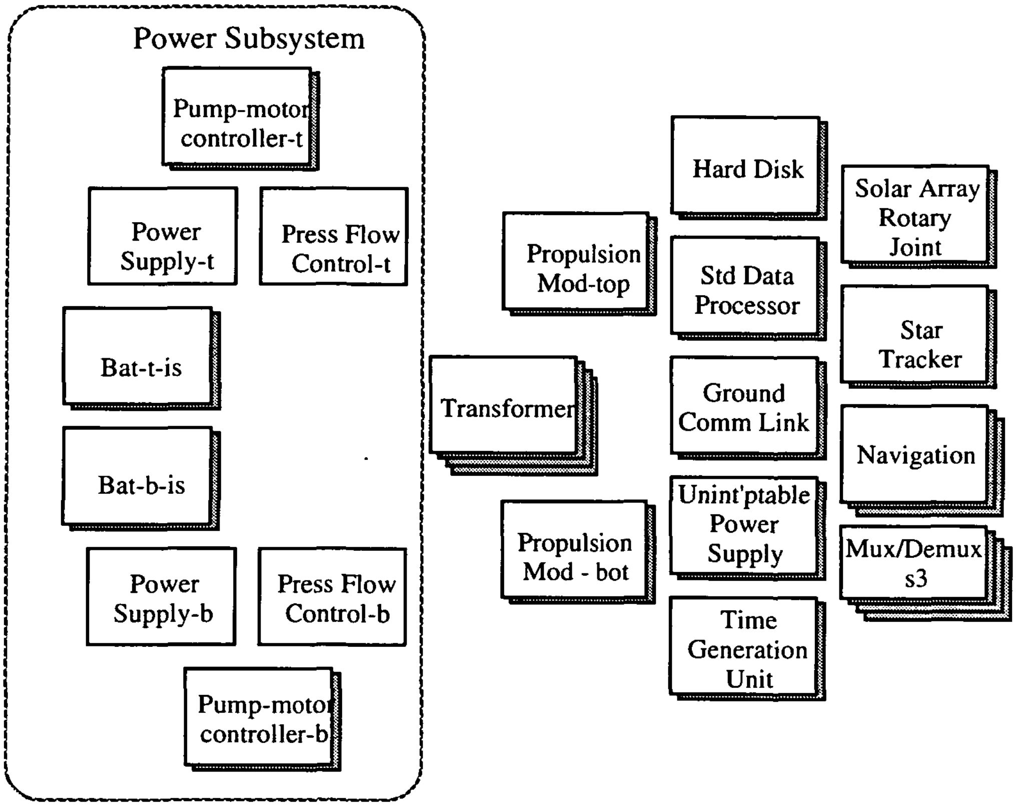


Figure 6.11: MB - 2 Exact System Architecture

In the power subsystem, both top pump motor controllers run off of the top power supply. This power supply in turn is cooled by the top pressure flow control system. Also, both of the top batteries need the top pressure flow control system and one of the two top pump motor controllers in order to function. The bottom half of the power subsystem contains a complementary set of dependencies between components. This architecture results in top and bottom power channels. In order for the top power channel to be operational, the top power supply, pressure flow control system, one of two batteries and one of two pump motor controllers must be unfailed. The bottom power channel is considered functional under a similar set of conditions.

The four transformers depend on the availability of top and bottom power channels but not on the availability of specific components within the power subsystem. These four transformers are used to supply power to the remainder of the exact system. The standard data processors however can also use power from the uninterruptable power supplies. The remaining components are used to directly sustain critical system functions such as ground communication or guidance and navigation.

The description of this system was entered into CAME. The resulting system description was then used to generate a four failure level model. This Markov model took 1 hour and 18 minutes to construct, and was evaluated with a mission time of 4320 hours, approximately 6 months. The evaluation produced bounds on system unreliability of between 0.209 and 0.290.

For this architecture, connectivity and mission time the system reliability is clearly unacceptable. This makes the lack of resolution in the resulting reliability estimate unimportant. However, in the interest of later comparison an attempt was made to improve the reliability estimate by constructing a five failure level model. In this case the machine ran out of memory before the model was completed. Both RAM and paging space (i.e., virtual memory) were exhausted.

