INFLUENCE DIAGRAM MODELS WITH CONTINUOUS VARIABLES

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ENGINEERING-ECONOMIC SYSTEMS

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

C. Robert Kenley

June 1986

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies & Research

INFLUENCE DIAGRAM MODELS WITH CONTINUOUS VARIABLES

ABSTRACT

In theory, influence diagrams are valid representations of both continuous and discrete variables for probabilistic and decision analysis; however, current implementations of influence diagrams represent only discrete variables. This dissertation develops influence diagram models for continuous-variable decision and inference problems.

A normal influence diagram with complete procedures for decision analysis is developed for the linear-quadraticGaussian decision problem. This includes algorithms for assessment of probabilistic and deterministic parameters of the model, probabilistic analysis, and decision analysis. The matrix algebra used in traditional representation and solution techniques is replaced with operations based on graphical manipulation of the influence diagram. Potential areas for application include Bayesian linear regression, business portfolio analysis, forecasting, causal modelling, path analysis, discrete-time filtering, and proximal decision analysis.

Discrete-time filtering models are represented using the normal influence diagram model. Algorithms are developed for the measurement and time updates that constitute discrete-time filtering. Special cases treated are colored process noise, noninformative prior distributions, and heuristic consider filters for bias modelling. Operation counts for influence diagram filtering and other filter implementations show that the influence diagram approach compares favorably with efficient, stable

techniques developed since Kalman's original work. For some cases, the influence diagram approach is the most efficient.

Normal influence diagram representation and operations are shown to be valid for decision problems with quadratic value functions and non-Gaussian state variables as long as these two conditions hold:

(1) state variables are not dependent on decisions, and

(2) observation of a state variable does not permit inference about unobserved state variables in the decision network.

Assuming both conditions hold, the normal influence diagram is extended to proximal decision analysis.

ACKNOWLEDGEMENTS

Ross Shachter, my principal advisor, suggested the topic of this dissertation. Without his suggestion, I would still be burdened by a topic abundant with mathematical sophistication and sparse with insight. I count myself as one among many students who have benefited from his enthusiasm for teaching and research.

Ron Howard was a significant influence on my education. He taught me enduring principles to formulate and solve complex practical problems. Perhaps more importantly, he demonstrated that integrity is as important as intellect.

I would like to thank Jim Matheson for finding time in his busy schedule to serve on my reading committee. I will never forget the "Don't Panic!" button he wore on his lapel for my oral examination.

Alma Gaylord was a constant source of caring and kindness. Mako Bockholt always was cheerful, while promptly and accurately typing this dissertation in addition to her full-time duties. Pink Foster did a superb job on final editing, book makeup, and printing.

This research was supported by Independent Research and Development funds from the Astronautics Division of Lockheed Missiles & Space Company, Inc. In particular, Chuck MacQuiddy was generous enough to approve Lockheed sponsorship of my research. Tuition was provided through the Stanford Honors Cooperative program at Lockheed.

My parents, Howard and Marcia Kenley, provided love and support while allowing me the freedom to develop as an individual. Their conscientiousness and concern for my intellectual training and development of personal values is greatly appreciated. Financing my undergraduate work at MIT is just one example of my parents' concern, from which the foundation was developed to successfully pursue this Ph.D.

Special thanks go to Annie Hayter, my grandmother, who celebrates her eighty-fifth birthday this year. She has always been kind and generous, and serves as a model of courage and perseverance.

Despite her busy life while completing her Ph.D., my wife, Susan, has been a constant source of enthusiastic support and love. During our seven years of doctoral studies, she still found the time and energy to devote to strengthening our marriage. Her loving presence significantly enhances the joy of completing this dissertation.

TABLE OF CONTENTS

TABLE OF CONTENTS

(Continued)

TABLE OF CONTENTS

(Continued)

LIST OF FIGURES

LIST OF FIGURES

(Continued)

LIST OF FIGURES

(Continued)

LIST OF TABLES

INTRODUCTION

For decision analysis, influence diagrams have become an established tool for developing models and communicating among people [6]. They are graphical representations of decision making under uncertainty, which explicitly reveal probabilistic dependence and the flow of information. Recent research has developed techniques for evaluating an influence diagram and determining the optimal policy for its decisions [13], and for automated probabilistic inference using an influence diagram [12].

In theory, influence diagrams are valid representations of both continuous and discrete variables; however, current implementations of influence diagrams represent only discrete variables. The linear-quadratic-Gaussian model is the most widely used continuous-variable model for decision making and statistical inference. Examples include optimal control [3], multivariate linear regression [4], and path analysis [16]. Current models rely either on matrix theory (optimal control and regression) or graphical representation (path analysis). The matrix theoretic approach permits formulation and solution of diverse problems from a unified perspective, but it tends to construct a communication barrier between analysts and decision makers. The graphical approach of path analysis provides a communication tool for formulating statistical models and presenting results of data analysis, but its analytic use is limited to calculating correlation coefficients.

An influence diagram representation of the linear-quadratic-Gaussian decision and inference model is developed in Chapter 1 of this dissertation. Complete procedures for decision analysis are presented. This includes algoriths for assessment of probabilistic and deterministic parameters of the model, probabilistic analysis, and decision analysis. It has all the advantages of the genera? influence diagram [131. In addition, it fosters understanding of Gaussian processes by using the natural representation of correlation between random variables first suggested by Yule [17}. Also, we replace matrix algebra used in traditional representation and solution techniques with operations based on graphical manipulation. Potential areas for application include Bayesian linear regression, business portfolio analysis, forecasting, causal modelling, path analysis, discrete-time filtering, and proximal decision analysis.

In Chapter 2, we address discrete-time filtering. It is well known that Kalman's filter [7] can be derived by application of Bayes' rule to the measurement model of observations from a dynamic system [3, pp. 382-88J. A normal influence diagram model for discrete-time filtering is presented. Normal diagram scalar operations are extended to vector operations, in order to develop influence diagram algorithms for the measurement and time updates that constitute discrete-time filtering. The special cases of colored process noise and noninformative prior distributions are analyzed. Heuristic consider filters for bias modelling are explained using influence diagrams. We compare floating point operation counts for influence diagram filtering and other filter implementations studied by Bierman and Thornton [1, pp. 82-90; 15]. The influence diagram approach compares favorably with efficient, stable techniques developed since Kalman's original work, and for some cases, the influence diagram approach is the most efficient.

Chapter 3 addresses the validity of the linear-quadratic-Gaussian influence diagram model and algorithms when the assumption of Gaussian random variables does not hold. The key result of Chapter 3 is that normal influence diagram representation of the covariance matrix for the multivariate normal distribution is a valid factorization of the covariance matrix of any distribution. Normal influence diagram representation and operations are valid for decision problems with quadratic value functions and non-Gaussian state variables as long as these two conditions hold:

(1) state variables are not dependent upon decisions, and

(2) observation of a state variable does not permit inference about unobserved state variables in the decision network.

Chapter 4 applies the linear-quadratic influence diagram of Chapter 3 to Howard's proximal decision analysis model [5]. The entrepreneur's problem is presented using influence diagrams.

CHAPTER 1

THE NORMAL INFLUENCE DIAGRAM

1.1 Introduction

In this chapter, we present an influence diagram representation of the continuous-variable linear-quadratic-Gaussian decision problem. An assessment procedure is provided for encoding probabilistic information required for the normal influence diagram data structure, which guarantees a positive semi-definite covariance matrix. Algorithms are presented for probabilistic analysis and decision analysis using the influence diagram data structure. The relationship between influence diagram and covariance matrix representations of the multivariate normal distribution is also discussed.

1.2 Influence Diagrams and the Consultant's Problem

Let N be a set of integers $\{1,\ldots,n\}$ and s be an ordered sequence of distinct elements of N. Associated with each integer $j \in N$ is a variable X_j, and associated with each sequence s is a vector of variables $X_s = (X_{s_1},\ldots,X_{s_m})$. For each $j \in N$, either X_j is a random variable or a controllable decision variable. We define two disjoint sets, C the set of chance variables and D the set of decision variables, such that $N = C \cup D$.

An alternative representation for the joint probability distribution of X_N is a network structure called an influence diagram. This is built on an acyclic, directed graph with nodes N and arcs to node $j \in N$ from nodes corresponding to either variables that condition random variable X_j for $j \in C$, or information available at the time of decision X_j for $j \in D$. Conditioning nodes are given by the mapping $C(\cdot) : N \to 2^N$, so that X_j is conditioned by $X_{C(j)}$ for $j \in C$. Similarly, informational nodes for X_j are $X_{I(j)}$ for $j \in D$. Note that because the graph has no cycles, we can order the nodes such that

$$j \in C(k) \Rightarrow j < k \text{ for } k \in C, \text{ and}$$

$$j \in I(k) \Rightarrow j < k \text{ for } k \in D.$$

Given the network structure for the influence diagram, it is possible to describe the joint distribution of X_N by factoring it into conditional distributions. For $j \in C$,

$$P\{X_j \,|\, X_{\{1,\ldots,j-1\}} = x_{\{1,\ldots,j-1\}}\} = P\{X_j \,|\, X_{C(j)} = x_{C(j)}\}.$$

If $k \in \{1,\ldots,j-1\}$ is not in $C(j)$, X_j given $X_{C(j)}$ is conditionally independent of X_k, and no arc is constructed from X_k to X_j.

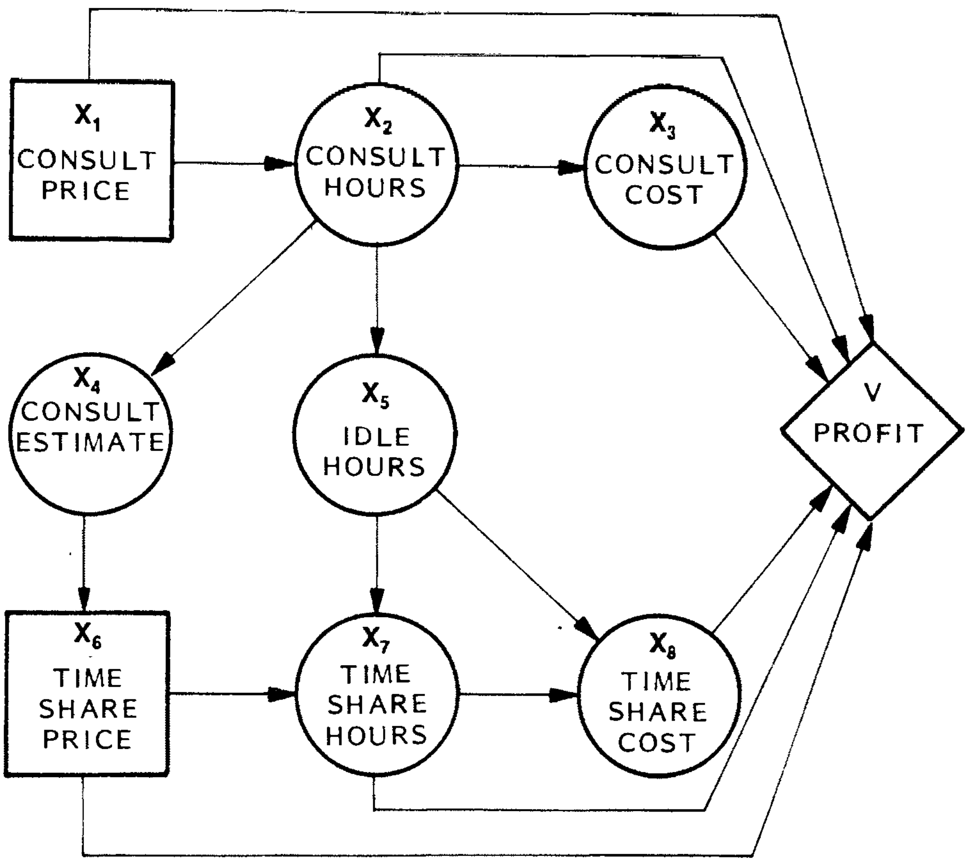
For decision problems, a special node called the value node V is needed. It represents the conditional expected value of a value function to be maximized by choice of X_D. An arc from a node $j \in N$ to V corresponds to dependency of the conditional expected value on X_j.

Figure 1-1 demonstrates use of an influence diagram for inference and decision making. A consultant owns a computer that is not fully utilized and has an opportunity to earn extra revenue with a time sharing service. Let

X_6 = time share price (an hourly rate),

X_7 = time share hours sold, and

X_8 = time share operating cost.



j	NAME	TYPE	C(j)	I(j)
1	CONSULT PRICE	DECISION	N/A	$\emptyset$
2	CONSULT HOURS	CHANCE	{1}	N/A
3	CONSULT COST	CHANCE	{2}	N/A
4	CONSULT ESTIMATE	CHANCE	{2}	N/A
5	IDLE HOURS	CHANCE	{2}	N/A
6	TIME SHARE PRICE	DECISION	N/A	{1, 4}
7	TIME SHARE HOURS	CHANCE	{5, 6}	N/A
8	TIME SHARE COST	CHANCE	{5, 7}	N/A
V	PROFIT	VALUE	{1, 2, 3, 6, 7, 8}	N/A

□ = DECISION NODE ○ = CHANCE NODE ◇ = VALUE NODE

Fig. 1-1 Influence Diagram for Consultant's Problem

Dependency of profit on these three variables is represented by arcs from them to the profit or value node. The relationships of hours sold with cost and of price with hours sold are shown by arcs from nodes X_7 to X_8 and from X_6 to X_7. Hours sold also depends on the number of idle hours (X_5) available for time share sale. A connection to the consulting line of business is provided by dependence of idle hours on consulting hours sold (X_2), since the consultant uses his computer for consulting services. Consulting hours sold also affects X_3, the cost of the consulting service. Prior to setting the time share price, an estimate of consulting hour sales (X_4) is available to the consultant. The hours sold depends on the price (X_1) for an hour of consulting service. The contribution of consulting price, hours sold, and cost to profit is shown by arcs from these nodes to the value node.

The lack of arcs between some pairs of nodes in the consultant's influence diagram represents an assertion regarding conditional independence. For instance, there is no arc from consulting price (X_1) to consulting hour sales estimate (X_4). This implies that, given actual consulting hours sold (X_2), the estimate (X_4) is independent of consulting price (X_1). This is because we believe the estimate is a good predictor of actual consulting hours sold, regardless of the price we select. This does not assert, however, independence of unconditional variables X_1 and X_4. In fact, both variables are likely to be dependent, because there is a directed path from X_1 to X_4, going through X_2.

1.3 Normal Model Definition

For the remainder of this chapter, we assume vector X_N has a multivariate normal distribution characterized by mean $\mu_N = E[X_N]$ and covariance matrix $\Sigma_{NN} = Var[X_N] = E[X_N X_N^T] - E[X_N]E[X_N^T]$. We will develop an influence diagram representation of X_N. There may be many influence diagrams corresponding to the same underlying joint distribution. This is because we have the freedom to permute the ordering of N, and there is a possibility that Σ_{NN} is singular.

1.3.1 Probabilistic Model

For the multivariate normal, the distribution of chance node X_j, given $X_{C(j)}$, is normal. Its expectation is a linear function of $X_{C(j)}$, and its variance is independent of $X_{C(j)}$ [4, pp. 108-111]. Let

$$b_{kj} = \begin{cases} \partial E[X_j | X_{C(j)} = x_{C(j)}]/\partial x_k, & \text{if } k \in C(j) \\ 0 & , \text{otherwise,} \end{cases}$$

and $v_j = Var[X_j | X_{C(j)} = x_{C(j)}]$. The variable $\{X_j | X_{C(j)} = x_{C(j)}\}$ is normally distributed with mean $(\mu_j + \sum_{k \in C(j)} b_{kj}(x_k - \mu_k))$ and variance v_j. If β is Yule's partial regression coefficient [17], the scalar

$$b_{kj} = \beta_{jk.\{1,\ldots,k-1,k+1,\ldots,j-1\}} = \beta_{jk.C(j)\setminus\{k\}},$$

where '\' represents set subtraction. Also, if $v_j = 0$, X_j is a deterministic, linear function of $X_{C(j)}$.

For decision node X_j, we let $b_{kj} = 0$ for all $k \in N$, and $v_j = 0$. Also, we let μ_j be a reference decision value for the decision variable.

In the context of decision analysis, μ_j could be the optimal X_j derived from a deterministic decision model.

1.3.2 Quadratic Value Function

We define the conditional expected value of the value function as

$$V(x_N) = \frac{1}{2} x_N^T Q x_N + p^T x_N + r,$$

where Q is a symmetric n×n matrix. The consultant's deterministic profit model in Fig. 1-1 is quadratic.

$$\text{Profit} = x_1 x_2 - x_3 + x_6 x_7 - x_8$$
$$= V(x_N) = \frac{1}{2} x_N^T Q x_N + p^T x_N + r,$$

where

$$Q = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$p = (0\ 0\ {-1}\ 0\ 0\ 0\ 0\ {-1})^T$, and

$r = 0.$

1.4 Probabilistic Assessment

The assessment of probabilistic information for our representation proceeds as follows:

(1) Assess an influence diagram

Determine the variables X_N

Determine the graph (C, D, C(j) for j ϵ C, I(j) for j ϵ D)

(2) Order the variables X_N so that

$$j \in C(k) \Rightarrow j < k, \quad \text{for} \quad k \in C$$

$$j \in I(k) \Rightarrow j < k, \quad \text{for} \quad k \in D$$

(3) Do j = 1 to n

If $(j \in C)$

$\mu_j = E[X_j | X_{C(j)} = \mu_{C(j)}]$

$v_j = \mathrm{Var}[X_j | X_{C(j)} = \mu_{C(j)}]$

For $i \in C(j)$

$$b_{ij} = \frac{\partial E[X_j | X_{C(j)\backslash\{i\}} = x_{C(j)\backslash\{i\}}, X_i = x_i]}{\partial x_i}$$

End

Else $(j \in D)$

μ_j = reference decision value

$v_j = 0$

End If

End Do

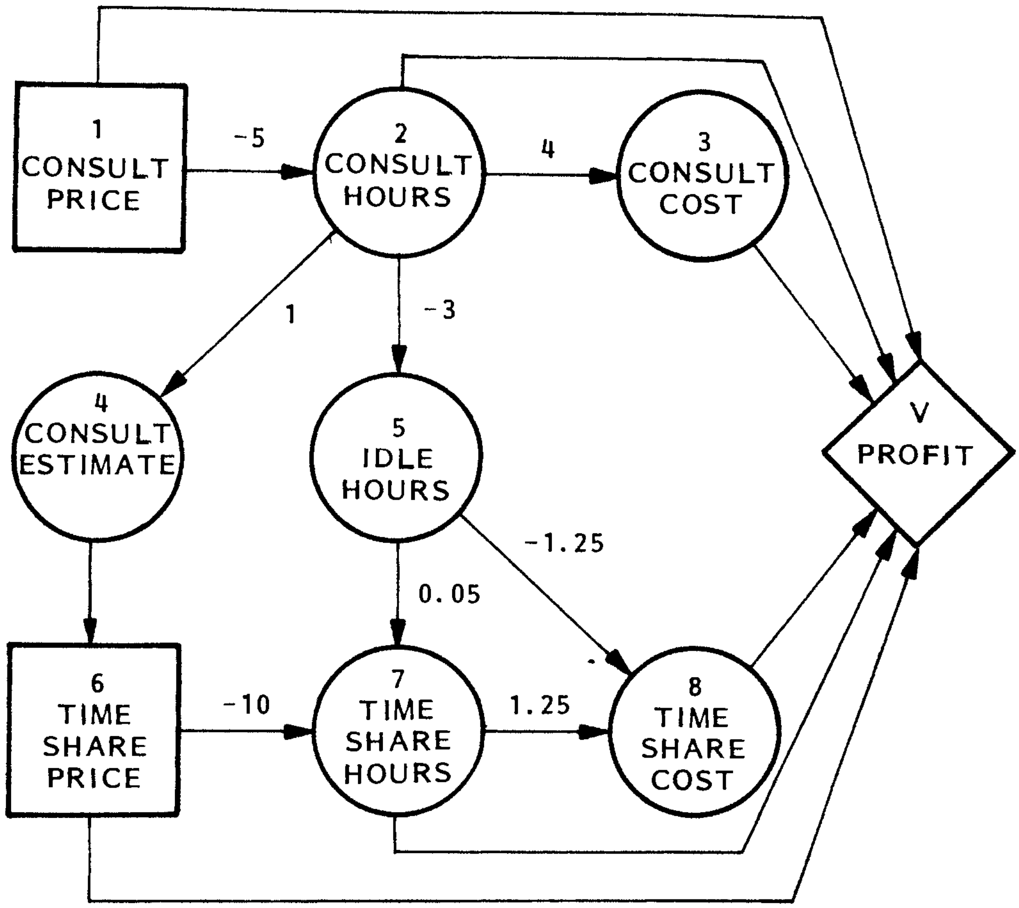
The assessment is simple and consistent, producing a positive semi-definite (PSD) covariance matrix, as long as v_N is non-negative (see Appendix 1.A for proof). This is a significant improvement over Oren's procedure [9], where a non-PSD covariance arises from assessing unconditional variances. In order to produce a PSD covariance matrix, Oren must use a complex algorithm based on conjugate gradients. For our procedure, it is only necessary to assess non-negative conditional variances.

In addition to simplicity and consistency, assessing conditional distributions takes advantage of the structure of the model and permits us to ask for information that is more accurate and readily assessed. For example, assessing the conditional variance of the consultant's time share cost, given an average value for idle hours and time share hours, is preferred to assessing the unconditional variance of the time share cost.

A normal influence diagram representation of the consultant's decision is shown in Fig. 1-2. The nonzero elements of $B = [b_{ij}]$ are placed on arcs between nodes. Reference values for the decision variables are 100 for μ_6, the consulting price, and 25 for μ_1, the time share price. The consulting estimate is unbiased, with mean equal to the consulting sales mean. The estimate is inaccurate, having a conditional standard deviation of 500 hours, given actual consulting hours.

1.5 Probabilistic Analysis

Once we have the distribution represented as an influence diagram, we can manipulate it directly. In this section, we consider integration of the joint distribution with respect to a variable in the model, which we shall call removing a chance node. Another manipulation to the diagram is reversal of an arc between two chance nodes, the influence diagram form of Bayes' rule. Reversal is used in decision problems for inference about unobserved variables, given observations of dependent variables.



$$v = (0\ \ 40{,}000\ \ 4{,}000{,}000\ \ 250{,}000\ \ 100\ \ 0\ \ 10{,}000\ \ 40{,}000)^T$$

$$\mu = (100\ \ 1{,}500\ \ 58{,}000\ \ 1{,}500\ \ 3{,}500\ \ 25\ \ 750\ \ 5{,}000)^T$$

$$Q = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad p = (0\ \ 0\ \ -1\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1)^T$$

$$r = 0$$

Fig. 1-2 Consultant's Normal Influence Diagram

1.5.1 Removing a Chance Node

Suppose chance node i has a single direct successor chance node j. In terms of $C(\cdot)$, $i \in C(j)$, but i is not an element of $C(N\backslash\{j\})$. Operations to remove node i into node j are as follows:

$$C(j) \leftarrow (C(j) \cup C(i))\backslash\{i\}$$
$$b_{kj} \leftarrow b_{kj} + b_{ki}b_{ij} \text{ for } k \in C(j)$$
$$v_j \leftarrow v_j + b_{ij}^2 v_i$$
$$N \leftarrow N\backslash\{i\}.$$

A proof for this operation appears in Appendix 1.B.