6.3.2 Hierarchical Analysis of Baseline SSF Configuration

To apply the hierarchical technique developed in previous chapters, a subsystem must be selected for decomposition from the exact system. As was noted previously, the power subsystem interacts with the remainder of the exact system through specific power channels. This suggests that the power subsystem could be replaced with effective top and bottom power components. Alternatively, the power subsystem along with the four transformers could be replaced by four effective transformers. This decomposition eliminates only four additional components from the exact system while adding two effective components to the approximate power subsystem. This would result in a great deal of added complexity with relatively little computational benefit. Consequently, the former decomposition was pursued.

The Markov model of figure 6.12 was used to determine effective component failure rates for the approximate subsystem.

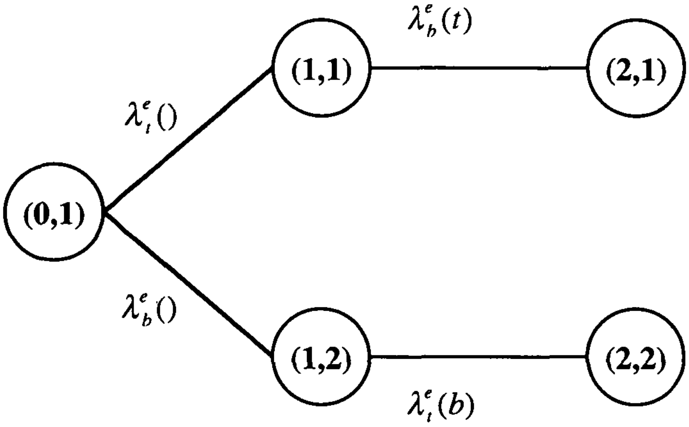


Figure 6.12: Approximate Power Subsystem Markov model

Given the Markov model of figure 6.12, state probabilities were needed to calculate the indicated effective failure rates. In past examples both the exact and approximate subsystems were small enough that their respective Markov models could be constructed by hand. However, for this system the automated model construction capabilities of the CAME program were used to obtain the necessary state probabilities.

CAME was used to build a Markov model for the exact subsystem with states which map into the states of figure 6.12. A system description for the exact power subsystem alone was produced. In this system description the top and bottom power channels are defined by the Boolean expressions of equations 6.1. The Markov model pictured in figure 6.12 was entered as a reconfiguration diagram with transitions triggered by the evaluation to "false" of the effective channel definitions. For example, the reconfiguration from state (0,1) to state (1,1) was triggered by the logical expression "not top-power". Finally, a different performance level was defined for each state in the approximate subsystem's Markov model.

top-power = (unfailed power-supply-t) and (unfailed pressure-flow-control-system-t)
and ((unfailed pump-motor-controller-1-t) (6.1a)
or (unfailed pump-motor-controller-2-t))
and ((unfailed battery-1-t) or (unfailed battery-2-t))

bot-power = (unfailed power-supply-b) and (unfailed pressure-flow-control-system-b)
and ((unfailed pump-motor-controller-1-b) (6.1b)
or (unfailed pump-motor-controller-2-b))
and ((unfailed battery-1-b) or (unfailed battery-2-b))

The resulting system description for the exact power subsystem was then used to generate the appropriate Markov model. Since the exact subsystem in question was reasonably small, this Markov model was built without truncation. Thus for this example, no error was introduced due to truncation of the exact subsystem model.

The failure rates for the exact subsystem components sum to $1.00\mathrm{x}10^{-3}$. This yields a rough estimate for the averaging interval of 50 hours, given the rule of thumb indicated in chapter 4 (i.e., $T_{avg} \approx [20(\sum \lambda_i]^{-1}$). However, 108 hours divides evenly into the mission time and this was used for the averaging interval.

In order to calculate all of the unknown transition rates for the Markov model in figure 6.12 the system of finite difference equations which describe that model was solved to obtain an approximation to the effective failure rates over several time intervals. The closed form solution to all of the unknown transition rates is directly associated with the approximate subsystem model. Consequently equations 4.4 represent a closed form solution to the set of effective transition rates for this subsystem as well as the subsystem of figure 4.2. In order to use equations 4.4 we identify the top power channel as effective component 1, and the bottom power channel as effective component 2.

The closed form of the approximate solution to all of the unknown transition rates was put into a spreadsheet. Once the Markov model for the exact subsystem was evaluated, all of the state probabilities were transferred to this spreadsheet to produce a piece-wise constant approximation to the effective transition rates. Once these rates were determined, they had to be expressed in terms of the model description tools available within CAME.

The state dependence of the effective failure rates was captured by using several components to express all of the possible states for each effective component. All of the components used to describe a given effective component were defined as "cold" spares. Reconfiguration diagrams were used to indicate which components were in use at any given time. The architecture used in the CAME description is shown in figure 6.13.

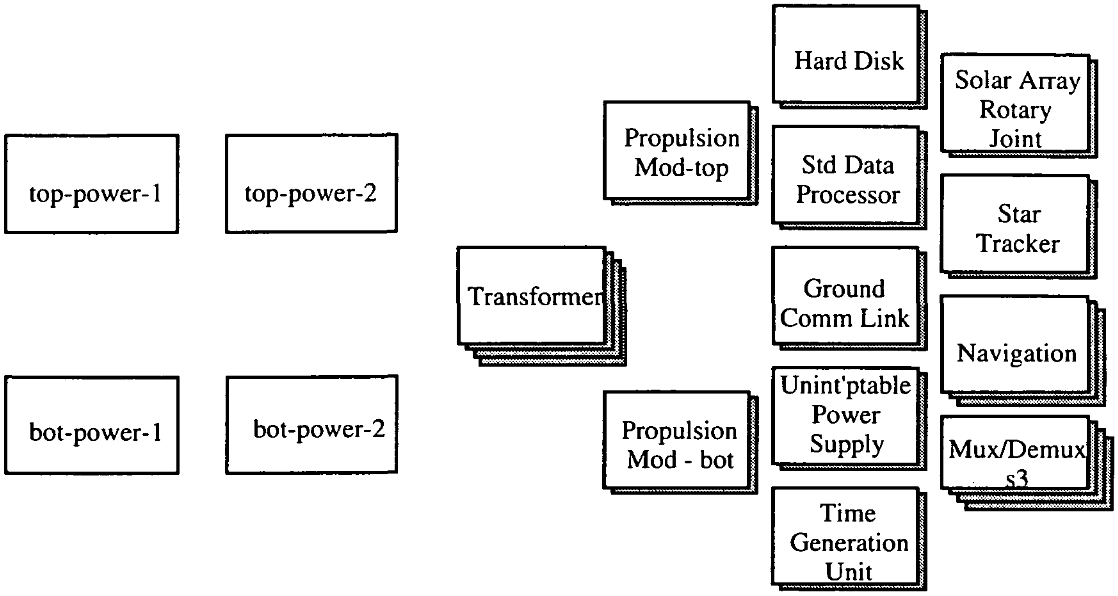


Figure 6.13: MB - 2 Approximate System Architecture

Since CAME only accepts constant failure rates, the time varying nature of the resulting effective failure rates was used merely to obtain bounds on overall system unreliability. The maximum value of each transition rate over the whole mission time was used to obtain an upper bound on system unreliability. The minimum value was used to obtain the corresponding lower bound. Since there was extremely little variation over time for this particular example, this did not prove to be a limiting factor on the accuracy of the final reliability estimate.

As was noted previously, the Markov model for the exact subsystem was built without truncation. Therefore the only contributing factors to the bounds on unreliability were truncation of the approximate system model and the bounds used to express the time varying failure rates.

The approximate system architecture in figure 6.13 was used to generate a four failure level Markov model. This model was produced in approximately 24 minutes. Since it took roughly 2 minutes to generate the exact subsystem model the total machine time necessary to produce this Markov model was about 26 minutes. As a figure of merit this time is misleading because it does not incorporate time spent by the user performing different facets of the hierarchical analysis by hand. However, it does suggest that if these tasks can be automated efficiently, that hierarchical modeling may produce a substantial reduction in the amount of computer time necessary to analyze this type of system.