1.5.2 Reversing an Arc Between Chance Nodes

Suppose chance node i is a conditioning variable for chance node j, but there is no other directed path in the network from i to j. If there were another directed path, reversing an arc would create a cycle. In terms of $C(\cdot)$, we require $i \in C(j)$, and

$$\forall s = \{s_1,\ldots,s_m\} \subset \{i+1,\ldots,j-1\},\ b_{is_1}b_{s_1s_2}\cdots b_{s_{m-1}s_m}b_{s_mj} = 0.$$

The reversal of the arc proceeds in two steps. The first step, similar to removal of node i into node j, consists of:

$$C(j) \leftarrow (C(j) \cup C(i))\backslash\{i\}$$
$$b_{kj} \leftarrow b_{kj} + b_{ki}b_{ij} \text{ for } k \in C(j)$$
$$v_j^{old} \leftarrow v_j$$
$$v_j \leftarrow v_j + b_{ij}^2 v_i$$
$$N \leftarrow N\backslash\{i\}.$$

At this point the new X_j is conditionally independent of X_i, given the new $X_{C(j)}$.

If the new v_j is zero, X_i and X_j are independent, and it is unnecessary to construct an arc from j to i. Otherwise we must proceed to a second step as follows:

If $(v_j > 0)$

$C(i) \leftarrow C(j) \cup \{j\}$

$v_{ratio} \leftarrow v_i/v_j$

$v_i \leftarrow v_j^{old} v_{ratio}$

$b_{ji} \leftarrow b_{ij} v_{ratio}$

$b_{ki} \leftarrow b_{ki} - b_{kj} b_{ji}$ for $k \in C(j)$

End If.

Proof for this operation appears in Appendix 1.B.

The second step updates v_i via a quotient and product of non-negative numbers. Thus, roundoff errors cannot produce negative conditional variances. Traditional formulas for applying Bayes' rule to the multivariate normal distribution can produce negative variances from roundoff errors [1]. This makes influence diagram processing attractive for real-time decision systems, where handling negative variances can be costly.

1.5.3 Use of Reversal During Assessment

Figure 1-3 depicts the information needed to assess μ_j, v_j, and b_{ij} for conditioning node i. We could assess b_{ij} by first asking the decision maker or a designated expert to give us $\mu_{j|i}(\mu_i+\Delta)$, the expected

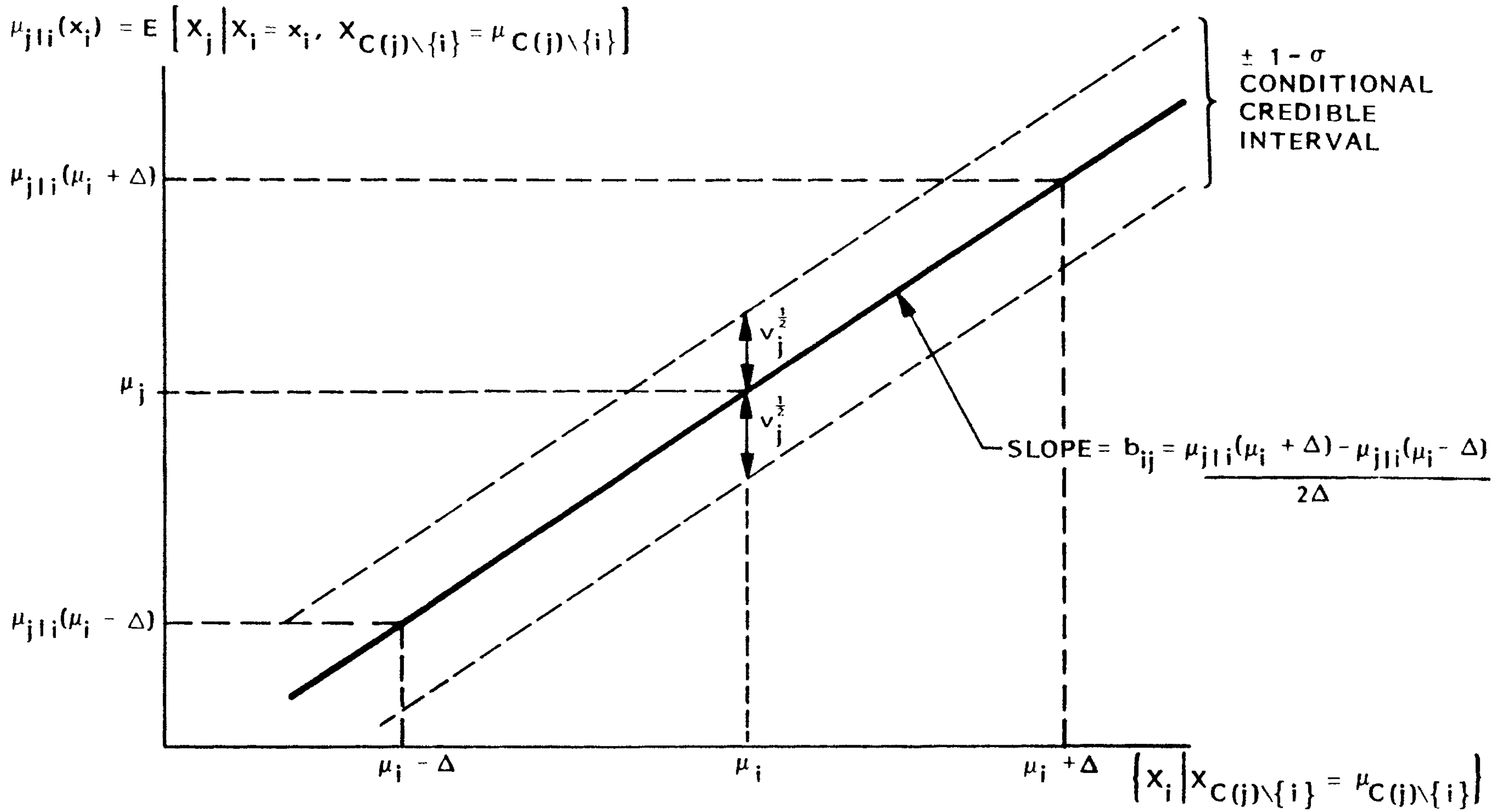


Fig. 1-3 Assessment

value of X_j, given X_i is $\mu_i+\Delta$ and $X_{C(j)\setminus\{i\}}$ is $\mu_{C(j)\setminus\{i\}}$. We also would ask for $\mu_{j|i}(\mu_i-\Delta)$, so that

$$b_{ij} = [\mu_{j|i}(\mu_i+\Delta) - \mu_{j|i}(\mu_i-\Delta)]/(2\Delta).$$

The constant Δ should be selected so that it represents a valid deviation of X_i from μ_i, given that all other conditioning variables of X_j are at their means. A reasonable choice of Δ is three times the conditional standard deviation $(\mathrm{Var}[X_i|X_{C(j)\setminus\{i\}} = \mu_{C(j)\setminus\{i\}}])^{1/2}$. To calculate the needed conditional variances for assessment, we can perform reversals using the influence diagram of the conditioning variables of X_j. Figure 1-4 demonstrates this for a trivariate distribution. The initial diagram available following assessment of X_1 and $\{X_2|X_1\}$ already has $\mathrm{Var}[X_2|X_1] = 1$. To assess b_{23}, we let $\Delta = 3$. Reversing X_1 and X_2, we have $\mathrm{Var}[X_1|X_2] = 1/2$. For assessment of b_{13}, we let $\Delta = 3/\sqrt{2}$ instead of the unconditional "three-sigma" value of 3 from the initial diagram.

1.6 Decision Analysis

For decision making, our criterion is to select X_j as the function of $X_{I(j)}$ that maximizes $V(x_{I(j)\cup\{j\}})$, the expected value of the value function conditioned on $X_{I(j)\cup\{j\}}$. Prior to selecting X_j, the diagram may contain chance nodes not in $I(j)$ and having an effect on the value function. We must remove these nodes by expectation into the value node before X_j can be selected. Similarly, decisions occurring after X_j must be removed by maximization into the value node.

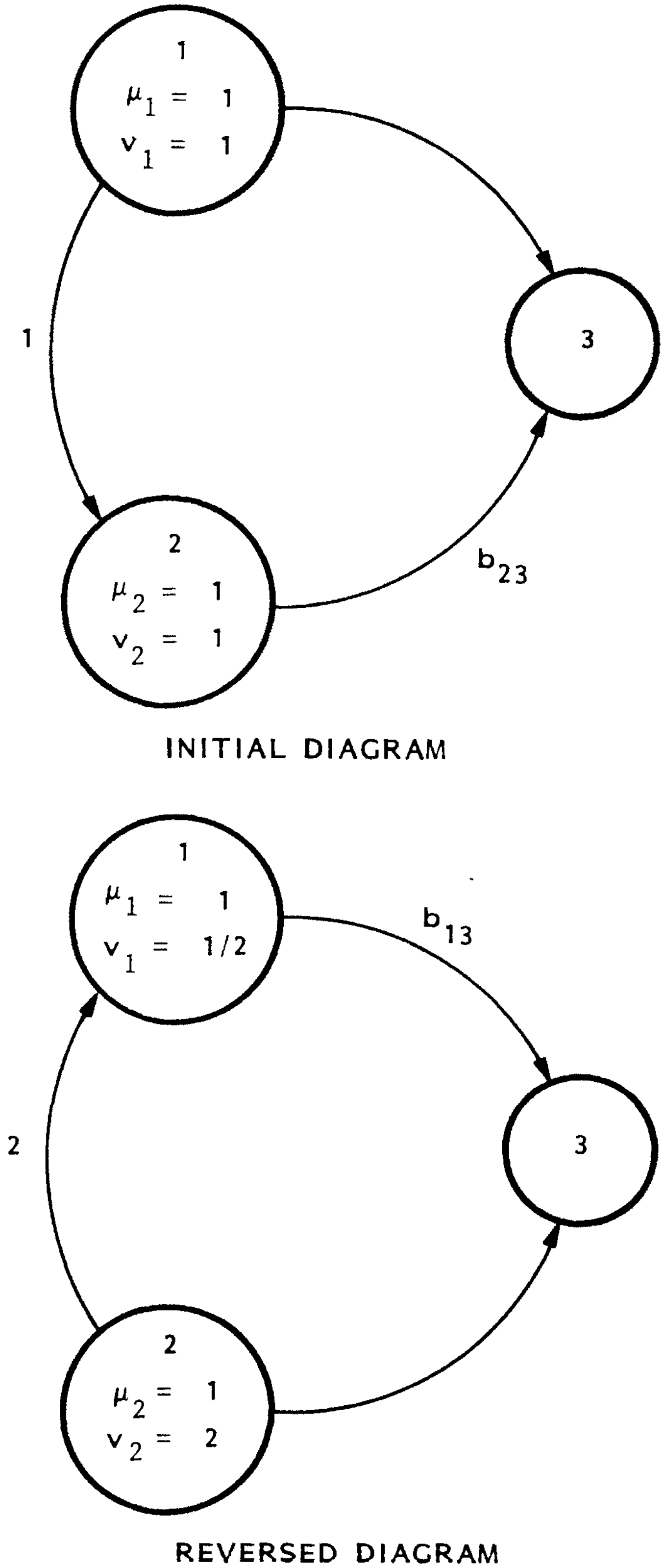


Fig. 1-4 Using Reversal During Assessment

1.6.1 Removing a Chance Node

Let d be a sequence corresponding to some of the variables, and let $s = N \backslash d$. We can partition Q into

$$\begin{bmatrix} Q_{dd} & Q_{ds} \\ Q_{sd} & Q_{ss} \end{bmatrix},$$

and p^T into (p_d^T, p_s^T). Suppose that we wish to take the expectation of $V(x_d, x_s)$ with respect to X_d. In a decision context, these would be random variables, that have no successors outside d and are not observed before any of the decisions must be made. The transformations to remove X_d into V are:

$$Q_{ss} \leftarrow Q_{ss} + Q_{sd}B_{sd}^T + B_{sd}Q_{ds} + B_{sd}Q_{dd}B_{sd}^T$$

$$p_s \leftarrow p_s + B_{sd}p_d + (Q_{sd} + B_{sd}Q_{dd})(\mu_d - B_{sd}^T\mu_s)$$

$$r \leftarrow r + \frac{1}{2}\,\mathrm{trace}(Q_{dd}\ \mathrm{Var}[X_d|X_s])$$

$$+ \frac{1}{2}\,(\mu_d - B_{sd}^T\mu_s)^T Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d^T(\mu_d - B_{sd}^T\mu_s)$$

$$N \leftarrow N \backslash d.$$

If d is a scalar variable, the update of r becomes

$$r \leftarrow r + \frac{1}{2}\,Q_{dd}v_d + \frac{1}{2}(\mu_d - B_{sd}^T\mu_s)^2 Q_{dd} + p_d(\mu_d - B_{sd}^T\mu_s).$$

A proof for this operation is in Appendix 1.C.

1.6.2 Removing Decision Nodes

Suppose that we wish to optimally choose X_d. In a decision context, these would be variables with no successors outside d and under decision maker control, which we may select after observing X_s. Assume Q_{dd} is negative definite and symmetric, so that an optimal choice exists and is unique. The preferred decision is

$$x_d^*(x_s) = -Q_{dd}^{-1}p_d - Q_{dd}^{-1}Q_{ds}x_s \ .$$

The transformations to remove X_d into V are:

$$Q_{ss} \leftarrow Q_{ss} - Q_{sd}Q_{dd}^{-1}Q_{ds}$$

$$p_s \leftarrow p_s - Q_{sd}Q_{dd}^{-1}p_d$$

$$r \leftarrow r - \frac{1}{2}\, p_d^T Q_{dd}^{-1} p_d$$

$$N \leftarrow N \backslash d.$$

A proof for this operation is in Appendix 1.C.

1.7 Solution of the Consultant's Problem

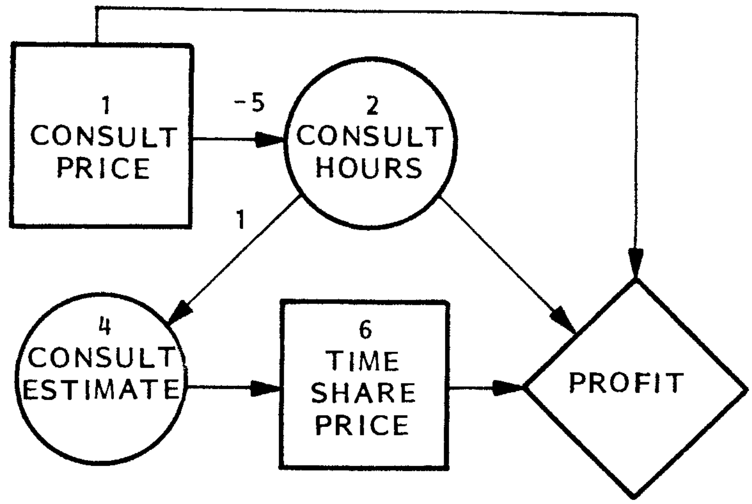
Solving the influence diagram (Fig. 1-2) for the consultant's optimal decision policy proceeds as follows:

(1) Successively remove chance nodes X_8, X_7, X_5, and X_3 into the value node

(2) Reverse the arc from X_2 to X_4 to perform inference about X_2 given X_4 (Fig. 1-5)

(3) Remove chance node X_2 into the value node to reduce the inference to the preposterior distribution of X_4 (Fig. 1-6)

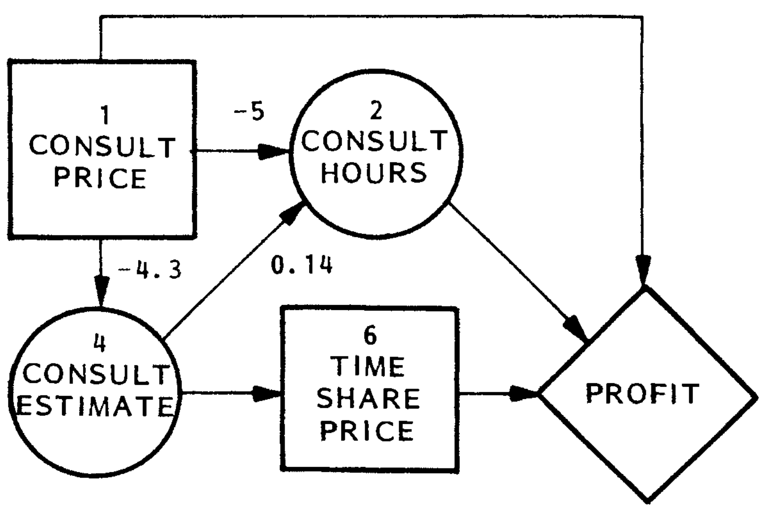
$N = \{1,2,4,6\}$



$V = (0\ \ 40{,}000\ \ 250{,}000\ \ 0)^T$

BEFORE REVERSAL

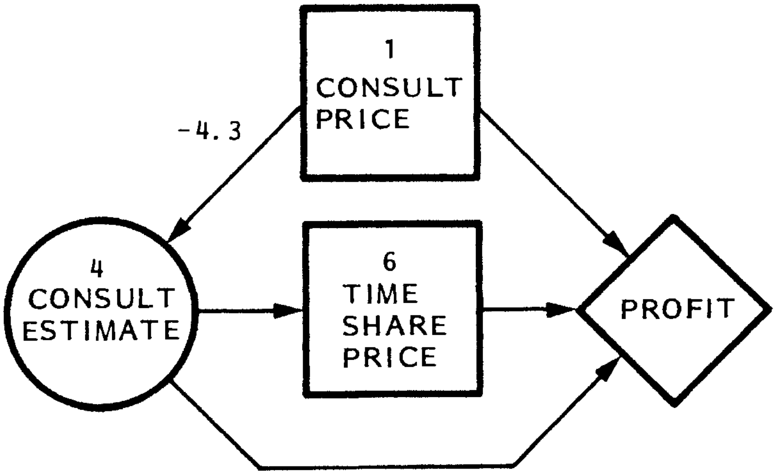
$N = \{1,4,2,6\}$



$v = (0\ \ 290{,}000\ \ 34{,}482.76\ \ 0)^T$

AFTER REVERSAL

Fig. 1-5 Reversal Step



$$N = \{1,\ 4,\ 6\}$$

$$Q = \begin{bmatrix} -8.6207 & 0.1379 & 0.6466 \\ 0.1379 & 0.0 & -0.0207 \\ 0.6466 & -0.0207 & -20.0 \end{bmatrix}$$

$$p = (1756.73\ \ -1.043\ \ 978.9)^T$$

$$r = -65{,}007.54$$

$$v = (0\ \ 34{,}482.76\ \ 0)^T$$

Fig. 1-6 Minimal Representation of Consultant's Problem

(4) Remove decision node X_6 into the value node (Fig. 1-7)

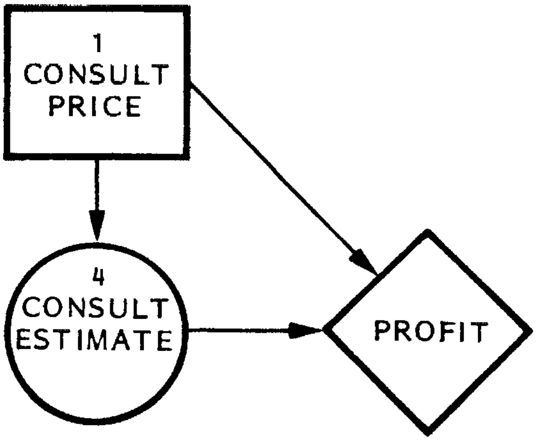
(5) Remove chance node X_4 into the value node (Fig. 1-7)

(6) Remove decision node X_1 into the value node (Fig. 1-7).

At any stage in the removal process, the expected value of the profit lottery, given the current reference values of the decision variables, is calculated by removing all remaining nodes into the value node as chance nodes. The value of r after completion of removals is the expected value of the profit lottery. Removing all nodes in the reference diagram (Fig. 1-2) as chance nodes, the value of r is \$105,750, the expected value of the reference decisions. If the consulting price decision is not changed from the reference of 100, an optimally selected time share price increases the expected profit to \$112,319.51. Optimal selection of the consulting price (Fig. 1-7) further increases the expected profit to \$170,347.98.

Figure 1-6 is a reduction of the consultant's problem to the minimal information necessary for selection of optimal decisions. The variables remaining in the diagram are the decisions and variables observed prior to making at least one of the decisions. This is the first step in the solution process that matrix Q is of full rank and the value node has a direct dependence on the consulting sales estimate.

In Fig. 1-7, the optimal time share price is a fixed value plus a linear correction for the consulting price decision and consulting sales estimate. This is a prescription for optimal action given knowledge of the past and the consequences of the decision on future events, even if the consulting price decision were not made optimally.



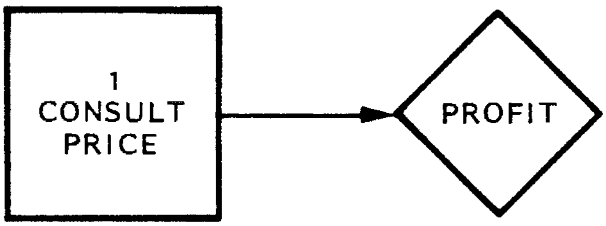
$N = \{1, 4\}$

$$Q = \begin{bmatrix} -8.5998 & 0.1373 \\ 0.1373 & 2.1403 \times 10^{-5} \end{bmatrix}$$

$p = (1988.38 \ -2.506)^T$

$r = -41{,}052.43$

$$x_6^* = 48.944 + 0.3233\, x_1 - 1.0345 \times 10^{-3}\, x_4$$



$N = \{1\}$

$Q = \begin{bmatrix} -9.9719 \end{bmatrix}$

$p = (2072.97)$

$r = -50{,}207.58$

$x_1^* = 207.88$

$r = 170{,}347.98$

Fig. 1-7 Optimal Solution of Consultant's Problem

After determining the optimal policies, we construct the consultant's policy diagram (Fig. 1-8), which converts decisions to deterministic policy nodes. We update the reference values of the policies and unconditional means of chance nodes affected by changes in decision variables. The reference value of each policy is the optimal decision setting, given that its policy predecessors are at their reference value and its chance predecessors are at their reference mean. Also, we place the linear sensitivity coefficients on arcs from informational predecessors to each policy node.

1.8 Relationship to Covariance Representation

Figure 1-9 demonstrates the relationship of covariances of a nonsingular trivariate normal distribution to conditional variances and arc coefficients of an influence diagram representation. Appendix 1.A contains detailed proofs about the relationship of the influence diagram representation to the covariance representation for arbritrary dimension and rank of the covariance matrix.

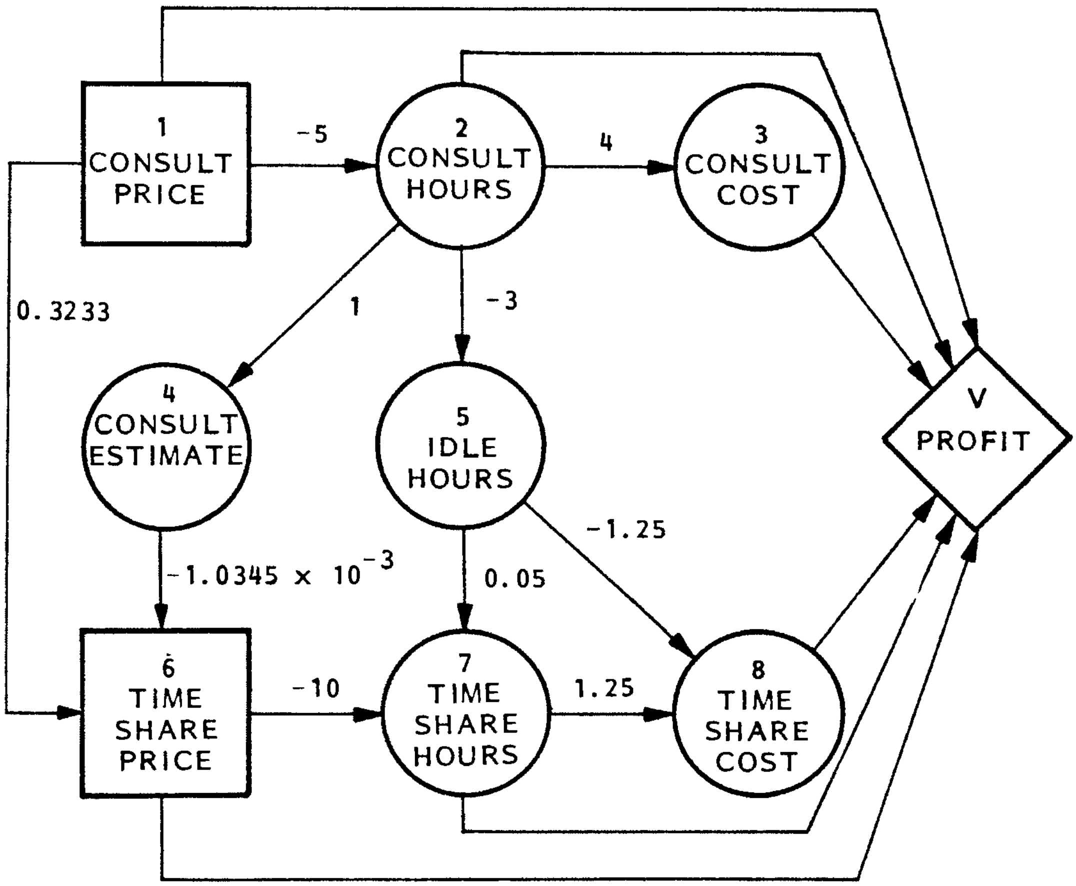
1.8.1 Constructing a Covariance Matrix from an Influence Diagram

A covariance matrix for X_N can be constructed from an influence diagram; it will be positive semi-definite. The results of Appendix 1.A are incorporated into an algorithm as follows:

(1) Order the variables in the influence diagram so that

$j \in C(k) \Rightarrow j < k$

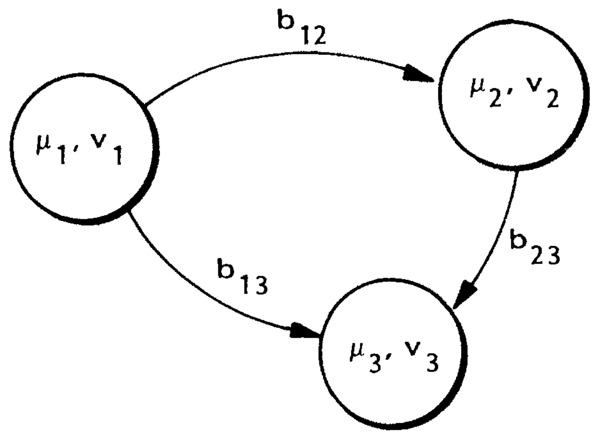
(2) $\sigma_{11} \leftarrow v_1$



$$v = (0\ \ 40{,}000\ \ 4{,}000{,}000\ \ 250{,}000\ \ 100\ \ 0\ \ 10{,}000\ \ 40{,}000)^T$$

$$\mu = (207.88\ \ 960.59\ \ 55842\ \ 960.59\ \ 5118.2\ \ 54.67\ \ 543.21\ \ 2707.48)^T$$

Fig. 1-8 Consultant's Policy Diagram



j	$C(j)$	$E\left[X_j \mid X_{c(j)} = x_{c(j)}\right]$	$VAR\left[X_j \mid X_{c(j)} = x_{c(j)}\right]$	v_j	b_{1j}	b_{2j}
1	ϕ	μ_1	v_1	σ_{11}	N/A	N/A
2	$\{1\}$	$\mu_2 + b_{12}(X_1 - \mu_1)$	v_2	$\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}}$	$\frac{\sigma_{12}}{\sigma_{11}}$	N/A
3	$\{1,2\}$	$\mu_3 + b_{13}(X_1 - \mu_1) + b_{23}(X_2 - \mu_2)$	v_3	$\sigma_{33} - \frac{\sigma_{22}\sigma_{13}^2 - 2\sigma_{12}\sigma_{23}\sigma_{13} + \sigma_{11}\sigma_{23}^2}{\sigma_{11}\sigma_{22} - \sigma_{12}^2}$	$\frac{\sigma_{22}\sigma_{13} - \sigma_{12}\sigma_{23}}{\sigma_{11}\sigma_{22} - \sigma_{12}^2}$	$\frac{\sigma_{11}\sigma_{23} - \sigma_{12}\sigma_{13}}{\sigma_{11}\sigma_{22} - \sigma_{12}^2}$

Fig. 1-9 Trivariate Normal Example

(3) Do $j = 2,n$

$s \leftarrow \{1,\ldots,j-1\}$

$\Sigma_{js} \leftarrow \Sigma_{ss}B_{sj}$

$\Sigma_{sj} \leftarrow \Sigma_{js}^{T}$

$\sigma_{jj} \leftarrow v_j + \Sigma_{js}B_{sj}$

End Do.

1.8.2 Constructing an Influence Diagram from a Covariance Matrix

Although an influence diagram corresponding to a particular multivariate normal distribution is not unique, this algorithm will determine a specific influence diagram:

(1) Order the variables in the model so that $j \geq k$ implies j is not in $C(k)$ after the diagram has been constructed

(2) $s \leftarrow \emptyset$

(3) Do $j = 1,n$

If $(s = \emptyset)$

$v_j \leftarrow \sigma_{jj}$

$C(j) \leftarrow \emptyset$

Else

$B_{sj} \leftarrow \Sigma_{ss}^{-1}\Sigma_{sj}$

$v_j \leftarrow \sigma_{jj} - \Sigma_{js}B_{sj}$

$C(j) \leftarrow \{i \in S : b_{ij} \neq 0\}$

End If

If $(v_j > 0)$ $s \leftarrow s \cup \{j\}$

End Do.

Afterwards, X_s explains all of the variance in the model, and Σ_{ss} is a maximal-dimension, linearly independent submatrix of the covariance matrix. Furthermore, the number of elements in s is equal to the rank of Σ_{NN}.

1.9 Conclusion

This chapter presented an influence diagram representation of the linear-quadratic-Gaussian decision problem. It has all the advantages of the general influence diagram [13]. In addition, it fosters understanding of Gaussian processes by using the natural representation of correlation between random variables first suggested by Yule [17]. Matrix algebra used in traditional representation and solution techniques has been removed. Potential areas for application include Bayesian linear regression, business portfolio analysis, forecasting, causal modelling, path analysis, discrete-time filtering, and proximal decision analysis. Discrete-time filtering is addressed in Chapter 2, and proximal decision analysis is addressed in Chapter 4.

APPENDIX 1.A

COVARIANCE REPRESENTATION PROOFS

This appendix presents the relationship between the influence diagram and covariance representations of the multivariate normal distribution. The matrix theoretic results are equivalent to Yule's results for partial regression coefficients [17].

Lemma 1

If $s = \{1,\ldots,j-1\}$ for some $j \in N$, then

$$\begin{bmatrix} \Sigma_{ss} & \Sigma_{sj} \\ \Sigma_{js} & \Sigma_{jj} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & v_j \end{bmatrix} + \begin{bmatrix} I_{j-1} \\ B_{sj}^T \end{bmatrix} \Sigma_{ss} [I_{j-1} \;\; B_{sj}] ,$$

where I_{j-1} is the $(j-1) \times (j-1)$ identity matrix.

Proof:

Let $t = \{1,\ldots,j\}$.

$$\begin{bmatrix} \Sigma_{ss} & \Sigma_{sj} \\ \Sigma_{js} & \Sigma_{jj} \end{bmatrix} = \Sigma_{tt} = \mathrm{Var}[X_t]$$

$$= E[\mathrm{Var}[X_t | X_s]] + \mathrm{Var}[E[X_t | X_s]]$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & v_j \end{bmatrix} + \mathrm{Var}\begin{bmatrix} X_s \\ \mu_j + B_{sj}^T(X_s - \mu_s) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 \\ 0 & v_j \end{bmatrix} + \begin{bmatrix} I_{j-1} \\ B_{sj}^T \end{bmatrix} \Sigma_{ss} [I_{j-1} \;\; B_{sj}] \qquad \square$$

Lemma 2

Let

$$S_j = \begin{bmatrix} \Sigma_{tt} & 0 & \cdots & 0 \\ 0 & v_{j+1} & \cdots & 0 \\ \vdots & & \ddots & v_n \end{bmatrix}, \quad \text{and}$$

$$U_j = \begin{bmatrix} I_{j-1} & B_{sj} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{n-j} \end{bmatrix}.$$

Then

$$S_j = U_{j-1}^T \; S_{j-1} \; U_{j-1} \; ,$$

and

$$\Sigma_{NN} = S_n = U_n^T \cdots U_1^T \; S_0 \; U_1 \cdots U_n \; .$$

Proof

Apply Lemma 1 and matrix multiplication to show $S_j = U_{j-1}^T \; S_{j-1} \; U_{j-1}$. By induction on j, $S_n = U_n^T \cdots U_1^T \; S_0 \; U_1 \cdots U_n$.

Theorem

Σ_{NN} is positive (semi-)definite if and only if $v_N > (\geq) \; 0$. Furthermore, the rank of Σ_{NN} is equal to the number of positive elements in v_N.

Proof

Assume Σ_{NN} is positive (semi-)definite. For $j \in N$, U_j is invertible, since it is upper triangular with nonzero elements along the diagonal. Thus,

$$\text{diag}(v_N) = S_0 = U_1^{-T} \cdots U_n^{-T} \Sigma_{NN} U_n^{-1} \cdots U_1^{-1} \quad .$$

Hence, $\text{diag}(v_N)$ is a congruence transformation of Σ_{NN} and is positive (semi-)definite, since Σ_{NN} is positive (semi-)definite [14, p. 259].

Assume $v_N > (\geq)\ 0$. Σ_{NN} is a congruence transformation of $S_0 = \text{diag}(v_N)$, and thus, Σ_{NN} is positive (semi-)definite.

The rank of $\text{diag}(v_N)$ is equal to the number of positive elements of v_N. The transformation matrices U_j and U_j^T are of full rank; therefore, the rank of Σ_{NN} is equal to the rank of $\text{diag}(v_N)$. □

APPENDIX 1.B

PROOFS FOR PROBABILISTIC ANALYSIS

<u>Theorem (Removal)</u>

Let node i have a single direct successor j, and let $C_0 = (C(j) \cup C(i))\backslash\{i\}$. Then

$$E[X_j | X_{C_0} = x_{C_0}] = \mu_j + \sum_{k \in C_0} (b_{kj} + b_{ki} b_{ij})(x_k - \mu_k) ,$$

and

$$\mathrm{Var}[X_j | X_{C_0} = x_{C_0}] = v_j + b_{ij}^2 \, v_i$$

<u>Proof</u>

Since both i and j have no successor nodes in C_0, the influence diagram definition of conditioning sets implies the following conditional independence relationships:

$$P\{X_i | X_{C_0} = x_{C_0}\} = P\{X_i | X_{C(i)} = x_{C(i)}\} , \quad \text{and}$$

$$P\{X_j | X_{C_0 \cup \{i\}} = x_{C_0 \cup \{i\}}\} = P\{X_j | X_{C(j)} = x_{C(j)}\} .$$

Thus,

$$\begin{aligned} E[X_j | X_{C_o} = x_{C_o}] &= E_{X_i}[E[X_j | X_{C_0 \cup \{i\}} = x_{C_0 \cup \{i\}}] | X_{C_0} = x_{C_0}] \\ &= E_{X_i}\Big[\mu_j + \sum_{k \in C(j)} b_{kj}(x_k - \mu_k) | X_{C_0} = x_{C_0}\Big] \\ &= \mu_j + \sum_{k \in C(j)\backslash\{i\}} b_{kj}(x_k - \mu_k) + b_{ij} \, E\big[X_i - \mu_i | X_{C(i)} = x_{C(i)}\big] \\ &= \mu_j + \sum_{k \in C(j)\backslash\{i\}} b_{kj}(x_k - \mu_k) + b_{ij} \sum_{k \in C(i)} b_{ki}(x_k - \mu_k) \end{aligned}$$

and

$$\mathrm{Var}[X_j|X_{C_o} = x_{C_o}] = E_{X_i}[\mathrm{Var}[X_k|X_{C_o \cup \{i\}} = x_{C_o \cup \{i\}}]|X_{C_o} = x_{C_o}]$$

$$+ \mathrm{Var}[E[X_j|X_{C_o \cup \{i\}} = x_{C_o \cup \{i\}}]|X_{C_o} = x_{C_o}]$$

$$= E_{X_i}[v_j|X_{C_o} = x_{C_o}]$$

$$+ \mathrm{Var}_{X_i}[\mu_j + \sum_{k \in C(j)} b_{kj}(x_k - \mu_k)|X_{C_o} = x_{C_o}]$$

$$= v_j + \mathrm{Var}[b_{ij}\, X_i|X_{C(i)} = x_{C(i)}]$$

$$= v_j + b_{ij}^2\, v_i \; . \qquad \square$$

Theorem (Reversal)

If node $i \in C(j)$, no other directed path exists from i to j, both v_i and v_j are non-zero, $C_0 = (C(j) \cup C(i))\backslash\{i\}$, and

$$b_{ji} = b_{ij} \frac{v_i}{v_j + b_{ij}^2\, v_i} \; ,$$

then

$$E[X_i|X_{C_0 \cup \{j\}} = x_{C_0 \cup \{j\}}] = \mu_i + \sum_{k \in C_0} (b_{ki} - (b_{kj} + b_{ki}b_{ij})b_{ji})(x_k - \mu_k)$$

$$+ b_{ji}(x_j - \mu_j) \; ,$$

and

$$\mathrm{Var}[X_i|X_{C_0 \cup \{j\}} = x_{C_0 \cup \{j\}}] = \frac{v_i\, v_j}{v_j + b_{ij}^2 v_i} \; .$$

Proof

For conditional variances,

$$\mathrm{Var}[X_i | X_j = x_j,\ X_{C_0} = x_{C_0}]$$

$$= \mathrm{Var}\left[X_i | X_{C_0} = x_{C_0}\right]$$

$$- \left(\mathrm{Cov}\left[X_i, X_j | X_{C_0} = x_{C_0}\right]\right)^2 \left(\mathrm{Var}\left[X_j | X_{C_0} = x_{C_0}\right]\right)^{-1} .$$

Similarly,

$$\mathrm{Var}[X_j | X_i = x_i,\ X_{C_0} = x_{C_0}]$$

$$= \mathrm{Var}\left[X_j | X_{C_0} = x_{C_0}\right]$$

$$- \left(\mathrm{Cov}\left[X_i, X_j | X_{C_0} = x_{C_0}\right]\right)^2 \left(\mathrm{Var}\left[X_i | X_{C_0} = x_{C_0}\right]\right)^{-1} .$$

Thus,

$$\mathrm{Var}[X_i | X_j = x_j,\ X_{C_0} = x_{C_0}]$$

$$= \frac{\mathrm{Var}[X_i | X_{C_0} = x_{C_0}]\mathrm{Var}[X_j | X_i = x_i,\ X_{C_0} = x_{C_0}]}{\mathrm{Var}\left[X_j | X_{C_0} = x_{C_0}\right]}$$

$$= \frac{v_i v_j}{v_j + b_{ij}^2 v_i} ,$$

applying the removal theorem to the denominator. The coefficients for conditional expectation are:

$$b_{ji} = \frac{\partial\ E\left[X_i | X_j = x_j,\ X_{C_0} = x_{C_0}\right]}{\partial x_j}$$

$$= \frac{\mathrm{Cov}[X_i, X_j | X_{C_0} = x_{C_0}]}{\mathrm{Var}[X_j | X_{C_0} = x_{C_0}]} ,$$

and, similarly,

$$b_{ij} = \frac{\mathrm{Cov}[X_i, X_j | X_{C_0} = x_{C_0}]}{\mathrm{Var}[X_i | X_{C_0} = x_{C_0}]} \quad .$$

Therefore,

$$b_{ji} = b_{ij} \frac{\mathrm{Var}[X_i | X_{C_0} = x_{C_0}]}{\mathrm{Var}[X_j | X_{C_0} = x_{C_0}]} \, ,$$

$$= b_{ij} \frac{v_i}{v_j + b_{ij}^2 v_i} \, .$$

Using primes to denote the updated coefficients after reversal, we have for $k \in C(i) \backslash \{j\}$,

$$E[X_i | X_k = x_k, X_{C_0} = \mu_{C_0}]$$

$$= E_{X_j}[E[X_i | X = x_j, X_k = x_k, X_{C_0} = \mu_{C_0}] | X_k = x_k, X_{C_0} = \mu_{C_0}]$$

$$= E_{X_j}[\mu_i + b'_{ki}(x_k - \mu_k) + b_{ji}(x_j - \mu_j) | X_k = x_k, X_{C_0} = \mu_{C_0}]$$

$$= \mu_i + b'_{ki}(x_k - \mu_k) + b_{ji}\, b'_{kj}(x_k - \mu_k)$$

$$= \mu_i + (b'_{ki} + b_{ji}\, b'_{kj})(x_k - \mu_k) \, .$$

Also,

$$E[X_i | X_k = x_k, X_{C_0} = \mu_{C_0}] = \mu_i + b_{ki}(\mu_k - x_k) \, .$$

Hence,

$$b'_{ki} = b_{ki} - b'_{kj}\, b_{ji}$$

$$= b_{ki} - (b_{kj} + b_{ki} b_{ij}) b_{ji} \, .$$

APPENDIX 1.C

DECISION ANALYSIS PROOFS

This appendix presents detailed proof of the two decision analysis operations on the normal influence diagram: removal of chance nodes into the value node, and removal of decision nodes into the value node.

<u>Theorem (Chance Node Removal)</u>

Let d be a sequence corresponding to some of the variables, and let $s = N\backslash d$. Partition Q into

$$\begin{bmatrix} Q_{dd} & Q_{ds} \\ Q_{sd} & Q_{ss} \end{bmatrix}$$

and p^T into (p_d^T, p_s^T). If

$$V(X_N) = \frac{1}{2} X_N^T Q X_N + p^T X_N + r ,$$

then

$$E[V(X_N) | X_s = x_s] = \frac{1}{2} x_s^T Q_{new} x_s + p_{new}^T x_s + r_{new} ,$$

where

$$Q_{new} = Q_{ss} + Q_{sd} B_{sd}^T + B_{sd} Q_{ds} + B_{sd} Q_{dd} B_{sd}^T ,$$

$$p_{new} = p_s + B_{sd} p_d + (Q_{sd} + B_{sd} Q_{dd})(\mu_d - B_{sd}^T \mu_s) ,$$

and

$$r_{new} = r + \frac{1}{2}\,\mathrm{trace}(Q_{dd}\ \mathrm{Var}[X_d|X_s = x_s])$$

$$+ \frac{1}{2}(\mu_d - B_{sd}^T\mu_s)^T Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d^T(\mu_d - B_{sd}^T\mu_s).$$

Proof

The quadratic term of the value function involving X_d has the expectation:

$$E[\frac{1}{2}\, X_d^T Q_{dd} X_d | X_s = x_s] = \frac{1}{2}\,\mathrm{trace}(Q_{dd}\ \mathrm{Var}[X_d|X_s = x_s])$$

$$+ \frac{1}{2}\, E[X_d|X_s = x_s]^T Q_{dd} E[X_d|X_s = x_s]$$

$$= \frac{1}{2}\,\mathrm{trace}(Q_{dd}\ \mathrm{Var}[X_d|X_s = x_s])$$

$$+ \frac{1}{2}\,[\mu_d + B_{sd}^T(x_s - \mu_s)]^T Q_{dd}[\mu_d + B_{sd}^T(x_s - \mu_s)]$$

$$= \frac{1}{2}\,\mathrm{trace}(Q_{dd}\ \mathrm{Var}[X_d|X_s = x_s])$$

$$+ \frac{1}{2}\,\mu_d^T Q_{dd}\mu_d + \mu_d^T Q_{dd} B_{sd}^T(x_s - \mu_s)$$

$$+ \frac{1}{2}(x_s - \mu_s)]^T B_{sd} Q_{dd} B_{sd}^T(x_s - \mu_s)$$

$$= \frac{1}{2}\,\mathrm{trace}(Q_{dd}\ \mathrm{Var}[X_d|X_s = x_s]) + \frac{1}{2}\,\mu_d^T Q_{dd}\mu_d - \mu_d^T Q_{dd} B_{sd}^T\mu_s$$

$$+ \mu_d^T Q_{dd} B_{sd}^T x_s + \frac{1}{2}\, x_s^T B_{sd} Q_{dd} B_{sd}^T x_s - \mu_s^T B_{sd} Q_{dd} B_{sd}^T x_s$$

$$+ \frac{1}{2}\,\mu_s^T B_{sd} Q_{dd} B_{sd}^T \mu_s$$

$$= \frac{1}{2}\,\mathrm{trace}(Q_{dd}\,\mathrm{Var}[X_d|X_s = x_s]) + \frac{1}{2}\,\mu_d^T Q_{dd}\mu_d - \mu_d^T Q_{dd} B_{sd}^T \mu_s$$

$$+ \frac{1}{2}\,\mu_s^T B_{sd} Q_{dd} B_{sd}^T \mu_s + (B_{sd} Q_{dd}\mu_d - B_{sd} Q_{dd} B_{sd}^T \mu_s)^T x_s$$

$$+ \frac{1}{2}\,x_s^T B_{sd} Q_{dd} B_{sd}^T x_s \;.$$

The value function can be multiplied out according to our partition as follows:

$$V(X_N) = V(X_s, X_d) = \frac{1}{2}\,X_s^T Q_{ss} X_s + \frac{1}{2}\,X_s^T Q_{sd} X_d + \frac{1}{2}\,X_d^T Q_{ds} X_s$$

$$+ \frac{1}{2}\,X_d^T Q_{dd} X_d + p_s^T X_s + p_d^T X_d + r \;.$$

Note that:

$$E[X_d|X_s = x_s] = \mu_d + B_{sd}^T(x_s - \mu_s) = \mu_d - B_{sd}^T\mu_s + B_{sd}^T x_s \;.$$