The four failure level model yielded somewhat unsatisfactory though better bounds on system unreliability (i.e., better than the original four failure level analysis of the exact subsystem). In order to test the limits of this technique, a five failure level Markov model, which could not be generated without decomposition, was produced. Although this proved to require a great deal of machine time, the bounds which resulted from this model were substantially better than could have otherwise been achieved. The results of this analysis are compared with previous analyses in table 6.1.

System	Failure Levels	Time(hours)	Bounds
Exact	4	1.30	0.209 - 0.290
Exact	5	N/A	N/A
Approximate	5	7.92	0.253 - 0.266

Table 6.1: Hierarchical vs. Traditional Analysis - Initial Architecture

6.3.3 Improving Baseline SSF Architecture

Although the bounds on system unreliability were large for the direct (i.e., not hierarchical) evaluation of baseline architecture, this point seems irrelevant given that even the lower bound unreliability is substantially more than would be tolerable. For all practical purposes, the reliability of the system must be improved before the accuracy of our final answer becomes a serious concern.

A Markov model for the exact system was used to help improve the reliability of the baseline SSF configuration in the following way [Zinchuk]. By varying component failure rates one at a time, components which affected the lower bound on system unreliability most strongly were identified. It was assumed that these components would also contribute strongly to the true unreliability of the system . Steps were then taken to improve the reliability of the function served by these components. In most cases additional cross-strapping was used to add redundancy.

The major result of the design improvement process was to cross strap top and bottom power channels. This yields additional redundancy without adding components to the system. Furthermore, the four central transformers in figure 6.11 were used to add some redundancy to the pump motor controllers.

In order to get a feel for the effect of these changes on system reliability, the exact system was described in CAME and a Markov model for this system was constructed. A four failure level model bounded system unreliability between 0.042 and 0.122. While this is an improvement over the baseline (table 6.1), these bounds do not allow determination of the acceptability of the reliability of this improved system. An attempt was made to improve the accuracy of this reliability estimate by extending the Markov model to the fifth failure level. Here again the machine ran out of memory before the model was completed.

6.3.4 Hierarchical Analysis of Improved SSF Configuration

In order to obtain tighter bounds on system unreliability, a hierarchical analysis was applied to the redesigned system referred to in §6.3.3. Unlike the baseline configuration for this system, the top and bottom power channels could no longer be replaced by their corresponding effective components. In this redesigned system the top and bottom power channels depend on components outside of the original power subsystem (i.e., the central transformers). This introduces a channelized dependence which violates the guidelines of chapter 3.

Notice that the remainder of the exact system depends on the power subsystem only through these transformers. Thus, one way to decompose the redesigned exact system is to incorporate the transformers into the power subsystem. This enlarged power subsystem can be replaced with four effective power components which reflect the functional status of the four transformers.

Since the hierarchical analysis described above uses a different set of effective components, a new Markov model for the approximate subsystem model had to be produced and the resulting set of effective transition rates had to be re-derived. The Markov model for the approximate subsystem is shown in figure 6.14. In developing this Markov model no assumptions have been made about component symmetry. However, the sequence of effective component failures is known to be of no importance to the remainder of the exact system, so aggregation is used that removes this level of detail and reduces the number of effective transition rates to be calculated.

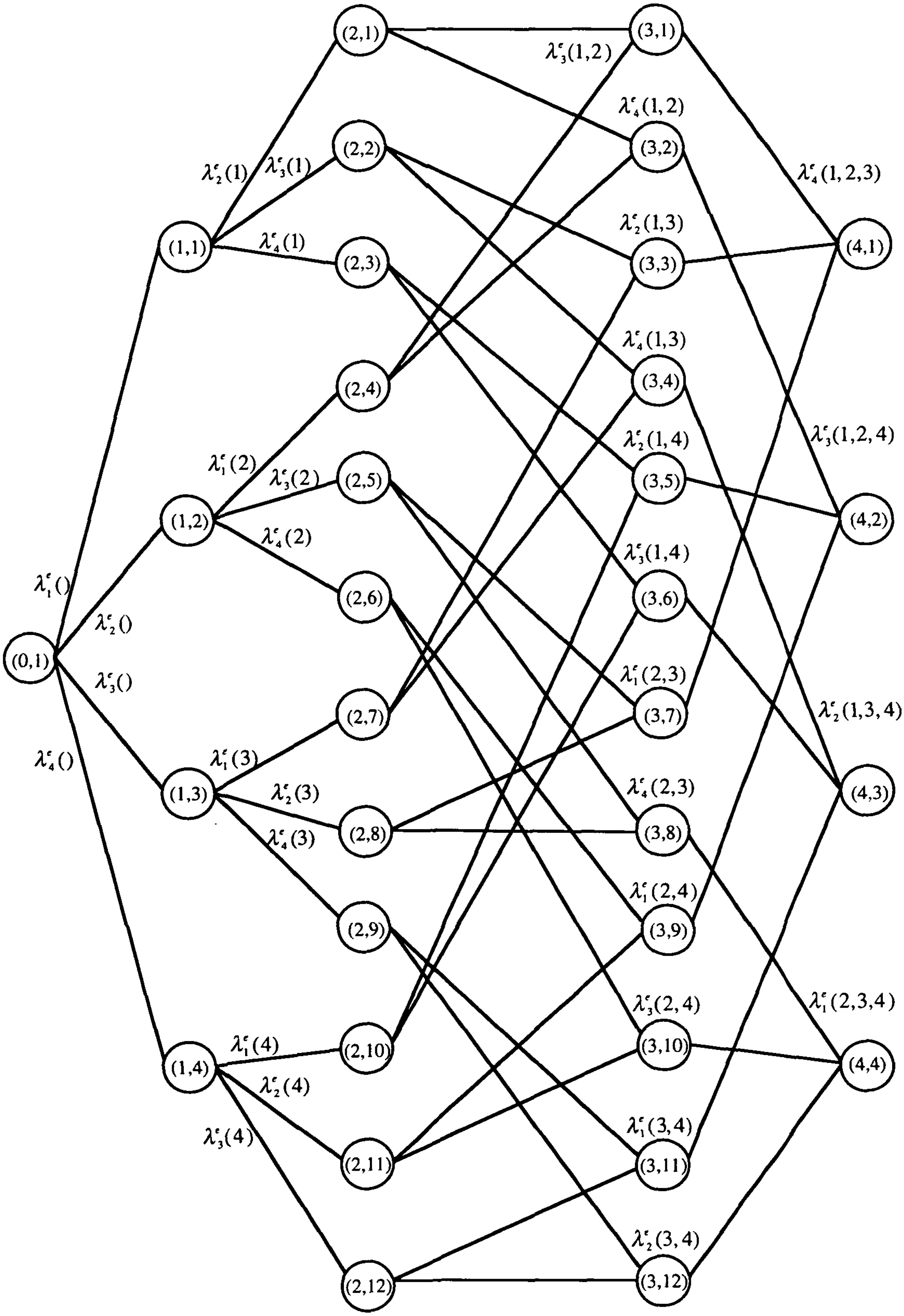


Figure 6.14: Approximate Power Subsystem Markov Model - Redesign

The exact subsystem description was entered into CAME and performance levels were defined for all of the states indicated in figure 6.14. This system description was used to construct a Markov model for the exact subsystem. The model was truncated at the seventeenth failure level. Although not exact, this model provided very accurate bounds on the final state probabilities.

The Markov model for the exact subsystem revealed that several of the states in figure 6.14 contained zero probability (i.e., several of the states in figure 6.14 represent unreachable states within the exact subsystem). All of the states in the exact subsystem model which correspond to having effective power one available and effective power three unavailable contained zero probability. Similarly, there was no probability associated with having effective power two available and effective power four failed. Physically, this means that power channel one cannot function if power channel three is not available, and power channel two cannot function if power channel four is not available. The revised Markov model of figure 6.15 incorporates this information and aggregates all of the system loss states into one. The revised set of unknown transition rates is also indicated in figure 6.15.