The expectation of the value function is:

$$E[V(X_N)|X_s = x_s] = \frac{1}{2}\,x_s^T Q_{ss} x_s + \frac{1}{2}\,x_s^T Q_{sd} E[X_d|X_s = x_s]$$

$$+ \frac{1}{2}\,E[X_d|X_s = x_s]^T Q_{ds} x_s + E[\frac{1}{2}\,X_d^T Q_{dd} X_d|X_s = x_s] + p_s^T x_s$$

$$+ p_d^T E[X_d|X_s = x_s] + r$$

$$= \frac{1}{2}\,x_s^T Q_{ss} x_s + \frac{1}{2}\,x_s^T Q_{sd}(\mu_d - B_{sd}^T\mu_s + B_{sd}^T x_s)$$

$$+ \frac{1}{2}\,(\mu_d - B_{sd}^T\mu_s + B_{sd}^T x_s)^T Q_{ds} x_s$$

$$+ \frac{1}{2}\,\mathrm{trace}(Q_{dd}\,\mathrm{Var}[X_d|X_s = x_s]) + \frac{1}{2}\,\mu_d^T Q_{dd}\mu_d - \mu_d^T Q_{dd} B_{sd}^T \mu_s$$

$$+ \frac{1}{2} \mu_s^T B_{sd} Q_{dd} B_{sd}^T \mu_s + (B_{sd} Q_{dd} \mu_d - B_{sd} Q_{dd} B_{sd}^T \mu_s)^T x_s$$

$$+ \frac{1}{2} x_s^T B_{sd} Q_{dd} B_{sd}^T x_s + p_s^T x_s + p_s^T (\mu_d - B_{sd}^T \mu_s + B_{sd}^T x_s) + r$$

$$= \frac{1}{2} x_s^T [Q_{ss} + Q_{sd} B_{sd}^T + B_{sd} Q_{ds} + B_{sd} Q_{dd} B_{sd}^T] x_s$$

$$+ [\frac{1}{2} Q_{sd}(\mu_d - B_{sd}^T \mu_s) + \frac{1}{2} Q_{ds}^T (\mu_d - B_{sd}^T \mu_s) + B_{sd} Q_{dd} \mu_d$$

$$- B_{sd} Q_{dd} B_{sd}^T \mu_s + p_s + B_{sd} p_d]^T x_s$$

$$+ \frac{1}{2} \operatorname{trace}(Q_{dd} \operatorname{Var}[X_d | X_s = x_s]) + \frac{1}{2} \mu_d^T Q_{dd} \mu_d - \mu_d^T Q_{dd} B_{sd}^T \mu_s$$

$$+ \frac{1}{2} \mu_s^T B_{sd} Q_{dd} B_{sd}^T \mu_s + p_d^T (\mu_d - B_{sd}^T \mu_s) + r$$

$$= \frac{1}{2} x_s^T [Q_{ss} + Q_{sd} B_{sd}^T + B_{sd} Q_{ds} + B_{sd} Q_{dd} B_{sd}^T] x_s$$

$$+ [p_s + B_{sd} p_d + (Q_{sd} + B_{sd} Q_{dd})(\mu_d - B_{sd}^T \mu_s)]^T x_s$$

$$+ \frac{1}{2} \operatorname{trace}(Q_{dd} \operatorname{Var}[X_d | X_s = x_s])$$

$$+ \frac{1}{2} (\mu_d - B_{sd}^T \mu_s) Q_{dd} (\mu_d - B_{sd}^T \mu_s) + p_d^T (\mu_d - B_{sd}^T \mu_s) + r .$$

□

<u>Theorem (Decision Node Removal)</u>

Let X_d be variables with no successors outside d and under decision maker control, which we may select after observing X_s. Assume Q_{dd} is negative definite and symmetric, so that an optimal choice exists and is unique. The preferred decision is

$$x_d^*(x_s) = -\,Q_{dd}^{-1}p_d - Q_{dd}^{-1}Q_{ds}x_s\ .$$

Also,

$$E[V(X_N)|X_s = x_s, X_d = x_d^*(x_s)] = \frac{1}{2}\,x_s^TQ_{new}x_s + p_{new}^Tx_s + r_{new}\ ,$$

where

$$Q_{new} = Q_{ss} - Q_{sd}Q_{dd}^{-1}Q_{ds}\ ,$$

$$p_{new} = p_s - Q_{sd}Q_{dd}^{-1}p_d\ ,$$

and

$$r_{new} = r - \frac{1}{2}\,p_d^TQ_{dd}^{-1}p_d\ .$$

<u>Proof</u>

The gradient of the value function with respect to the decision is

$$\partial V(x_s,x_d)/\partial x_d = Q_{ds}x_s + Q_{dd}x_d + p_d\ .$$

Setting the gradient to zero, the optimal decision is

$$x_d^*(x_s) = x_d^*(x_s) = -\,Q_{dd}^{-1}(Q_{ds}x_s + p_d) = -\,Q_{dd}^{-1}p_d - Q_{dd}^{-1}Q_{ds}x_s\ .$$

Substituting $x_d^*(x_s)$ into the value function,

$$E[V(X_N)|X_s = x_s, X_d = x_d^*(x_s)] = \frac{1}{2}\,x_s^TQ_{ss}x_s - (p_d^T + x_s^TQ_{ds})Q_{dd}^{-1}Q_{ds}x_s$$

$$+ \frac{1}{2}\,(p_d^T + x_s^TQ_{ds})Q_{dd}^{-1}Q_{dd}Q_{dd}^{-1}(Q_{ds}x_s + p_d) + p_s^Tx_s$$

$$-\ p_d^TQ_{dd}^{-1}(Q_{ds}x_s + p_d) + r$$

$$= \frac{1}{2}\,x_s^T(Q_{ss} - Q_{sd}Q_{dd}^{-1}Q_{ds})x_s + (p_s - Q_{sd}Q_{dd}^{-1}p_d)^Tx_s$$

$$+\ r - \frac{1}{2}\,p_d^TQ_{dd}^{-1}p_d\ . \qquad \square$$

APPENDIX 1.D

EXPONENTIAL UTILITY WITH QUADRATIC VALUE FUNCTION

Techniques for evaluating the conditional expected utility of the value lottery and selecting decisions to maximize conditional expected utility are presented in this appendix. The value function is assumed to be quadratic, and the utility is exponential with constant risk aversion coefficient γ [10].

<u>Theorem (Removal of chance node into value node)</u>

Assume the utility is deterministic for a given state vector x_N,

$$U(x_N) = a + b \exp\{-\gamma(\tfrac{1}{2}\, x_N^T Q x_N + p^T x_N + r)\} ,$$

with $b < 0$. If removal of the scalar random variable X_d is permitted, the conditional expected utility is

$$U(x_s) = a + b_{new} \exp \{-\gamma(\tfrac{1}{2}\, x^T Q_{new} x_s + p_{new}^T x_s + r_{new})\},$$

where $s = N \backslash d$. If $v_d \neq 0$,

$$Q_{new} = Q_{ss} + B_{sd} B_{sd}^T/(\gamma v_d) - [v_d/(\gamma + \gamma^2 v_d Q_{dd})](\gamma Q_{sd} - B_{sd}/v_d)$$

$$\times (\gamma Q_{sd} - B_{sd}/v_d)^T ,$$

$$p_{new} = p_s + (\mu_d - B_{sd}^T \mu_s) B_{sd}/(\gamma v_d)$$

$$- [v_d/(\gamma + \gamma^2 v_d Q_{dd})][\gamma p_d - (\mu_d - B_{sd}^T \mu_s)/v_d](\gamma Q_{sd} - B_{sd}/v_d) ,$$

$$r_{new} = r + (\mu_d - B_{sd}^T \mu_s)^2/(2\gamma v_d)$$

$$- \tfrac{1}{2}[v_d/(\gamma + \gamma^2 v_d Q_{dd})][\gamma p_d - (\mu_d - B_{sd}^T \mu_s)/v_d]^2 ,$$

and

$$b_{new} = b/(1 + \gamma v_d/Q_{dd})^{1/2} .$$

If $v_d = 0$,

$$Q_{new} = Q_{ss} + Q_{sd}B_{sd}^T + B_{sd}Q_{ds} + B_{sd}Q_{dd}B_{sd}^T ,$$

$$p_{new} = p_s + B_{sd}p_d + (Q_{sd} + B_{sd}Q_{dd})(\mu_d - B_{sd}^T\mu_s) ,$$

$$r_{new} = r + \frac{1}{2}(\mu_d - B_{sd}^T\mu_s)^T Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d^T(\mu_d - B_{sd}^T\mu_s) ,$$

and

$$b_{new} = b .$$

<u>Proof</u>

$$U(x_s) = E[U(X_N)|X_s = x_s]$$

$$= E[a + b\exp\{-\gamma(\tfrac{1}{2} X_N^T Q X_N + p^T X_N + r)\}|X_s = x_s]$$

$$= a + b\ E[\exp\{-\gamma(\tfrac{1}{2} X_s^T Q_{ss} X_s + X_s^T Q_{sd} X_d + \tfrac{1}{2} Q_{dd} X_d^2 + p_s^T X_s$$

$$+ p_d X_d + r)\}|X_s = x_s]$$

$$= a + b\exp\{-\gamma(\tfrac{1}{2} x_s^T Q_{ss} x_s + p_s^T x_s + r)\}$$

$$\times E[\exp\{-\gamma(\tfrac{1}{2} Q_{dd} X_d^2 + [p_d + Q_{ds} x_s] X_d)\}|X_s = x_s] \quad (1.D.1)$$

For $v_d \neq 0$,

$$E[\exp\{-\gamma(\tfrac{1}{2} Q_{dd} X_d^2 + [p_d + Q_{ds} x_s] X_d)\}|X_s = x_s]$$

$$= \int (2\pi v_d)^{-1/2} \exp\{-\tfrac{1}{2}[x_d - \mu_d - B_{sd}^T(x_s - \mu_s)]^2/v_d\}$$

$$\times \exp\{-\gamma(\tfrac{1}{2} Q_{dd}x_d^2 + [p_d + Q_{ds}x_s]x_d)\}dx_d$$

$$= \int (2\pi v_d)^{-1/2} \exp\{-\tfrac{1}{2} x_d^2/v_d + [\mu_d + B_{sd}^T(x_s - \mu_s)]x_d/v_d$$

$$- \tfrac{1}{2} [\mu_d + B_{sd}^T(x_s - \mu_s)]^2/v_d - \tfrac{1}{2} \gamma Q_{dd}x_d^2$$

$$- \gamma(p_d + Q_{ds}x_s)x_d\}dx_d$$

$$= \int (2\pi v_d)^{-1/2} \exp\{-[\tfrac{1}{2}(1/v_d + \gamma Q_{dd})x_d^2$$

$$+ \{\gamma(p_d + Q_{ds}x_s) - [\mu_d + B_{sd}^T(x_s - \mu_s)]/v_d\}x_d$$

$$+ \tfrac{1}{2}\{\mu_d + B_{sd}^T(x_s - \mu_s)\}^2/v_d]\}dx_d .$$

From Graybill [4, p. 48],

$$\int a_0 \exp\{-[Bx^2 + bx + b_0]\}dx = a_0(\pi/B)^{1/2} \exp\{b^2/(4B) - b_0\} .$$

Thus,

$$E[\exp\{-\gamma(\tfrac{1}{2} Q_{dd}X_d^2 + [p_d + Q_{ds}x_s]X_d)\}|X_s = x_s]$$

$$= (2\pi v_d)^{-1/2}(\pi/[\tfrac{1}{2}(1/v_d + \gamma Q_{dd})])^{1/2}$$

$$\times \exp\{\{\gamma(p_d + Q_{ds}x_s) - [\mu_d + B_{sd}^T(x_s - \mu_s)]/v_d\}^2$$

$$/(4[\tfrac{1}{2}(1/v_d + \gamma Q_{dd})]) - \tfrac{1}{2}[\mu_d + B_{sd}^T(x_s - \mu_s)]^2/v_d\}$$

$$= (1 + \gamma v_d Q_{dd})^{-1/2} \exp\{\tfrac{1}{2}(1/v_d + \gamma Q_{dd})^{-1}$$

$$\times (\gamma p_d - [\mu_d - B_{sd}^T \mu_s]/v_d$$

$$+ [\gamma Q_{sd} - B_{sd}/v_d]^T x_s)^2 - \tfrac{1}{2}[\mu_d + B_{sd}^T(x_s - \mu_s)]^2/v_d\}$$

$$= (1 + \gamma v_d Q_{dd})^{-1/2} \exp\{\tfrac{1}{2}[v_d/(1 + \gamma v_d Q_{dd})] x_s^T(\gamma Q_{sd} - B_{sd}/v_d)$$

$$\times (\gamma Q_{sd} - B_{sd}/v_d)^T x_s$$

$$+ [v_d/(1+\gamma v_d Q_{dd})](\gamma p_d - [\mu_d - B_{sd}^T \mu_s]/v_d)(\gamma Q_{sd} - B_{sd}/v_d)^T x_s$$

$$+ \frac{1}{2}\,[v_d/(1+\gamma v_d Q_{dd})](\gamma p_d - [\mu_d - B_{sd}^T \mu_s]/v_d)^2$$

$$- \frac{1}{2}\, x_s^T B_{sd} B_{sd}^T x_s / v_d$$

$$- [(\mu_d - B_{sd}^T \mu_s) B_{sd}]^T x_s / v_d - \tfrac{1}{2}(\mu_d - B_{sd}^T \mu_s)^2 / v_d\}$$

$$= (1 + \gamma v_d Q_{dd})^{-1/2} \exp\{-\gamma[\tfrac{1}{2}\, x_s^T [B_{sd} B_{sd}^T/(\gamma v_d)$$

$$- \{v_d/(\gamma + \gamma^2 v_d Q_{dd})\}(\gamma Q_{sd} - B_{sd}/v_d)(\gamma Q_{sd} - B_{sd}/v_d)^T] x_s$$

$$+ [(\mu_d - B_{sd}^T \mu_s) B_{sd}/(\gamma v_d)$$

$$- \{v_d/(\gamma + \gamma^2 v_d Q_{dd})\}[\gamma p_d - (\mu_d - B_{sd}^T \mu_s)/v_d]$$

$$\times (\gamma Q_{sd} - B_{sd}/v_d)]^T x_s + (\mu_d - B_{sd}^T \mu_s)^2/(2\gamma v_d)$$

$$- \tfrac{1}{2}[v_d/(\gamma + \gamma^2 v_d Q_{dd})][\gamma p_d - (\mu_d - B_{sd}^T \mu_s)/v_d]^2]\}.$$

Substituting this into (1.D.1) for the conditional expectation yields the desired result for $v_d \neq 0$. For $v_d = 0$, the conditional expectation is a deterministic substitution as follows:

$$E[\exp\{-\gamma(\tfrac{1}{2}Q_{dd}X_d^2 + [p_d + Q_{ds}x_s]X_d)\}|X_s = x_s]$$

$$= \exp\{-\gamma(\tfrac{1}{2}Q_{dd}[\mu_d + B_{sd}^T(x_s - \mu_s)]^2$$

$$+ [p_d + Q_{ds}x_s][\mu_s + B_{sd}^T(x_s - \mu_s)])\}$$

$$= \exp\{-\gamma(\tfrac{1}{2}\ x_s^T[Q_{sd}B_{sd}^T + B_{sd}Q_{ds} + B_{sd}Q_{dd}B_{sd}^T]x_s$$

$$+ [B_{sd}p_d + (Q_{sd} + B_{sd}Q_{dd})(\mu_d - B_{sd}^T\mu_s)]^T x_s$$

$$+ \tfrac{1}{2}(\mu_d - B_{sd}^T\mu_s)^T Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d^T(\mu_d - B_{sd}^T\mu_s))\}\ .$$

Substituting this into (1.D.1) yields the desired result for $v_d = 0$. □

Theorem (Decision node removal)

Let X_d be variables with no successors outside d and under decision maker control, which we may select after observing X_s. Assume Q_{dd} is negative definite and symmetric. The preferred decision is

$$x_d^*(x_s) = -Q_{dd}^{-1}p_d - Q_{dd}^{-1}Q_{ds}x_s\ .$$

Also,

$$E[U(X_N)|X_s = x_s,\ X_d = x_d^*(x_s)]$$

$$= a + b\ \exp\{-\gamma(\tfrac{1}{2}\ x_s^T Q_{new}x_s + p_{new}^T x_s + r_{new})\}\ .$$

where

$$Q_{new} = Q_{ss} - Q_{sd}Q_{dd}^{-1}Q_{ds} ,$$

$$p_{new} = p_s - Q_{sd}Q_{dd}^{-1}p_d ,$$

and

$$r_{new} = r - \frac{1}{2} p_d^T Q_{dd}^{-1} p_d .$$

Proof

The gradient of the utility function with respect to the decision is

$$\partial U(x_s,x_d)/\partial x_d = -b\gamma \exp\{-\gamma(\frac{1}{2} x_N^T Qx_N + p^T x_N + r)\}$$
$$\times [Q_{ds}x_s + Q_{dd}x_d + p_d] .$$

Setting the gradient to zero, the optimal decision is

$$x_d^*(x_s) = x_d^*(x_s) = -Q_{dd}^{-1}(Q_{ds}x_s + p_d) = -Q_{dd}^{-1}p_d - Q_{dd}^{-1}Q_{ds}x_s .$$

The Hessian of the utility function is

$$\partial^2 U(x_s,x_d)/\partial x_d \partial x_d^T = -b\gamma \exp\{-\gamma(\frac{1}{2} x_N^T Qx_N + p^T x_N + r)\}$$
$$\times [Q_{dd} + (Q_{ds}x_s + Q_{dd}x_d + p_d)(Q_{ds}x_s + Q_{dd}x_d + p_d)^T] .$$

Evaluating at $x_d^*(x_s)$, the Hessian is

$$\partial^2 U(x_s,x_d)/\partial x_d \partial x_d^T = -b\gamma \exp\{-\gamma(\frac{1}{2} x_N^T Qx_N + p^T x_N + r)\} Q_{dd} .$$

The scalar expression multiplying Q_{dd} is positive, and Q_{dd} is negative definite. Hence, the Hessian is negative definite, and

$x_d^*(x_s)$ is a maximizing choice of x_d. Substituting $x_d^*(x_s)$ into the utility function,

$$
\begin{aligned}
E[U(X_N)|X_s = x_s, X_d = x_d^*(x_s)] &= a + b\exp\{-\gamma(\tfrac{1}{2}\, x_s^T Q_{ss} x_s \\
&\quad - (p_d^T + x_s^T Q_{ds}) Q_{dd}^{-1} Q_{ds} x_s \\
&\quad + \tfrac{1}{2}\,(p_d^T + x_s^T Q_{ds}) Q_{dd}^{-1} Q_{dd} Q_{dd}^{-1} (Q_{ds} x_s + p_d) + p_s^T x_s \\
&\quad - p_d^T Q_{dd}^{-1} (Q_{ds} x_s + p_d) + r)\} \\
&= a + b\exp\{-\gamma(\tfrac{1}{2}\, x_s^T (Q_{ss} - Q_{sd} Q_{dd}^{-1} Q_{ds}) x_s \\
&\quad + (p_s - Q_{sd} Q_{dd}^{-1} p_d)^T x_s + r - \tfrac{1}{2}\, p_d^T Q_{dd}^{-1} p_d)\} \;. \qquad \square
\end{aligned}
$$

APPENDIX 1.E

VARIANCE OF THE VALUE LOTTERY

In this appendix, we develop techniques for evaluating the conditional variance of the value lottery. The following identities for a multivariate normal vector x with mean μ and covariance matrix Σ will be used:

(1) $\operatorname{Cov}[a^T x, b^T x] = a^T \Sigma b,$

(2) $\operatorname{Cov}[a^T x, \frac{1}{2} x^T B x] = a^T \Sigma B \mu,$

(3) $E[\frac{1}{2} x^T A x] = \frac{1}{2} \operatorname{trace}\{A\Sigma\} + \frac{1}{2} \mu^T A \mu,$ and

(4) $\operatorname{Var}[\frac{1}{2} x^T A x] = \frac{1}{2} \operatorname{trace}\{(A\Sigma)^2\} + \mu^T A \Sigma A \mu$.

Proofs of these identities are in Searle [11, pp. 55-57].

<u>Theorem</u>

Assume the value function is deterministic for a given state vector x_N,

$$V(x_N) = \frac{1}{2} x_N^T Q x_N + p^T x_N + r .$$

If removal of chance nodes X_d is permitted, the conditional variance of the value function $\operatorname{Var}[V(X_N) | X_s = x_s]$ is a quadratic form,

$$\frac{1}{2} x_s^T A x_s + b^T x_s + c ,$$

where A is a positive semi-definite symmetric matrix.

Proof

$$\mathrm{Var}[V(X_N)|X_s = x_s] = E[\mathrm{Var}[V(X_N)|X_N = x_N]|X_s = x_s]$$

$$+ \mathrm{Var}[E[V(X_N)|X_N = x_N]|X_s = x_s]$$

$$= E[0|X_s = x_s] + \mathrm{Var}[\tfrac{1}{2}X_N^TQX_N + p^TX_N + r|X_s = x_s]$$

$$= \mathrm{Var}[X_s^TQ_{sd}X_d|X_s = x_s] + \mathrm{Var}[\tfrac{1}{2}\, X_d^TQ_{dd}X_d|X_s = x_s]$$

$$+ 2\mathrm{Cov}[X_s^TQ_{sd}X_d, \tfrac{1}{2}\, X_d^TQ_{dd}X_d|X_s = x_s] + \mathrm{Var}[p_d^TX_d|X_s = x_s]$$

$$+ 2\mathrm{Cov}[p_d^TX_d, X_s^TQ_{sd}X_d|X_s = x_s] + 2\mathrm{Cov}[p_d^TX_d, \tfrac{1}{2}\, X_d^TQ_{dd}X_d|X_s = x_s]$$

$$= x_s^TQ_{sd}\mathrm{Var}[X_d|X_s = x_s]Q_{ds}x_s + \frac{1}{2}\,\mathrm{trace}\{(Q_{dd}\mathrm{Var}[X_d|X_s = x_s])^2\}$$

$$+ \{u_d + B_{sd}^T(x_s-\mu_s)\}^TQ_{dd}\mathrm{Var}[X_d|X_s = x_s]Q_{dd}\{\mu_d+B_{sd}^T(x_s-\mu_s)\}$$

$$+ 2x_s^TQ_{sd}\mathrm{Var}[X_d|X_s = x_s]Q_{dd}\{\mu_d+B_{sd}^T(x_s-\mu_s)\} + p_d^T\mathrm{Var}[X_d|X_s = x_s]p_d$$

$$+ 2p_d^T\mathrm{Var}[X_d|X_s = x_s]Q_{ds}x_s + 2p_d^T\mathrm{Var}[X_d|X_s = x_s]Q_{dd}\{\mu_d+B_{sd}^T(x_s-\mu_s)\}$$

$$= x_s^T[(Q_{sd} + B_{sd}Q_{dd})\mathrm{Var}[X_d|X_s = x_s](Q_{ds} + Q_{dd}B_{sd}^T)]x_s$$

$$+ 2[(Q_{sd} + B_{sd}Q_{dd})\mathrm{Var}[X_d|X_s = x_s]\{Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d\}]^Tx_s$$

$$+ \frac{1}{2}\,\mathrm{trace}\{(Q_{dd}\mathrm{Var}[X_d|X_s = x_s])^2\}$$

$$+ \{Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d\}^T\mathrm{Var}[X_d|X_s = x_s]\{Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d\}\,.$$

Letting

$$A = 2(Q_{sd} + B_{sd}Q_{dd})\mathrm{Var}[X_d|X_s = x_s](Q_{ds} + Q_{dd}B^T_{sd}) ,$$

$$b = 2[(Q_{sd} + B_{sd}Q_{dd})\mathrm{Var}[X_d|X_s = x_s]\{Q_{dd}(\mu_d - B^T_{sd}\mu_s) + p_d\}], \quad \text{and}$$

$$c = \frac{1}{2}\,\mathrm{trace}\{(Q_{dd}\mathrm{Var}[X_d|X_s = x_s])^2\}$$

$$+ \{p_d + Q_{dd}(\mu_d - B^T_{sd}\mu_s)\}^T\mathrm{Var}[X_d|X_s = x_s]\{p_d + Q_{dd}(\mu_d - B^T_{sd}\mu_s)\} ,$$

the assertion regarding the quadratic form is proved. A is positive semi-definite, since it is a congruence transformation of the positive semi-definite covariance matrix $\mathrm{Var}[X_d|X_s = x_s]$. □

<u>Corollary</u>

Assume the expected value of the value function for a given state vector x_N is

$$E[V|X_N = x_n] = \frac{1}{2}\, x_N^T Q x_N + p^T x_N + r .$$

Also, assume the conditional variance of the value function is a quadratic form,

$$\mathrm{Var}[V|X_N = x_N] = \frac{1}{2}\, x_N^T A x_N + b^T x_N + c ,$$

where A is a positive semi-definite symmetric matrix. If removal of X_d is permitted, then the transformations to remove X_d are:

$$A_{ss} \leftarrow A_{ss} + A_{sd}B^T_{sd} + B_{sd}A_{ds} + B_{sd}A_{dd}B^T_{sd}$$

$$+ 2(Q_{sd} + B_{sd}Q_{dd})\mathrm{Var}[X_d|X_s = x_s](Q_{ds} + Q_{dd}B^T_{sd})$$

$$b_s \leftarrow b_s + (A_{sd} + B_{sd}A_{dd})(\mu_d - B_{sd}^T\mu_s) + A_{sd}b_d$$

$$+ 2[(Q_{sd} + B_{sd}Q_{dd})\mathrm{Var}[X_d|X_s = x_s]\{Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d\}]$$

$$c \leftarrow c + \frac{1}{2}\,\mathrm{trace}(A_{dd}\mathrm{Var}[X_d|X_s = x_s])$$

$$+ \frac{1}{2}\,\mathrm{trace}\{(Q_{dd}\mathrm{Var}[X_d|X_s = x_s])^2\}$$

$$+ \frac{1}{2}(\mu_d - B_{sd}^T\mu_s)^T A_{dd}(\mu_d - B_{sd}^T\mu_s) + b_d^T(\mu_d - B_{sd}^T\mu_s)$$

$$+ \{Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d\}^T\mathrm{Var}[X_d|X_s = x_s]\{Q_{dd}(\mu_d - B_{sd}^T\mu_s) + p_d\}$$

$$N \leftarrow N\backslash d. \quad \square$$

CHAPTER 2

DISCRETE-TIME FILTERS

2.1 Introduction

Kalman [7] solved the problem of sequential least squares estimation of state vectors for discrete-time linear systems with noise. A solution equivalent to Kalman's filter [3, pp. 382-88] defines a multivariate normal model of the system and applies Bayes' rule to derive the posterior state vector distribution, which also is multivariate normal. In this chapter, the Bayesian solution is implemented using the normal influence diagram. Processing of data using influence diagram techniques compares favorably with efficient and stable factorization techniques developed since Kalman's original work [1, pp. 82-90; 15].