In the Markov model of figure 6.15, all of the λ^e*(to-SL)* are the same value, and the distinct system loss (SL) states are introduced simply to follow the failure level convention. All SL states in figure 6.15 correspond to only one state equation for the purpose of calculating effective failure rates.

We conclude the following about this phase of the hierarchical analysis:

1. The four effective components in our final approximate subsystem could have created as many as thirty two unknown effective transition rates.

2. The actual number of unknown effective transition rates is substantially less than the total possible number, due to the unreachability of certain exact subsystem states.

Since the Markov model for the exact subsystem contains a truncation state, the probability for this state is used to obtain optimistic and pessimistic values for the unknown transition rates. By adding the truncating state probability to the system loss state of figure 6.15 this produces a set of effective failure rates which will drive the power subsystem to failure somewhat prematurely. By adding the truncating state probability to the initial state of figure 6.15 this produces a set of effective failure rates which will drive the power subsystem to failure too slowly.

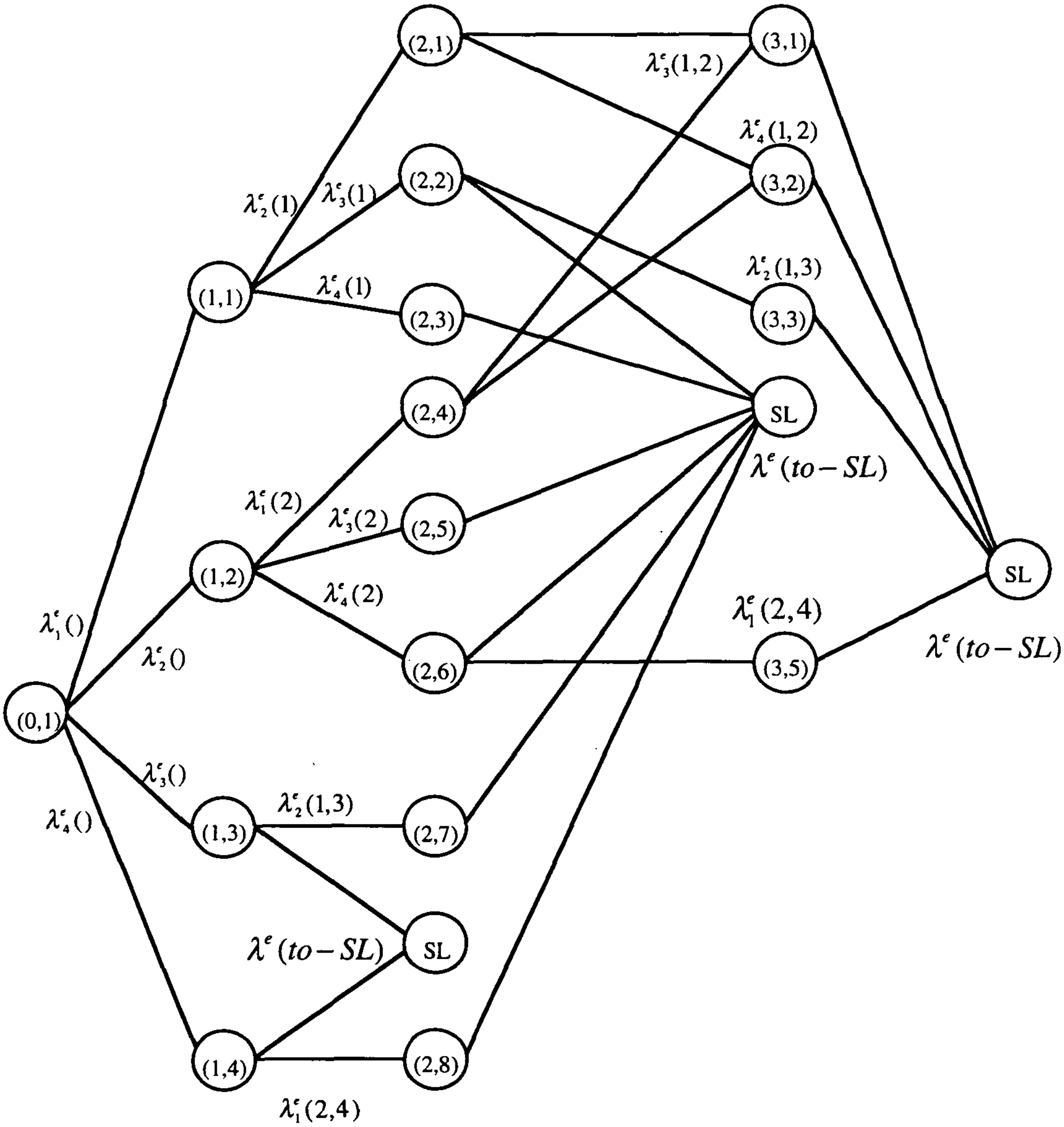


Figure 6.15: Revised Approximate Power Subsystem Markov model

The state probabilities of figure 6.15 were used to solve for the unknown transition rates and these rates were used in the approximate system model to define the appropriate effective power units. This approximate system was used to estimate the reliability of the exact system. The results of this analysis are compared to the original exact system evaluation in table 6.2.

System	Failure Levels	Time(hours)	Bounds
Exact	4	4.10	0.042 - 0.122
Exact	5	N/A	N/A
Approximate	5	6.64	0.061 - 0.075

Table 6.2: Hierarchical vs. Traditional Analysis - Final Architecture

6.4 Conclusions

In this chapter the hierarchical analysis technique developed in the previous chapters is used in the analysis of some small sample systems as well as a more realistic example (Space Station Freedom). The quantitative benefits of this technique have been demonstrated in the analysis of space station freedom. This technique has been used to improve the resolution in system reliability estimates by as much as a factor of six (see table 6.1).

Many systems could not previously be evaluated using Markovian analysis due to their size. Because of the improvements in resolution afforded by the hierarchical technique demonstrated here, many of these systems now can be evaluated using Markov models.

Chapter 7

Summary and Conclusions

7.1 Summary of Thesis