2.2 Model Notation

A discrete-time filtering problem [3, p. 360] is described by a dynamic process

$$x(k+1) = \Phi(k)x(k) + \Gamma(k)w(k) \tag{2.2.1}$$

and a measurement process

$$z(k) = H(k)x(k) + v(k) \text{ for } k = 0,\ldots,N; \tag{2.2.2}$$

where

$$x(k) \in R^n,\ w(k) \in R^r,\ z(k) \in R^p, \text{ and } v(k) \in R^p.$$

The objective is to perform inference about the state vector $x(k+1)$, given measurements $z(0),\ldots,z(k)$, for $k = 0,\ldots,N$. The probabilistic structure of the variables is multivariate normal with the following properties:

$$E[x(0)] = \mu_0; \tag{2.2.3}$$

$$\mathrm{Var}[x(0)] = P_0; \tag{2.2.4}$$

$$E[w(k)] = 0 \text{ for } k = 0,\dots,N; \tag{2.2.5}$$

$$\mathrm{Cov}[w(j),w(k)] = \delta_{jk}Q_k \quad \text{for } j = 0,\dots,N \text{ and } k = 0,\dots,N; \tag{2.2.6}$$

$$\mathrm{Cov}[x(0),w(0)] = 0; \tag{2.2.7}$$

$$E[v(k)] = 0 \text{ for } k = 0,\dots,N; \tag{2.2.8}$$

$$\mathrm{Cov}[v(j),v(k)] = \delta_{jk}R_k \quad \text{for } j = 0,\dots,N \text{ and } k = 0,\dots,N; \tag{2.2.9}$$

$$\mathrm{Cov}[w(j),v(k)] = 0 \quad \text{for } j = 0,\dots,N \text{ and } k = 0,\dots,N; \tag{2.2.10}$$

$$\mathrm{Cov}[x(0),v(k)] = 0 \text{ for } k = 0,\dots,N. \tag{2.2.11}$$

Here,

$$\delta_{jk} = \begin{cases} 1 & \text{for } j = k \\ 0 & \text{for } j \neq k \end{cases} .$$

Whenever possible, the indexing argument k will be omitted to simplify notation.

The vectors v and w are noise vectors, where v is the measurement noise, and w is the process noise. Because the noise covariance matrices Q and R are assumed to be diagonal, the noise components are independent. If they were not diagonal, they could be transformed to diagonal matrices. Implementation of the transformations is an order n^3

algorithm and can be done prior to data collection, particularly when $Q_k = Q$ and $R_k = R$ for $k = 0,\ldots,N$. The noise covariances can be time-varying, particularly in real-time navigation systems where process noise covariances can be a function of time between observations, due to accelerations that cannot be modelled deterministically. In these cases, transformation to diagonal noise covariances may not be desirable, because of data storage, computational efficiency, and accuracy considerations. Influence diagram processing does not require the assumed independence of noise components; however, the processing rules can take advantage of independence if it is part of the probabilistic structure.

The multivariate normal influence diagram representation of discrete-time filtering is shown in Fig. 2-1. Each node is a vector, and the matrix on each arc between nodes is the gradient of the conditional expectation of the successor node with respect to the predecessor node's value. Letting $E[x(k)] = \mu$,

$$E[x(k+1)\,|\,x(k) = \xi,\ w(k) = 0] = \Phi(k)\mu + \Phi(k)(\xi-\mu) = \Phi(k)\xi.$$

Thus,

$$\partial E[x_i(k+1)\,|\,x(k) = \xi,\ w(k) = 0]/\partial\xi_j = \Phi_{ij}(k).$$

Arcs in Fig. 2-1 with no matrix are assumed to be labeled with the appropriately dimensioned identity matrix. A deterministic linear model, such as the dynamic process (equation 2.2.1) and the measurement process (equation 2.2.2), is represented in Fig. 2-1 by a double-circle deterministic node (the left-hand side of the equation) and its conditioning nodes and arcs from them (the right-hand side of the equation).

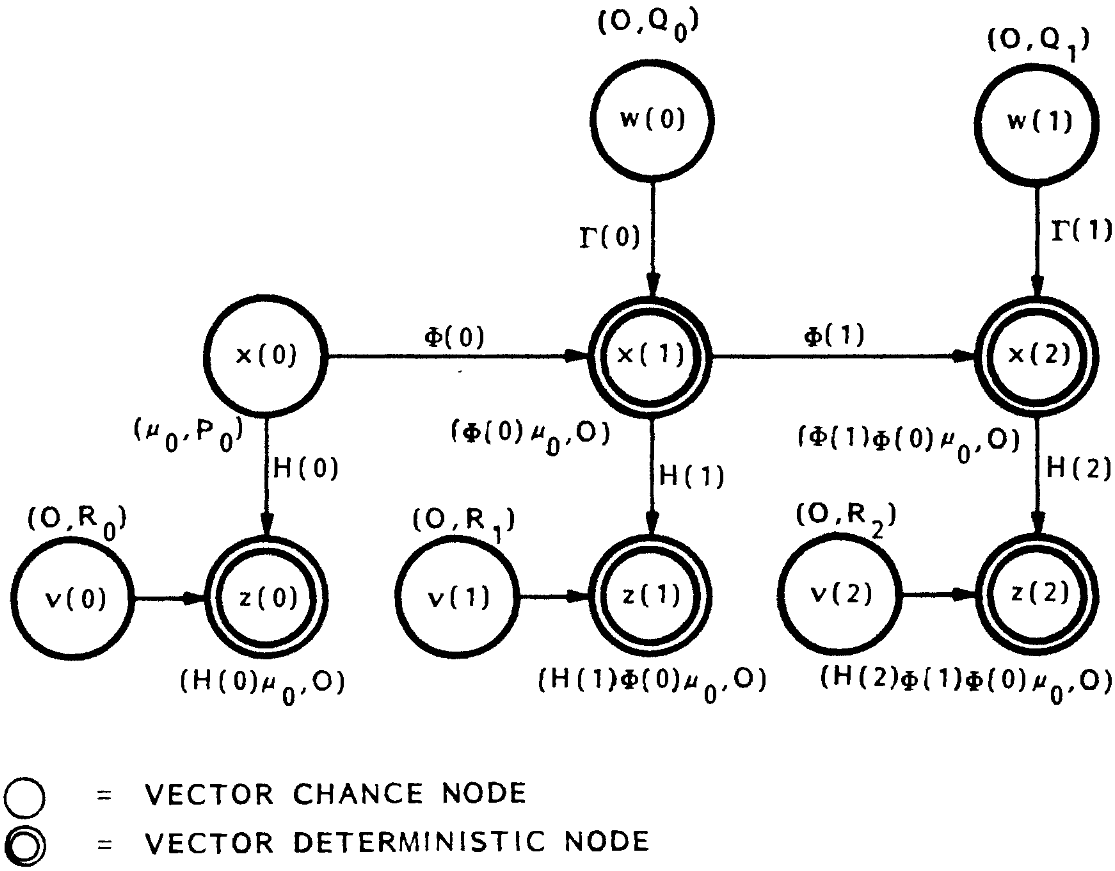


Fig. 2-1 Influence Diagram Representation of Discrete-Time Filtering

In an influence diagram, the lack of an arc between two vector nodes represents conditional independence of the random vectors. For the multivariate normal, a covariance of zero is a necessary and sufficient condition for independence. For example, the independence assumption of equation (2.2.6) is represented by not having an arc in Fig. 2-1 between nodes w(j) and w(k) for $j \neq k$. Similarly, the independence assumptions of equations (2.2.7), (2.2.9), (2.2.10), and (2.2.11) are also shown by lack of arcs in Fig. 2-1.

Each vector node is represented by its own influence subdiagram. For a scalar measurement z, the influence subdiagram for a scalar measurement process is shown in Fig. 2-2. The scalar values on each arc are the partial derivatives of the successor's conditional expectation with respect to the predecessor's value. The arcs from $x_1,\ldots,x_n$ to z are the elements of H in the measurement process equation (2.2.2). Associated with each scalar node are two values, the unconditional mean of the random variable and the conditional variance of the random variable, given that all the direct and indirect predecessor random variables in the diagram are known. A node is a deterministic linear function of its predecessors if and only if its conditional variance is zero. In Fig. 2-2, a specific dependence ordering of x has been assumed; however, any of the n! orderings could be used. For purposes of algorithm implementation, all nodes x(k) for $k = 1,\ldots,N$ are assumed to have the same ordering as x(0). A subdiagram for a time update process is shown in Fig. 2-3, with $n = 3$ and $r = 3$.

For a set of scalar nodes, the combination of arc coefficients and conditional variances is a factorization of the covariance matrix of the variables. Referring to Fig. 2-3, let

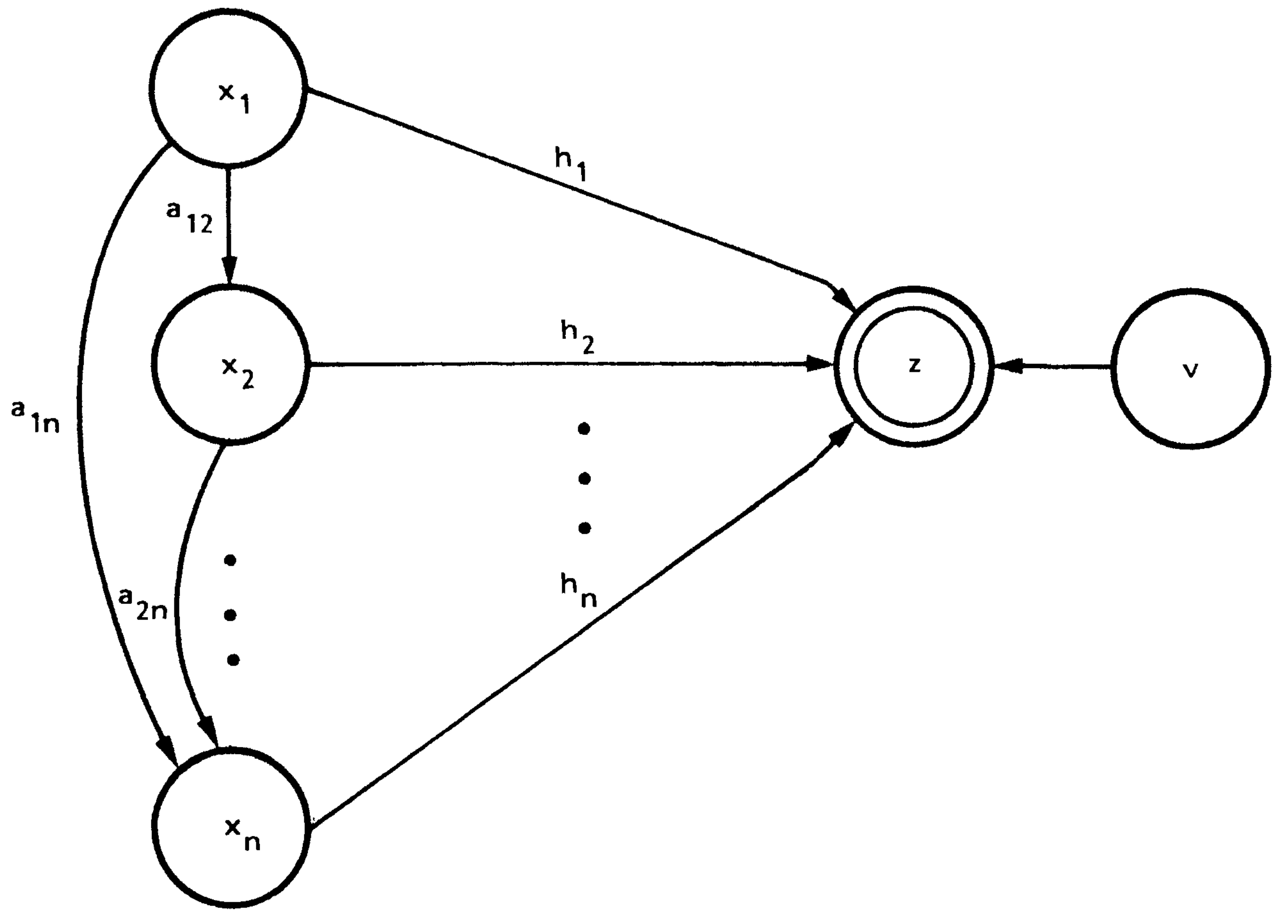


Fig. 2-2 Scalar Measurement Update Subdiagram

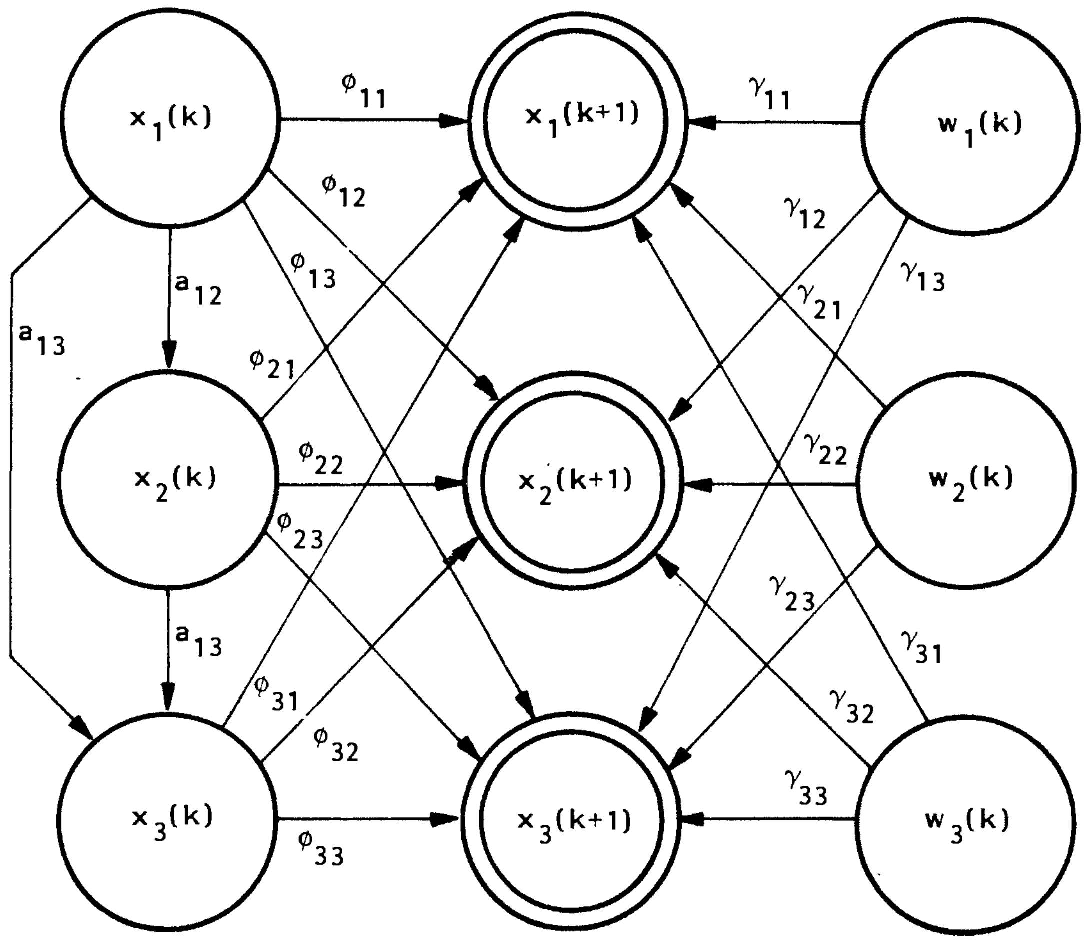


Fig. 2-3 Time Update Subdiagram

$$U_j = \begin{bmatrix} I_{j-1} & a_{\cdot j} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{n-j} \end{bmatrix}$$

where

$$a_{\cdot j} = [a_{1j}, \ldots, a_{j-1,j}]^T.$$

Appendix 1.A showed that the covariance matrix of $x = [x_1, \ldots, x_n]^T$ is

$$\begin{aligned} P &= U_n^T \ldots U_1^T \mathrm{diag}(v_1, \ldots, v_n) U_1 \ldots U_n \\ &= U^T D U, \end{aligned}$$

where $v_1, \ldots, v_n$ are the conditional variances of $x_1, \ldots, x_n$. Thus, a normal influence diagram could be interpreted as a finer level version of the $U^T DU$ factorization of P [14]. Bierman [1, pp. 76-81, pp. 124-9] uses the $U^T DU$ factorization in his U-D filter. Although a similarity exists, it will be shown that discrete-time filtering using influence diagram techniques is different from Bierman's U-D filter in solution approach and computational efficiency.

2.3 Influence Diagram Implementation of Discrete-Time Filtering

Solution of the filtering problem for each stage k involves two steps, a measurement update and a time update. In Fig. 2-4, a measurement update for stage 0 consists of removing v(0) into z(0), reversing z(0) with x(0), and updating the expectation of x(0) via instantiation of z(0) with its observed value. A time update follows the measurement update and consists of removing x(0) into x(1), and removing w(0) into x(1). The removal and reversal operations involve vector nodes connected

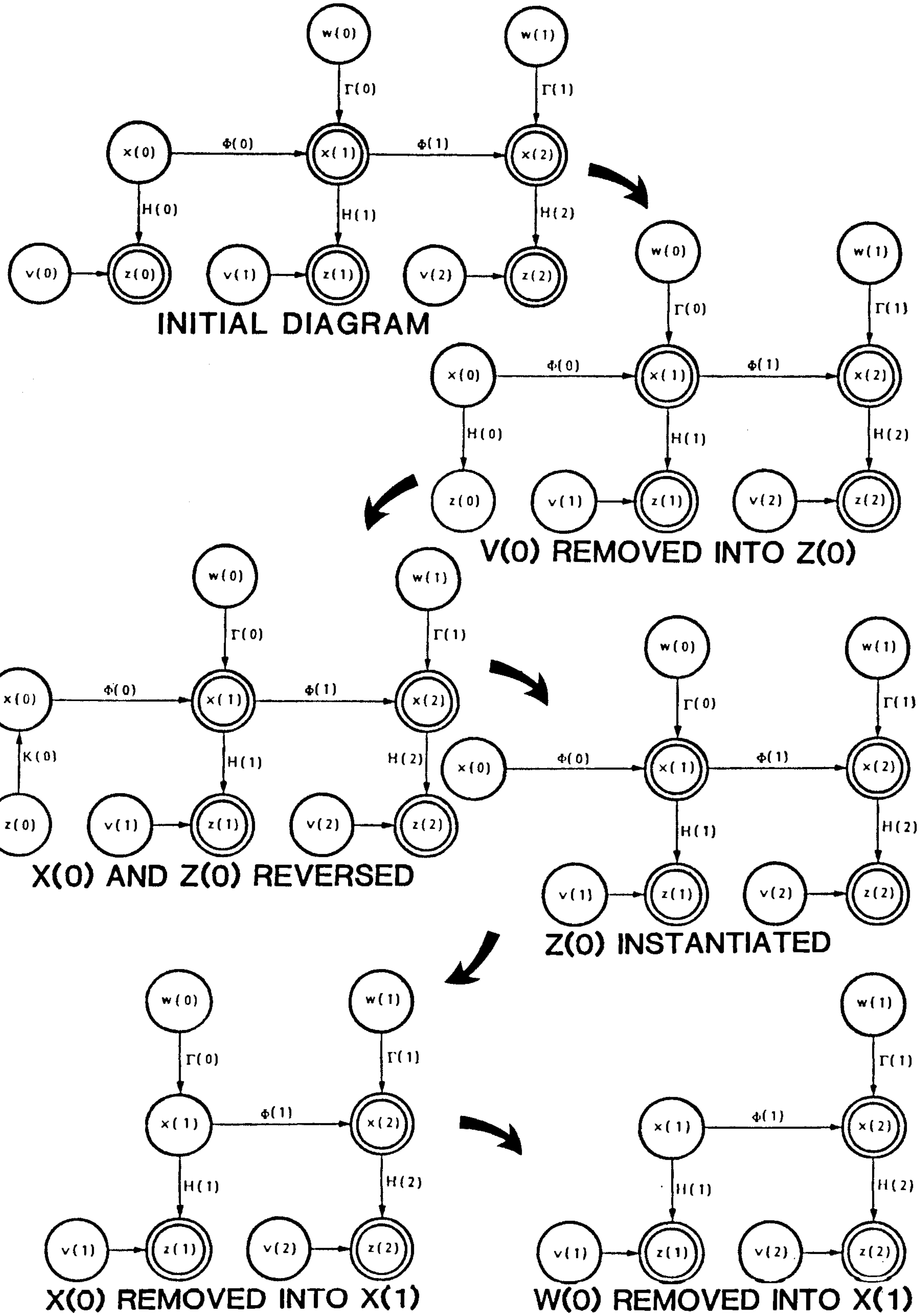


Fig. 2-4 Influence Diagram Implementation of Discrete-Time Filtering

by a matrix of arcs. We defer detailed description of operations involving vector nodes to Section 2.4. For the moment, these operations can be treated as if the nodes involved were scalar nodes joined by scalar arcs.

If the components of v are independent and the components of z, given x and v, are conditionally independent, parallel removal of the p components of v is permitted.

$$
\begin{aligned}
P\{z_1,\ldots,z_p|x\} &= \int dvP\{z_1,\ldots,z_p|x,v_1,\ldots,v_p\}P\{v_1,\ldots,v_p\} \\
&= \int dv[P\{z_1|x,v_1\}\ldots P\{z_p|x,v_p\}][P\{v_1\}\ldots P\{v_p\}] \\
&= \int dv_1P\{z_1|x,v_1\}P\{v_1\}\ldots\int dv_pP\{z_p|x,v_p\}P\{v_p\} \\
&= \prod_{i=1}^{p} \int dv_iP\{z_i|x,v_i\}P\{v_i\} \\
&= \prod_{i=1}^{p} P\{z_i|x\}.
\end{aligned}
$$

Thus, after parallel removal of v into z, components of z remain conditionally independent. This permits sequential processing of the vector data, z, as scalar measurements.

$$
\begin{aligned}
P\{x|z_1,\ldots,z_p\} &= P\{z_1,\ldots,z_p|x\}P\{x\}/P\{z_1,\ldots,z_p\} \\
&= P\{z_1|x\}\ldots P\{z_p|x\}P\{x\}/P\{z_1,\ldots,z_p\} \\
&= P\{z_1|x\}\ldots P\{z_p|x\}P\{x\}/[P\{z_1\}P\{z_2,\ldots,z_p|z_1\}] \\
&= \frac{P\{z_1|x\}P\{x\}}{\int dxP\{z_1|x\}P\{x\}} \; \frac{P\{z_2|x\}\ldots P\{z_p|x\}}{\int dxP\{z_2,\ldots,z_p|z_1,x\}P\{x\}} \\
&= \frac{P\{z_1|x\}P\{x\}}{\int dxP\{z_1|x\}P\{x\}} \; \frac{P\{z_2,\ldots,z_p|x\}}{\int dxP\{z_2,\ldots,z_p|x\}P\{x\}} \\
&= P\{x|z_1\}P\{z_2,\ldots,z_p|x\}/P\{z_2,\ldots,z_p\}.
\end{aligned}
$$

Reversal and instantiation of z_1 with x can be completed without adding dependence from z_1 to $z_2,\ldots,z_n$. For all i, reversal of z_i with x does not add dependence arcs to the other components of z, since the ordering $\{1,\ldots,p\}$ is arbitrary. This permits sequential instantiation of each z_i without adjusting probabilistic relationships of other components of z. Under these independence assumptions, other discrete-time filter implementations offer the choice of either updating x using p scalar measurement updates, or incorporating all p components of z in a single batch [1]. The choice of updating scheme is determined by considerations of computational efficiency in those implementations. In the case of influence diagram processing, sequential reversal and instantiation of the p scalar nodes is equivalent to scalar update processing for other filters.

2.4 Removal and Reversal Between Vector Nodes

In this section, algorithms are presented to perform two basic operations necessary for discrete-time filter implementation, reversal of arcs between vector nodes, and removal of a vector node into another vector node. Reversal is the influence diagram version of Bayes' rule, and is the main step in performing measurement updates. Removal is the elimination of a random vector by expectation from the conditioning set of a conditional probability function, and is used to perform time updates in filtering. Referring to Fig. 2-5, assume the vector nodes x_0, x_1, and x_2 are ordered with arc coefficient matrix

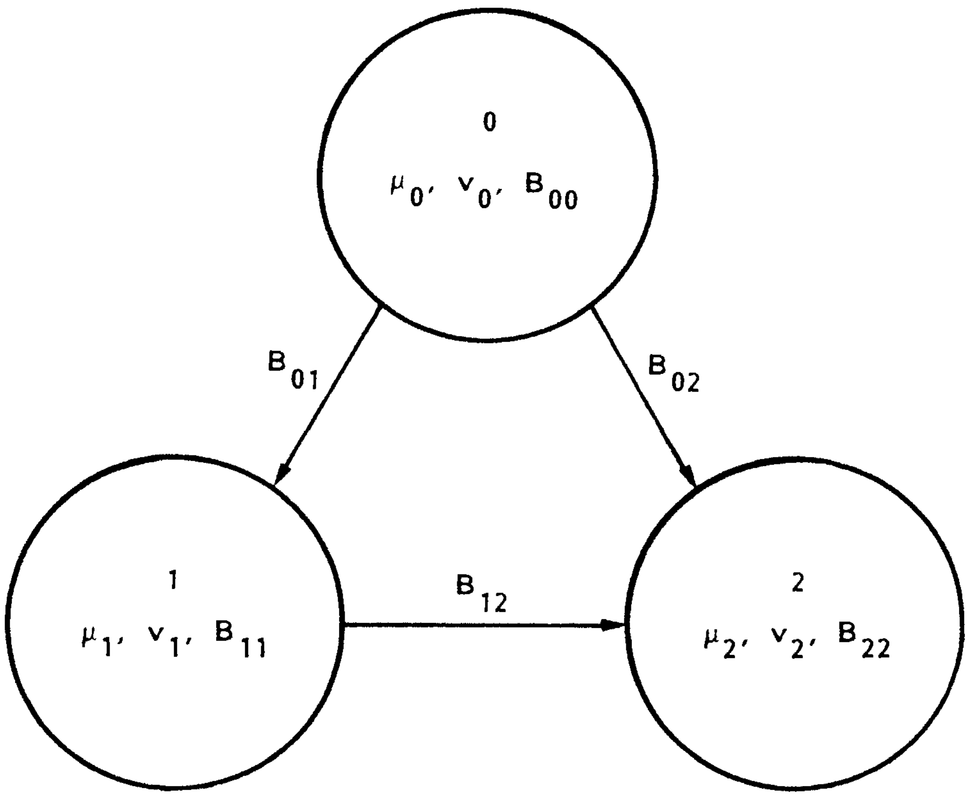


Fig. 2-5 Influence Diagram for Vector Removal and Reversal

$$B = \begin{bmatrix} B_{00} & B_{01} & B_{02} \\ 0 & B_{11} & B_{12} \\ 0 & 0 & B_{22} \end{bmatrix} .$$

The number of components of vector node x_i is n_i for $i = 0,1,2$; and the dimension of B_{ij} is $n_i \times n_j$ for $i = 0,1,2$ and $j = 0,1,2$. The components of each x_i are ordered such that the matrices B_{ii} for $i = 0,1,2$ are strictly upper triangular. For example, to reverse arcs from $x(k)$ to $x(k+1)$ in Fig. 2-3, the following assignments would be made:

$x_0 \leftarrow w(k)$,

$x_1 \leftarrow x(k)$,

$x_2 \leftarrow x(k+1)$, and

$$B \leftarrow \begin{bmatrix} 0 & 0 & \Gamma \\ 0 & 0 & \Phi \\ 0 & 0 & A \end{bmatrix} .$$

Let $v^T = (v_0^T, v_1^T, v_2^T)$ be the $(n_0+n_1+n_2)$-dimensional vector of conditional variances of $x^T = (x_0^T, x_1^T, x_2^T)$. The procedure for reversal of x_1 with x_2 is as follows:

```
Procedure Vreverse(B,v,n0,n1,n2)
Do i = n1, 1, -1               ; loop through x1 in reverse order
  Do j = 1, n2                 ; reverse with {x2(1),...,x2(n2)}
    Reverse scalar node x1(i) with scalar node x2(j), given
    conditioning variables {x0(1),...,x0(n0)}; {x1(1),...,x1(i-1)};
    and {x2(1),...,x2(j-1)}.
  End Do
End Do
```

Removal of x_1 into x_2 is performed by first reversing x_1 with elements of $\{x_2(1),\ldots,x_2(n_2-1)\}$ and then removing each component of x_1 into $x_2(n_2)$ as follows:

Procedure Vremove(B,v,n_0,n_1,n_2)

Call Vreverse (B,v,n_0,n_1,n_2-1) ; reverse x_1 with $\{x_2(1),\ldots,x_2(n_2-1)\}$

Do $i = n_1$, 1, -1 ; remove $\{x_1(n_1),\ldots,x_1(1)\}$ into $x_2(n_2)$

Remove scalar node $x_1(i)$ into scalar node $x_2(n_2)$, given conditioning variables $\{x_0(1),\ldots,x_0(n_0)\}$; $\{x_1(1),\ldots,x_1(i-1)\}$; and $\{x_2(1),\ldots,x_2(n_2-1)\}$.

End Do

Reversal and removal involving pairs of scalar nodes and their conditioning variables was detailed in Chapter 1. Appendix 2.A contains a complete listing of vector reversal and removal algorithms.

2.5 Measurement Update Algorithm

Referring back to Fig. 2-2, let

z = observed scalar measurement,

v = scalar measurement error,

τ = variance of v,

x = n-dimensional state vector,

$\mu = E[x]$,

σ = (n+1)-dimensional vector of conditional variances of $(x,z)^T$,

$B_{11} = A = n \times n$ strictly upper triangular matrix of arc coefficients for x,

$B_{12} = [h_1,\ldots,h_n]^T$ = n×1 matrix of arc coefficients from x to z,

B_{21} = 1×n matrix of zeros,

B_{22} = 1×1 zero matrix,

and

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

The algorithm for performing a scalar measurement update of x in Fig. 2-2 is a reversal and instantiation of z with x as follows:

```
e = z - B12^T μ                 ; prediction residual
σ(n+1) = τ                      ; removal of v into  z
Call Vreverse(B,σ,0,n,1)        ; perform reversal via Bayes' rule
Do j = 1, n                     ; update E[x] (instantiation of z)
  Δ(j) = B21(j) * e             ; direct effect of residual on update
  Do i = 1, j-1
    Δ(j) = Δ(j) + B11(i,j) * Δ(i) ; indirect effect of predecessor
                                    on update
  End Do
  μ(j) = μ(j) + Δ(j)            ; update E[x_j]
End Do
```

For processes with time-invariant state vectors ($\Phi = I_n$) and no process noise ($\Gamma = 0$), only the measurement update step is required to process data. This corresponds to a sequential data processing implementation for the classical least squares problem.

Following removal of v into z, the vector reversal algorithm of Section 2.4 is applied. Figure 2-6 shows the sequence of scalar reversals involved. Note that there is no freedom to choose which arc from $\{h_1,\ldots,h_n\}$ to reverse first. Reversing any arc other than h_n will result in a directed cycle in the diagram, which is not permitted.

2.6 Time Update Algorithm

The influence diagram algorithm for performing time update of $x(k)$ and $w(k)$ with $x(k+1)$ is performed in three steps:

(1) update of the unconditional mean of $x(k+1)$ using standard matrix multiplication, $\nu = \Phi\mu$,

(2) removal of $x(k)$ into $x(k+1)$, and

(3) removal of $w(k)$ into $x(k)$.

Generalizing from Fig. 2-3, let

$x(k)$ = n-dimensional state vector for stage k,

$\mu = E[x(k)|z(0),\ldots,z(k)]$,

Φ = $n\times n$ state transition matrix from $x(k)$ to $x(k+1)$,

$w(k)$ = r-dimensional process noise vector ($E[w(k)] = 0$),

ω = vector of conditional variances of $w(k)$,

Γ = $n\times r$ process noise transformation matrix,

$x(k+1)$ = n-dimensional state vector for stage k+1,

σ = vector of influence diagram conditional variances of $x(k)$,

A = $n\times n$ strictly upper triangular matrix of arc coefficients for $x(k)$,

$\nu = E[x(k+1)|z(0),\ldots,z(k)]$ to be computed,

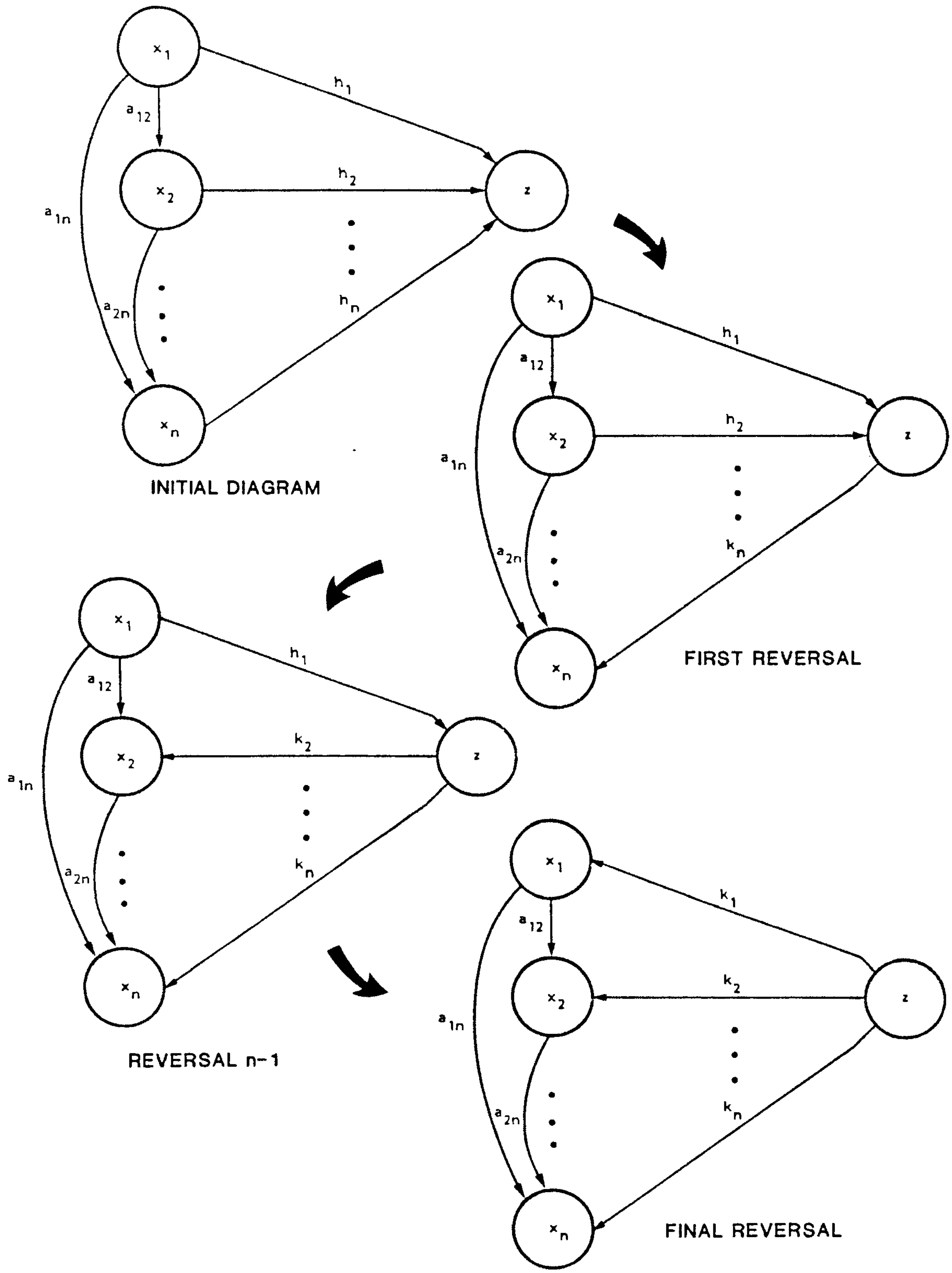


Fig. 2-6 Reversal Steps for Measurement Update

τ = vector of influence diagram conditional variances of x(k+1) to be computed, and

B = n×n matrix of arc coefficients for x(k+1) to be computed.

Removal of x(k) into x(k+1) is as follows:

$D_{00} = 0$; process noise vector components are uncorrelated

$D_{01} = 0$; x(k) is independent of w(k)

$D_{02} = \Gamma$; process noise mapping onto x(k+1)

$D_{11} = A$; dependence among elements of x(k)

$D_{12} = \Phi$; state vector mapping from period k to k+1

$D_{22} = 0$; initially elements of x(k+1) are deterministic and uncorrelated

$v_0 = \omega$; process noise variances

$v_1 = \sigma$; conditional variances of x(k)

$v_2 = 0$; x(k+1) is deterministic before removal

Call Vremove(D,v,r,n,n).

Removal of w(k) into x(k+1) uses the outputs of removal of x(k) into x(k+1). Let

$$D = \begin{bmatrix} D_{01} & D_{02} \\ 0 & D_{22} \end{bmatrix}$$

and $v^T = (v_1^T, v_2^T)$, with all submatrices and subvectors outputs from removal of x(k) into x(k+1). Removal of w(k) into x(k+1) is completed as follows:

Call Vremove(D,v,0,r,n) ; remove w(k) into x(k+1)

$\tau = v_1$

$B = D_{22}$.

In Fig. 2-7, removal of x(k) into x(k+1) begins with reversal of $x_3(k)$ and the deterministic node

$$x_1(k+1) = \phi_{11}x_1(k) + \phi_{21}x_2(k) + \phi_{31}x_3(k) + \gamma_{11}w_1(k) + \gamma_{21}w_2(k) + \gamma_{31}w_3(k). \tag{2.6.1}$$

If $\phi_{31} \neq 0$, $x_3(k)$ has a conditional variance of zero and is a deterministic node after reversal. The calculation of arc values from its conditioning nodes is equivalent to making $x_3(k)$ the dependent variable in the prior deterministic equation (2.6.1) for $x_1(k+1)$.

$$x_3(k) = [x_1(k+1) - \phi_{11}x_1(k) - \phi_{21}x_2(k) - \gamma_{11}w_1(k) - \gamma_{21}w_2(k) - \gamma_{31}w_3(k)]/\phi_{31}. \tag{2.6.2}$$

Prior to reversal, $x_3(k)$ does not have w(k) as a conditioning variable. After reversal, all three elements of w(k) have an influence on $x_3(k)$, as shown in equation (2.6.2). These arcs from w(k) remain active in the diagram until removal of $x_3(k)$ into $x_3(k+1)$, when their influence is added to Γ, the coefficients for the influence of w(k) on x(k+1). During reversal of $x_3(k)$ with $x_1(k+1)$, arc values from $x_m(k)$ to $x_3(k)$ for m = 1,2 are calculated via three steps:

$$\phi_{m1} \leftarrow \phi_{m1} + a_{m3} * \phi_{31},$$
$$\phi_{31} \leftarrow 1 / \phi_{31}, \text{ and}$$
$$a_{m3} \leftarrow a_{m3} - \phi_{m1} * \phi_{31}.$$

Combining these steps,

$$a_{m3} \leftarrow a_{m3} - [\phi_{m1} + a_{m3} * \phi_{31}] * [1/\phi_{31}] = -\phi_{m1}/\phi_{31}.$$

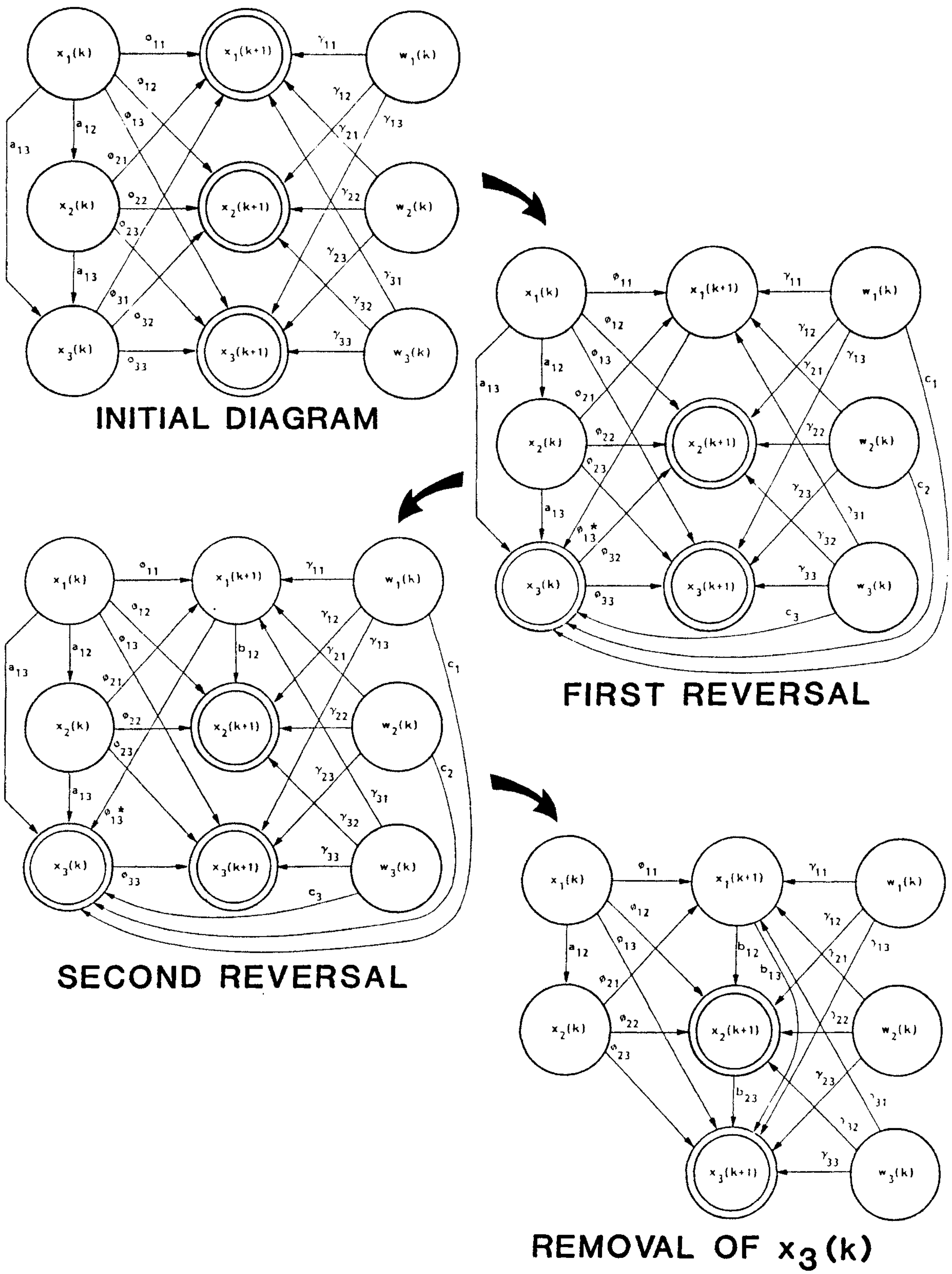


Fig. 2-7 Operations for Removal of $X_3(k)$

For this special case, the probabilistic reversal algorithm is mathematically equivalent to a deterministic Gaussian elimination algorithm [14, Chapter 1]. Processing of such a reversal using Gaussian elimination would be slightly more efficient. The reversal process is an order n^3 algorithm (see Section 2.10), and the opportunity to use Gaussian elimination occurs at most n-1 times. If it is used, the cost of added logical complexity and program storage may exceed any gain in processing speed.

Note that the second reversal in Fig. 2-7 involving $x_3(k)$ and $x_2(k+1)$ results in the coefficient ϕ^*_{23} being equal to zero, because $x_3(k)$ is already deterministic before reversal. As a consequence, no updates are needed for the coefficients on the arcs to $x_3(k)$ from its conditioning nodes. As will be shown in Section 2.10, the computational efficiency of influence diagram processing compares favorably with other algorithms. A major source of efficiency for influence diagram processing comes from exploiting reversals involving deterministic nodes.