In chapter 2, a general outline was developed for a hierarchical technique which can be applied to systems which exhibit a simple dependence on a given subsystem. This special sort of dependence is used to replace the given subsystem with some smaller set of effective components. The effective components must be produced in such a way that this approximate system accurately reflects the reliability of the exact system.

Guidelines were established in chapter 3 to help guide the analyst in determining the validity of a given decomposition. These guidelines define the allowed interactions which may exist between the subsystem to be decomposed and the remainder of the exact system. Examples are also presented here to justify and explore the underlying principles of these guidelines.

In order for the approximate system to accurately reflect the reliability characteristics of the exact system, the approximate subsystem must accurately reflect the reliability characteristics of the exact subsystem. One method for producing such an approximate subsystem is to determine numerical values for reliability parameters of the effective components which make up this approximate subsystem. In chapter 4, Markov models were used to determine effective component failure rates for the case in which the exact subsystem is non-repairable and has perfect coverage. This yields effective failure rates which exhibit both time and state dependence.

The number of unknown transition rates which must be solved for in the approximate subsystem model grows rapidly with the number of effective components in the approximate subsystem. Some of these transition rates are zero due to the unreachability of certain states. In order to more easily discover zero transition rates, a particularly sparse Markov model was recommended. Also, truncation of the approximate subsystem Markov model was explored in order to help reduce the modeling complexity.

The technique of determining effective failure rates developed in chapter 4 is highly sensitive to roundoff and integration error. These issues were explored more fully in chapter 5. Integration based on the trapezoidal rule was recommended here as a result of numerical comparison with other integration techniques.

In chapter 6 two small systems were analyzed hierarchically as well as by direct analysis of the exact system. The hierarchical and traditional analyses were compared to gain some intuition about the accuracy of the hierarchical modeling technique. Finally, the Space Station Freedom (SSF) was used as an example of the application of this technique to a real, and previously intractable, problem. The real advantage of this technique was demonstrated in its capacity to dramatically improve the accuracy of reliability estimates for a given system.