2.7 Colored Process Noise Time Update Algorithm

The important special case of discrete-time filtering with colored process noise is shown in Fig. 2-8. The dynamic process is

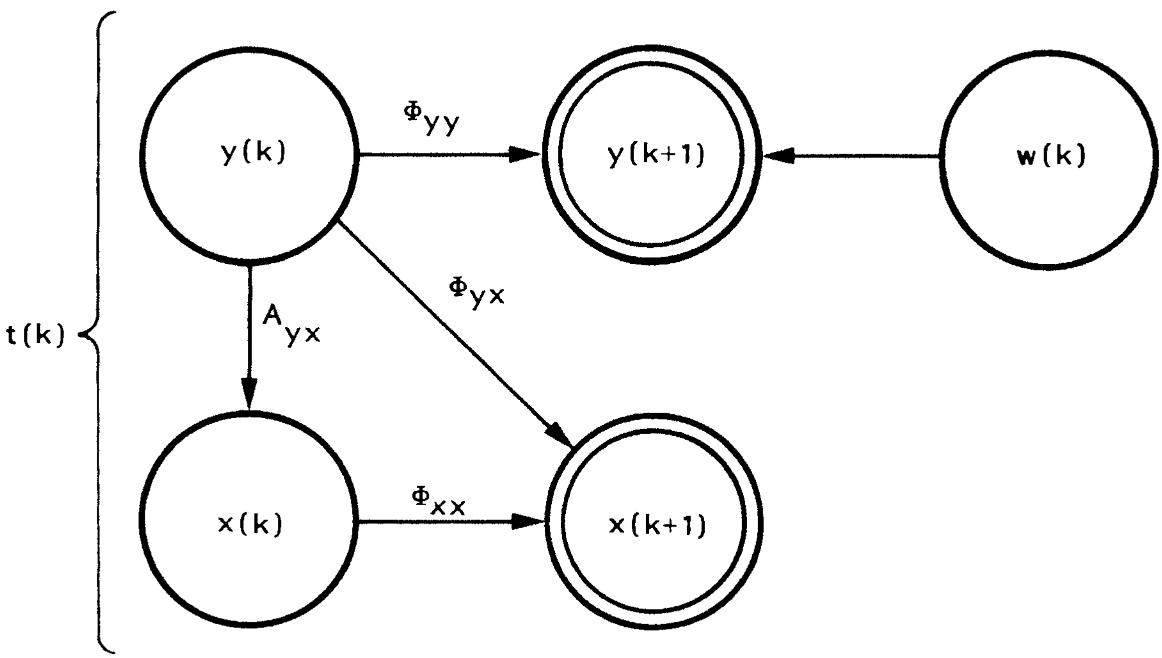
$$x(k+1) = \Phi_{yx}(k) + \Phi_{xx}x(k) \text{ , and}$$

$$y(k+1) = \Phi_{yy}y(k) + w(k),$$

where

$$x(k) \text{ and } x(k+1) \in R^n,\ y(k) \text{ and } y(k+1) \in R^r, \text{ and } w(k) \in R^r.$$

Also, Φ_{yy} is assumed to be diagonal. The probabilistic structure is:



Q_k = COV [W(k)], DIAGONAL

Φ_{yy}, DIAGONAL

Fig. 2-8 Colored Noise Time Update

$E[t(k)] = \mu$, where $t(k)^T = (y(k)^T, x(k)^T)$;

$Var[t(k)] = P_k$;

$E[w(k)] = 0$;

$Var[w(k)] = diag(\omega_1, \ldots, \omega_r)$;

$Cov[x(k), w(k)] = 0$;

$Cov[y(k), w(k)] = 0$.

For efficient influence diagram processing, the ordering of $x(k)$ and $y(k)$ is important. All elements of $y(k)$ are assumed to be predecessors of $x(k)$. Thus, the vector $t(k)^T$ is defined to be $(y(k)^T, x(k)^T)$ instead of $(x(k)^T, y(k)^T)$. Define

$$\Phi = \begin{bmatrix} \Phi_{yy} & \Phi_{yx} \\ 0 & \Phi_{xx} \end{bmatrix}$$

an $(r+n)\times(r+n)$ state transition matrix,

$$A = \begin{bmatrix} A_{yy} & A_{yx} \\ 0 & A_{xx} \end{bmatrix}$$

an $(r+n)\times(r+n)$ triangular matrix of arc coefficients for $t(k)$,

σ = an $(r+n)$-vector of influence diagram conditional variances of $t(k)$,

w = r-dimensional process noise vector ($E[w] = 0$),

ω = vector of variances of $w(k)$,

ν = $E[t(k+1)]$, an $(r+n)$-vector to be computed,

$\tau^T = (\tau_y^T, \tau_x^T)$, an (r+n)-vector of influence diagram conditional variances of t(k+1) to be computed, and

C = (r+n)×(r+n) strictly upper triangular matrix of arc coefficients for t(k+1) to be computed.

The algorithm for performing time update of t(k) and w(k) with t(k+1) is performed in three steps:

(1) update of the unconditional mean of t(k+1) using matrix multiplication, $\nu = \Phi\mu$,

(2) removal of w(k) into y(k+1), and

(3) removal of t(k) into t(k+1) .

Removal of w(k) into y(k+1) is a simple replacement of the conditional variance of y(k+1) with the variances of w(k), $\tau_y = \omega$.

Let

$$B = \begin{bmatrix} A & \Phi \\ 0 & 0 \end{bmatrix},$$

a (2r+2n)x(2r+2n) matrix. Let $\tau_x = 0$, and $v^T = (\sigma^T, \tau^T)$. Removal of t(k) into t(k+1) is as follows:

Call Vremove (B,v,0,r+n,r+n)

$C = B_{22}$.

2.8 Noninformative Prior for x(0)

When little information about x(0) is available or a sequential least squares implementation is preferred, the standard approach is to derive the filter assuming either P_0 is a diagonal matrix with diagonal elements infinitely large or P_0^{-1} is an $n \times n$ zero matrix. For influence diagram processing, the assumption of noninformative prior knowledge affects only the conditional variance update for removals, and both the conditional variance and arc updates for reversals. For removal of scalar node x_i with conditional variance $v_i = \infty$ into scalar node x_j, the conditional variance update is simply $v_j = \infty$. For reversal of two nodes with at least one of the nodes having a noninformative density, the conditional variance and arc updates are as follows:

```
b_ji = 1/b_ij
If ((v_i = ∞) and (v_j ≠ ∞))
     v_i = v_j * b_ji * b_ji
End If
v_j = ∞
b_ij = 0.
```

2.9 Bias Errors and Consider Filters

A bias error arises from a measuring device with a time-invariant, additive error. An influence diagram model of a bias error is shown in Fig. 2-9, which is an adaptation of Fig. 2-1 with a bias node β added.

Since the bias is a predecessor of the measurement $z(0)$, the arc from β to $z(0)$ must be reversed to update our state of information when $z(0)$ is realized (Fig. 2-10). Instantiation of $z(0)$ requires an update of the means of both $x(0)$ and β, which have an arc between them after reversal and become dependent random variables once we have observed $z(0)$.

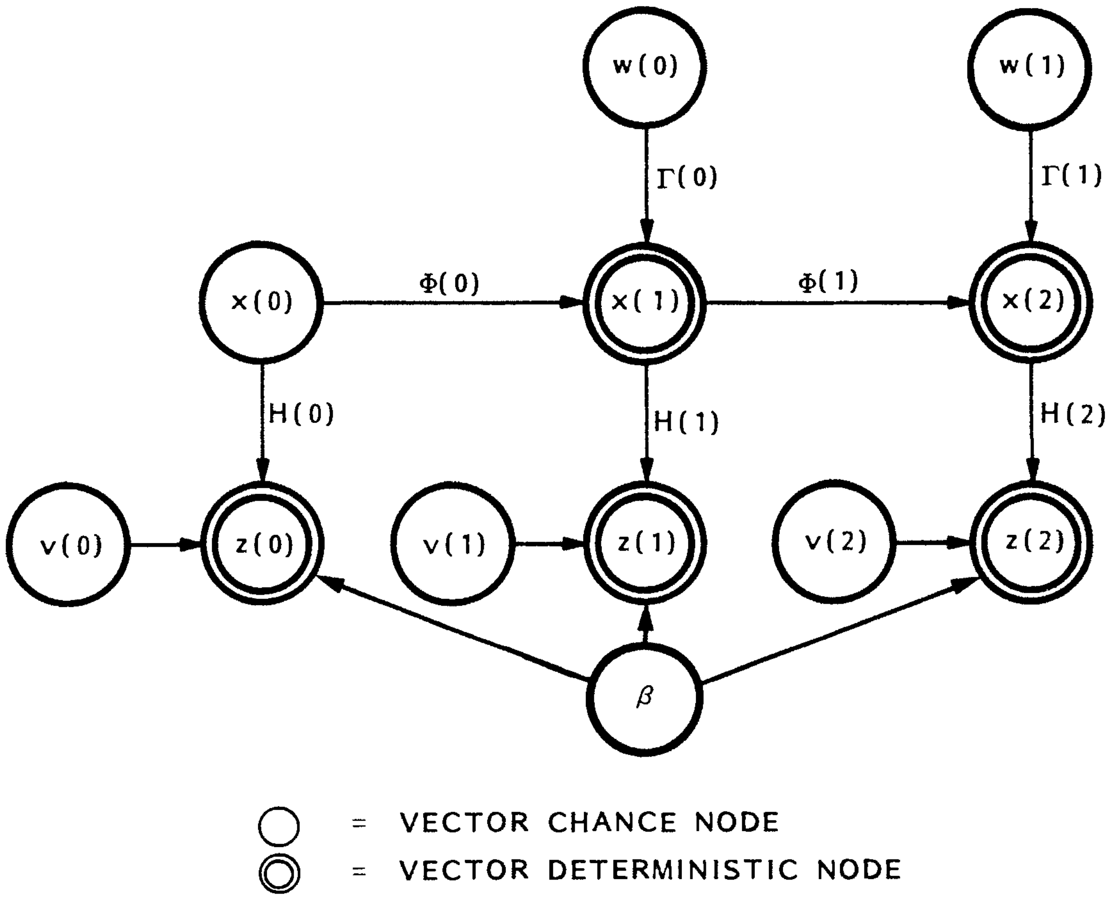
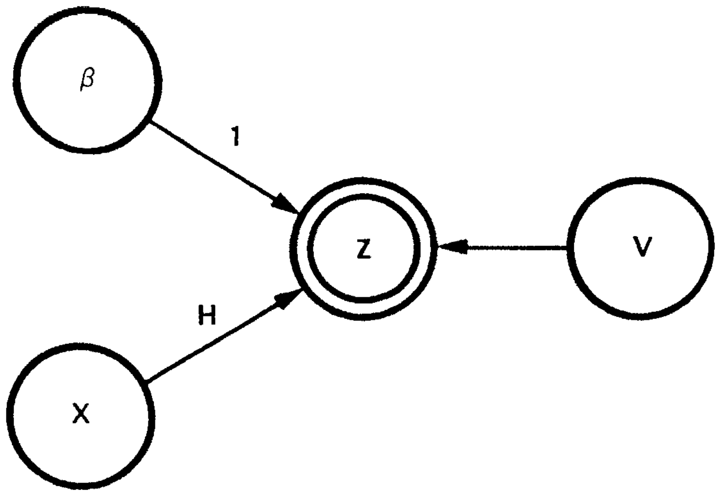
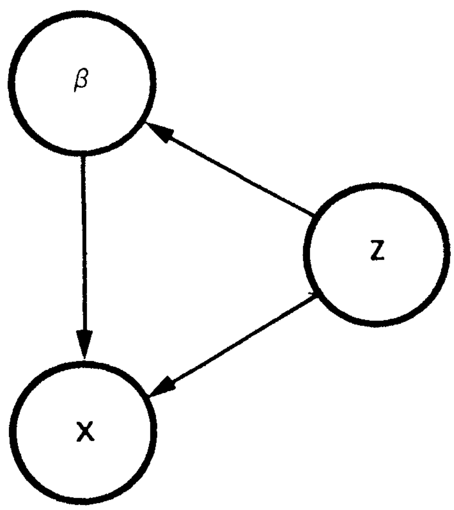


Fig. 2-9 Influence Diagram Representation of Bias



LIKELIHOOD DIAGRAM



POSTERIOR DIAGRAM

Fig. 2-10 Measurement Update With Bias

The standard procedure for bias estimation in discrete-time filtering is to augment the state vectors $x(k)$ with $\beta(k) = \beta$ for $k = 0,\ldots,N$ (Fig. 2-11). The measurement sensitivity matrix $H(k)$ is augmented as follows:

$$\begin{bmatrix} H(k) \\ I_p \end{bmatrix} .$$

The state transition matrix from time k to time k+1 is:

$$\begin{bmatrix} \Phi(k) & 0 \\ 0 & I_p \end{bmatrix} .$$

Bias estimation is not a significant computational burden for inference about a single state vector and a single source of bias. Estimating many state vectors using observations from many sources of bias can become computationally overwhelming. Consider the case of two observers and two tracks with time-invariant state vectors. Figure 2-12 shows how correlation arises between the first observer's bias and the first track's state vector after Observer 1 sees Track 1. In Fig. 2-13, Observer 2 sees Track 1. At this point, both observer biases are correlated, and we must update our estimates of β_1 and β_2 when z_2 is realized. Track 2 is first observed by Observer 1 in Fig. 2-14. Nodes β_2 and s_1 do not have a direct dependence on z_3, and it appears that no update of their means is required. This is not the case, as all direct and indirect successors of z_3 must be updated. Figure 2-15 shows addition of correlation between β_2 and s_2 when Observer 2 sees Track 2. From

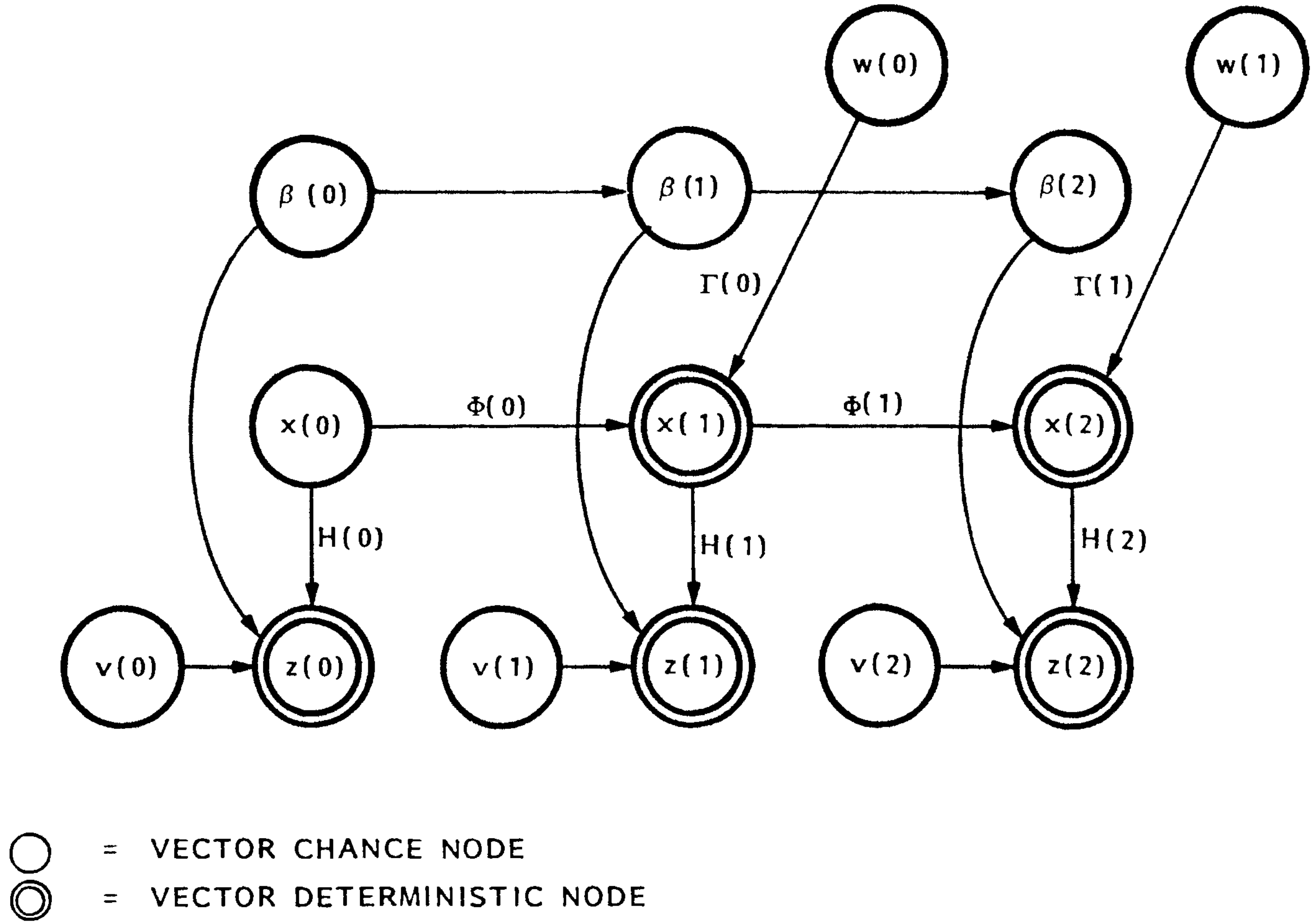
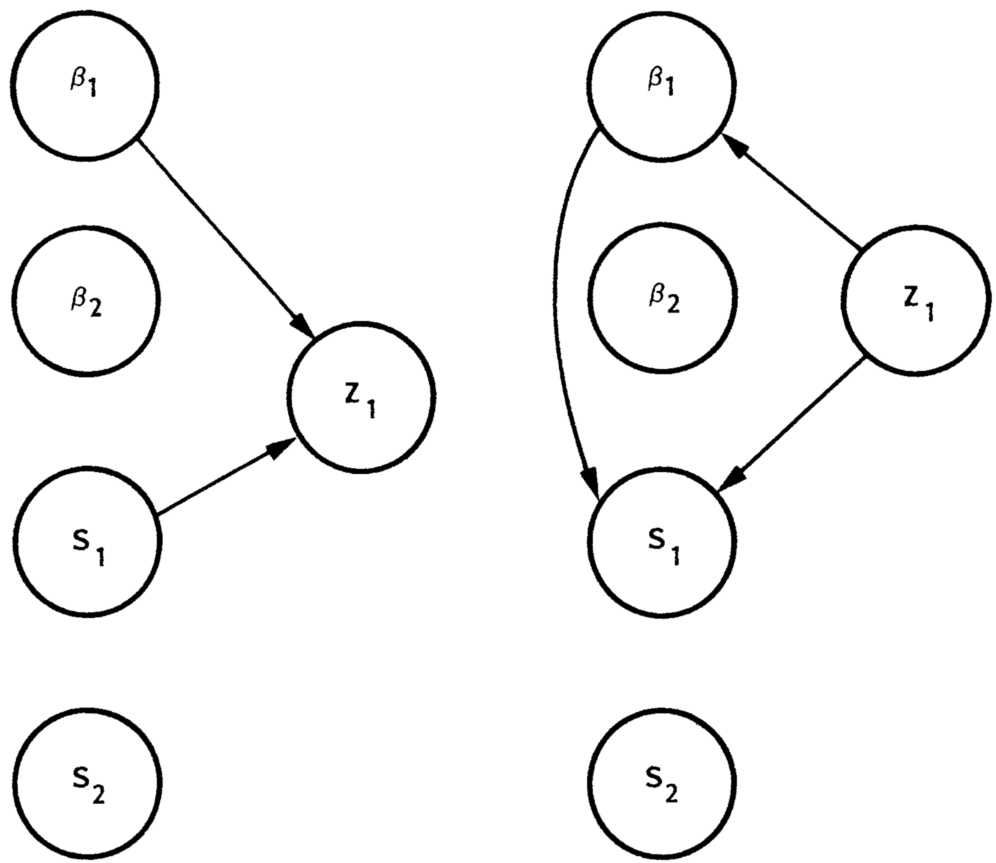
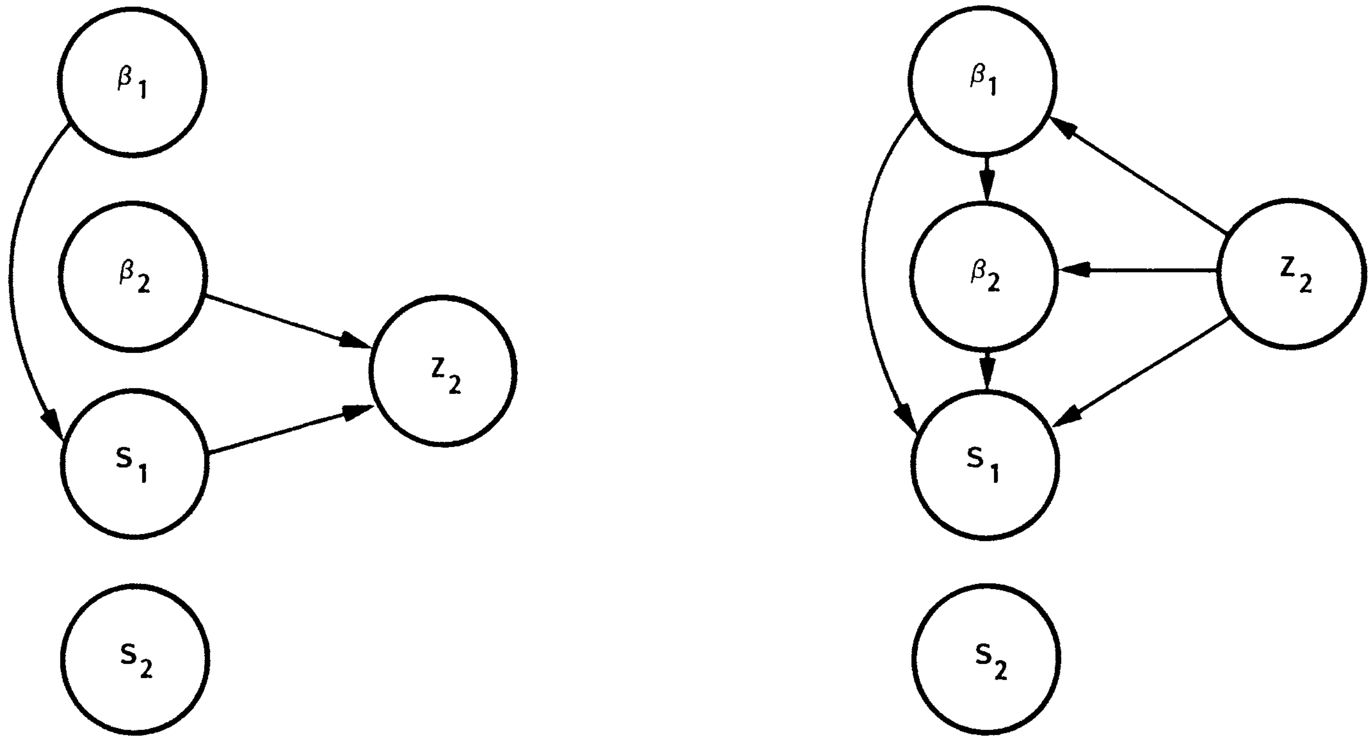