7.2 Limitations and Suggestions for Further Work

The hierarchical modeling technique presented in this thesis was developed with a specific problem in mind: the reliability analysis of the attitude control function of the Space Station Freedom. Emphasis was placed on producing an engineering method for handling a real world problem. Most statements were supported by example and simple reasoning. A more rigorous proof of the accuracy of this analysis technique may yield more precise information about how to apply and extend this type of analysis.

The work in this thesis was developed for purely decaying subsystems with perfect coverage. The effect of imperfect coverage in most cases is to drive the system to failure with fewer component faults. For an exact subsystem with imperfect coverage, this corresponds to a common mode failure of certain subsystem channels. It is possible that imperfect coverage within the subsystem could be dealt with by simply ignoring coverage values, in the same way that common mode failures were ignored in §4.2.2 for the effective components. The resulting approximate subsystem should still accurately capture the probability of being in specific states of the exact subsystem.

Dealing with a repairable subsystem is not so straightforward, because the techniques developed for calculating effective failure rates cannot easily be extended to calculate effective repair rates. Work in this area may prove useful as most repairable systems tend not to permit much aggregation and truncation is not as effective as in purely decaying systems. Thus, repairable systems run into state space limitations for much smaller numbers of components in the system.

Finally, the real importance of the technique developed in this thesis is that it gives the analyst a methodology for examining different parts of a system separately. It is possible that many of the details of this technique could be performed differently. There may be a method for calculating effective failure rates which is less sensitive to roundoff and integration error than the method proposed. Indeed there may be ways for combining the reliability of different parts of the system without calculating effective failure rates. Perhaps this latter function could be performed using a decomposition based on the law of total probability.

References

Abraham, Gilley, Lee, and Rennels "A Numerical Technique for the Hierarchical Evaluation of Large, Closed Fault-Tolerant Systems," Dependable Computing for Critical Applications 2, J. F. Meyer and R. D. Schlichting, eds., Springer-Verlag, 1992, pp. 96-114

Babcock, Philip S., An Introduction to Reliability Modeling of Fault Tolerant Systems, Charles Stark Draper Laboratory Report R-1899, September 1986

Babcock, Philip S., The Symbolic Solution of Non-Cyclical Markov Models by Inspection, Charles Stark Draper Laboratory Report R-1954, April 1987

Bazovsky, Igor, Reliability: Theory and Practice, Prentice-Hall Space Technology Series, Englewood Cliffs, New Jersey, 1961

Bulirsch, and Stoer, Introduction to Numerical Analysis, Springer-Verlag, New York, New York, 1980

Goyal, and Lavenberg, "Modeling and Analysis of Computer System Availability", IBM Journal of Research and Development, 31, 6: 651-664. 1987

Hutchins, Babcock, and Rosch, An Introduction to the CAME Program Via Example, Charles Stark Draper Laboratory Report R-2082, July 1988

McCarragher, Brenan J., The Propagation of Errors in the Numerical Solution of Markov Models, Charles Stark Draper Laboratory Report T-1021, May 1989

Moler, and van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix Exponential", SIAM Review, Volume 20, Number 4, October 1978

Motyka, Megna, Schor, and Zinchuk, The Definition and Evaluation of an Integrated Flight and Propulsion Control System Architecture for an ASTOVL Aircraft, Charles Stark Draper Laboratory Report R-2227, February 1990

Nicole, Palumbo, and Rifkin, User's Guide to the Reliability Estimation System Testbed (REST), NASA Technical Memorandum 107596, June 1992

Stewart, James, Calculus, Brooks/Cole Publishing Company, Monterrey California, 1987

Strang, Gilbert, Linear Algebra and its Applications, 3rd ed., Harcourt Brace Jovanovich, Inc., San Diego, California, 1988

Zinchuk, Babcock, "Fault-Tolerant Design Optimization: Application to an Underwater Vehicle Navigation System", Symposium on Autonomous Underwater Vehicles Proceedings, IEEE/OES, Washington D.C., June 1990