Fig. 2-11 Augmented State Vector With Bias



LIKELIHOOD DIAGRAM   POSTERIOR DIAGRAM

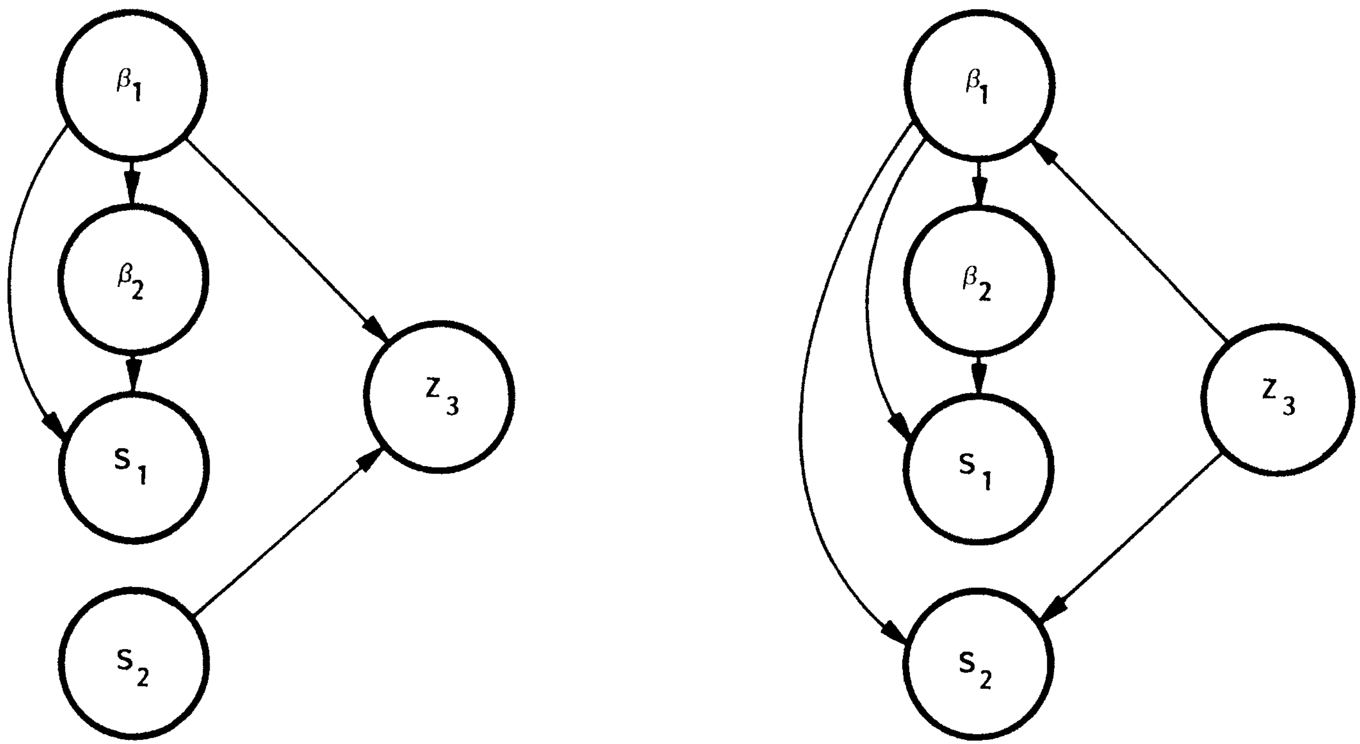
Fig. 2-12   Observer 1 Sees Track 1



LIKELIHOOD DIAGRAM

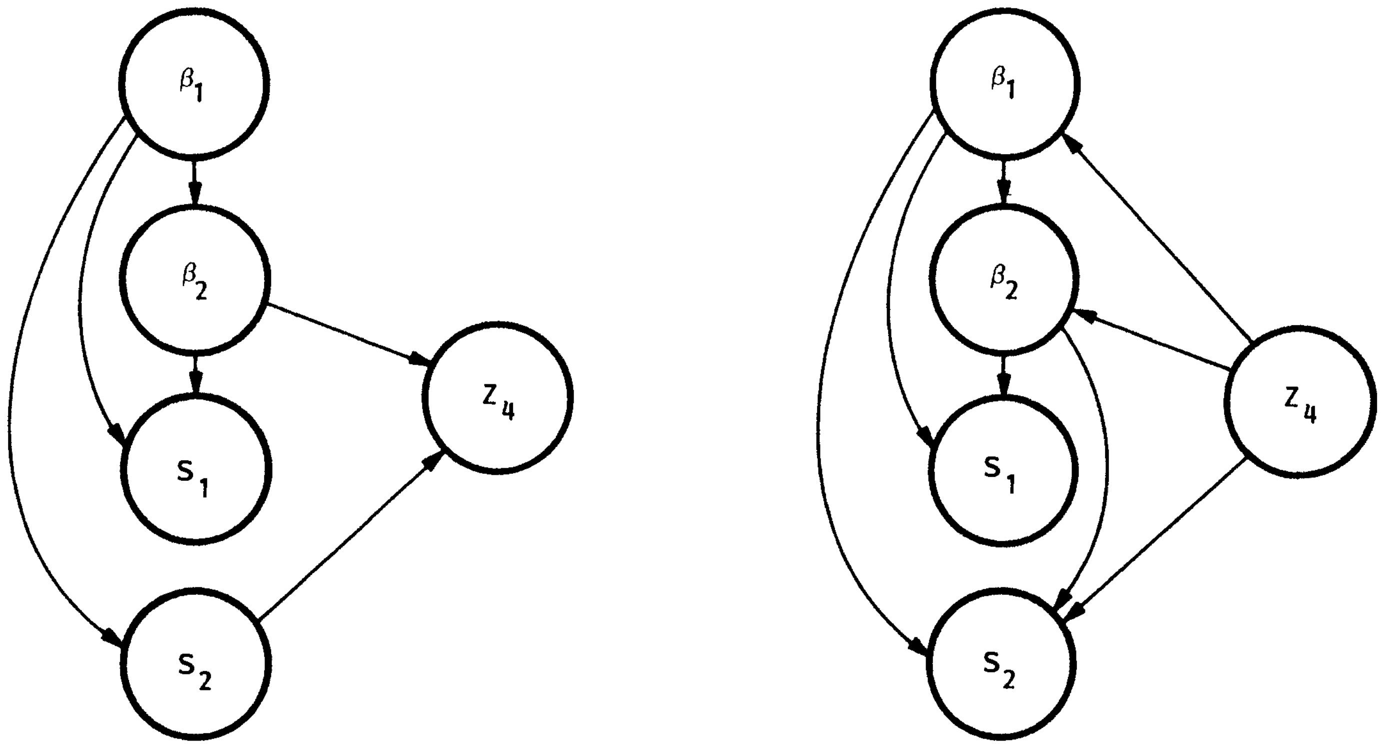
POSTERIOR DIAGRAM

Fig. 2-13 Observer 2 Sees Track 1



LIKELIHOOD DIAGRAM　　　　POSTERIOR DIAGRAM

Fig. 2-14　Observer 1 Sees Track 2



LIKELIHOOD DIAGRAM

POSTERIOR DIAGRAM

Fig. 2-15 Observer 2 Sees Track 2

this point forward in time, any observation requires us to update β_1, β_2, s_1, and s_2.

To avoid the additional calculation needed for proper treatment of biases in multitarget, multiobserver tracking, heuristic consider filters have been proposed that constrain the bias component to be independent of the state vector for the likelihood model of each observation [1, pp. 165-71]. Figure 2-16 is an influence diagram model equivalent to consider filter models. $\beta(k)$ for $k = 0,1,2$ are independent, identically distributed bias errors. The measurement update for Schmidt's consider filter is implemented in Fig. 2-17. Schmidt's approach treats bias as a source of random measurement noise in addition to $v(k)$, the original measurement noise. The measurement update for the standard consider filter is shown in Fig. 2-18. The standard consider filter updates the mean of the state vector as if bias were not present. After updating the mean, the covariance matrix of the state vector is updated as if the bias had been estimated. Curkendall's consider filter is the same as the standard consider filter, except the covariance matrix is multiplied by a factor $\phi > 1$ before performing the operations in Fig. 2-18. For the influence diagram implementation of Curkendall's filter, the conditional variances of the influence diagram factorization of $x(0)$ are multiplied by ϕ.

2.10 Comparison of Operation Counts With Other Filtering Implementations

Table 2-1, an extension of Bierman's work [1, pp. 82-90], presents instruction counts for various processing techniques used for measurement updates. Table 2-2 presents weighted operation counts for UNIVAC

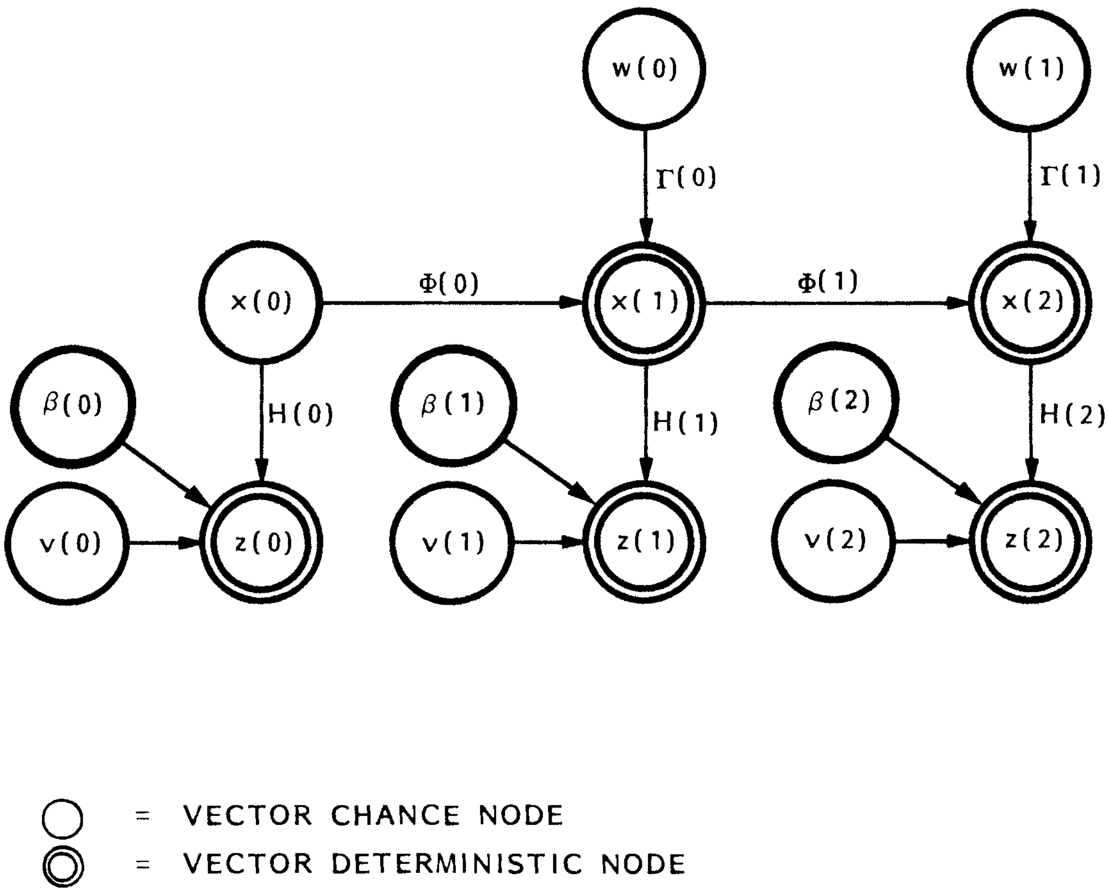


Fig. 2-16 Consider Filtering Influence Diagram

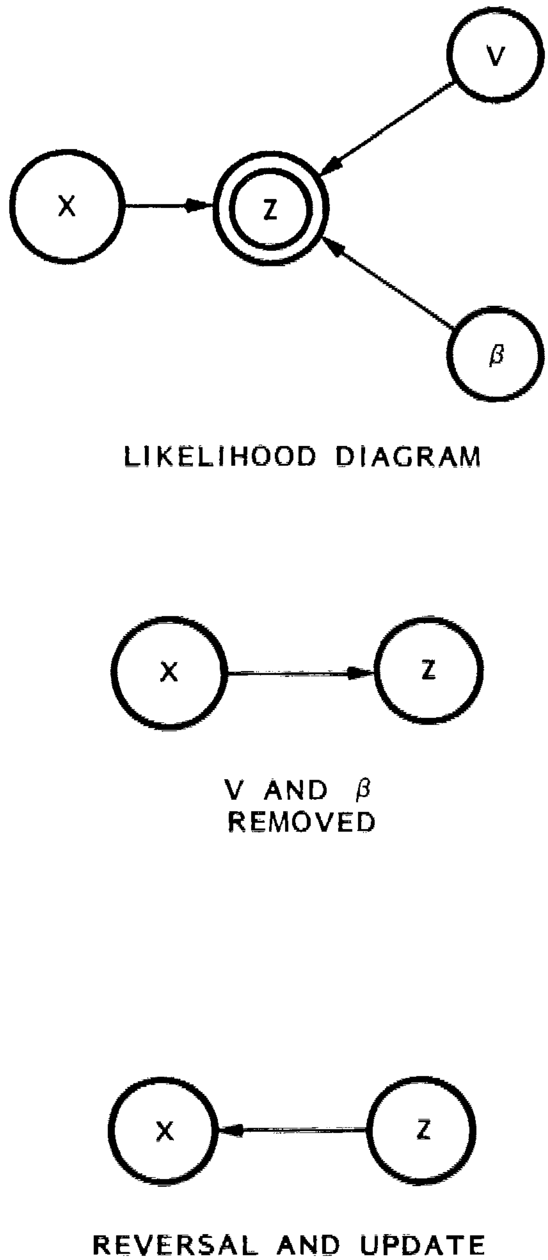


Fig. 2-17 Measurement Update for Schmidt Consider Filter

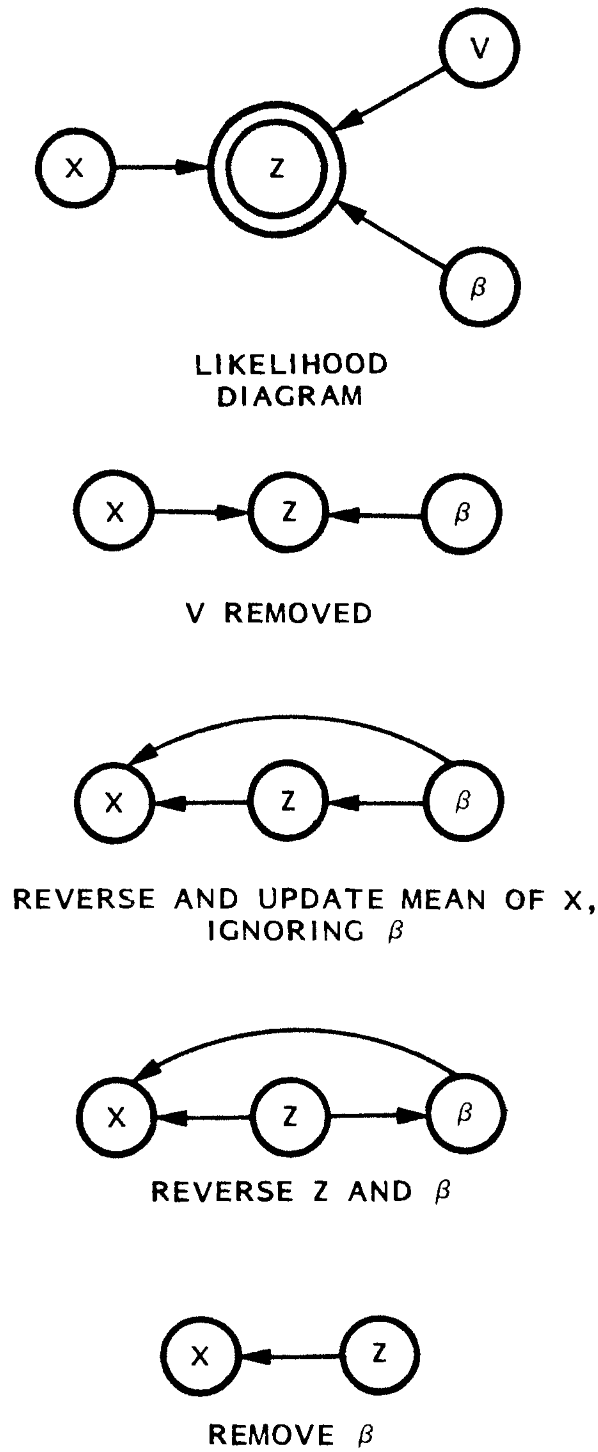


Fig. 2-18 Measurement Update for Standard Consider Filter

Table 2-1 OPERATION COUNTS FOR PROCESSING A VECTOR OF p MEASUREMENTS

	ALGORITHM	ADDITIONS	MULTIPLICATIONS	DIVISIONS	SQUARE ROOTS
SCALAR PROCESSING ALGORITHMS	INFLUENCE DIAGRAM	$(1.5n^2 + 1.5n)p$	$(1.5n^2 + 4.5n)p$	np	0
	CONVENTIONAL KALMAN	$(1.5n^2 + 3.5n)p$	$(1.5n^2 + 4.5n)p$	p	0
	U-D COVARIANCE	$(1.5n^2 + 1.5n)p$	$(1.5n^2 + 5.5n)p$	np	0
	SQUARE ROOT COVARIANCE	$(1.5n^2 + 3.5n)p$	$(2n^2 + 5n)p$	$2np$	np
	POTTER SQUARE ROOT	$(3n^2 + 3n)p$	$(3n^2 + 4n)p$	$2p$	p
	KALMAN STABILIZED	$(4.5n^2 + 5.5n)p$	$(4n^2 + 7.5n)p$	p	0
BATCH PROCESSING ALGORITHMS	SRIF, R TRIANGULAR	$(n^2 + 2n)p+$ $1.5n^2 + 2.5n$	$(n^2 + 3n)p+$ $2n^2 + 3n$	$2n$	n
	NORMAL EQUATION	$(0.5n^2 + 1.5n)p+$ $(n^3 + 6n^2 + 5n)/6$	$(0.5n^2 + 2.5n)p$ $(n^3 + 9n^2 - 4n)/6$	$2n$	n
	SRIF, R GENERAL	n^2p+ $(4n^3 + 3n^2 + 5n)/6$	n^2p+ $(4n^3 + 9n^2 - n)/6$	$2n$	n

Table 2-2 WEIGHTED OPERATION COUNTS FOR PROCESSING A VECTOR OF p MEASUREMENTS

ALGORITHM	WEIGHTED OPERATION COUNTS
INFLUENCE DIAGRAM	$(3.6n^2 + 12.3n)p$
CONVENTIONAL KALMAN	$(3.6n^2 + 9.8n + 4.5)p$
U-D COVARIANCE	$(3.6n^2 + 13.7n)p$
SQUARE ROOT COVARIANCE	$(4.3n^2 + 40.9n)p$
POTTER SQUARE ROOT	$(7.2n^2 + 8.6n + 30.4)p$
KALMAN STABILIZED	$(10.1n^2 + 16n + 4.5)p$
SRIF, R TRIANGULAR	$(2.4n^2 + 6.2n)p + 4.3n^2 + 37.1n$
NORMAL EQUATION	$(1.2n^2 + 5.0n)p + 0.4n^3 + 3.1n^2 + 30.3n$
SRIF, R GENERAL	$2.4n^2p + 1.6n^3 + 2.6n^2 + 31n$

OPERATION	WEIGHT
+	1
X	1.4
÷	4.5
$\sqrt{\ }$	21.4

1108 floating point operations, where the weightings are unity for addition, 1.4 for multiplication, 4.5 for division, and 21.4 for square roots. The last three algorithms incorporate all p observations in a single batch and, thus, have overhead operation counts that are not multiplied by p. In real-time applications, where the estimates must be computed for N batches of p-dimensional measurement vectors, the overhead operations must be repeated N times, and the batch algorithms become less efficient. Of the scalar measurement processing algorithms shown in Table 2-2, the influence diagram approach is second only to the conventional Kalman filter. Its order n^2p instruction count coefficient is the same as the U-D filter, as both techniques use a triangular factorization of the prior covariance matrix of the state vector.

Instruction counts for time updates using influence diagram processing and implementations studied by Thornton and Bierman [15] are given in Table 2-3. For influence diagram processing, we assume that all relevant elements of A, Φ, and Γ in Fig. 2-3 are nonzero. Weighted operation counts are given in Table 2-4. Influence diagram processing has the smallest n^3 coefficient, but has a larger order n^2r coefficient and an order nr^2 term that does not appear in the other implementations. The larger coefficient for the n^2r term results from adding temporary arcs from elements of w(k) to x(k) during removal of x(k) into x(k+1). Similarly, the additional nr^2 term is caused by adding temporary correlations between elements of w(k) during removal of w(k) into x(k+1). Thornton and Bierman also analyze time update processing for correlated process noise. Extension of their results to include influence diagram processing is shown in Tables 2-5 and 2-6. Of the

Table 2-3 OPERATION COUNTS FOR TIME UPDATE WITH WHITE PROCESS NOISE

ALGORITHM	ADDITIONS	MULTIPLICATIONS	DIVISIONS	SQUARE ROOTS
INFLUENCE DIAGRAM	$1.17n^3 + 0.17n^2 - 3n + 3 +$ $(2.5n^2 - 2.5n + 1)r +$ $(n - 1)r^2$	$1.17n^3 + 1.3n^2 - 4.5n + 5 +$ $(2.5n^2 + 0.5n - 1)r +$ $(n - 1)r^2$	$0.5n^2 + 0.5n +$ $(n - 1)r$	0 0
CONVENTIONAL KALMAN	$1.5n^3 + 2n^2 + 0.5n +$ $(0.5n^2 + 0.5n)r$	$1.5n^3 + 1.5n^2 +$ $(0.5n^2 + 1.5n)r$	0	0
U-D COVARIANCE	$1.5n^3 + 0.5n^2 +$ n^2r	$1.5n^3 + 2.5n^2 - n +$ $(n^2 + 2n - 1)r$	$n - 1$	0
SQUARE ROOT COVARIANCE	$1.7n^3 + 2n^2 + 0.3n +$ $(n^2 + n)r$	$1.7n^3 + 2n^2 + 0.3n +$ $(n^2 + n)r$	n	n

Table 2-4 WEIGHTED OPERATION COUNTS FOR TIME UPDATE WITH WHITE PROCESS NOISE

ALGORITHM	WEIGHTED OPERATION COUNTS
INFLUENCE DIAGRAM	$2.8n^3 + 3.95n^2 - 11.55n + 10 + (6n^2 + 2.7n - 4.9)r + (2.4n - 2.4)r^2$
CONVENTIONAL KALMAN	$3.6n^3 + 4.1n^2 + 0.5n + (1.2n^2 + 2.6n)r$
U-D COVARIANCE	$3.6n^3 + 4n^2 + 3.1n - 4.5 + (2.4n^2 + 4.2n - 2.8)r$
SQUARE ROOT COVARIANCE	$4n^3 + 4.8n^2 + 26.7n + (2.4n^2 + 2.4n)r$

Table 2-5 OPERATION COUNTS FOR TIME UPDATE WITH COLORED PROCESS NOISE

ALGORITHM	ADDITIONS	MULTIPLICATIONS	DIVISIONS	SQUARE ROOTS
INFLUENCE DIAGRAM	$1.17n^3 + 0.17n^2 - 3n + 1 +$ $(2.5n^2 - 0.5n - 2)r +$ $(3n - 1)r^2 +$ r^3	$1.17n^3 + 1.3n^2 - 4.5n + 5 +$ $(2.5n^2 + 3.5n - 3.5)r +$ $(3n + 0.5)r^2 +$ r^3	$0.5n^2 + 0.5n - 1 +$ $(n - 0.5)r +$ $0.5r^2$	0
CONVENTIONAL KALMAN	$1.5n^3 + 1.5n^2 +$ $(2.5n^2 + 1.5n + 1)r +$ nr^2	$1.5n^3 + 1.5n^2 +$ $(2.5n^2 + 2.5n + 2)r +$ $(n + 1)r^2$	0	0
U-D COVARIANCE	$1.5n^3 + 0.5n^2 +$ $(2n^2 + 1.2n)r +$ $(1.5n - 0.5)r^2 +$ $0.3r^3$	$1.5n^3 + 2.5n^2 - n +$ $(2n^2 + 2.5n + 4.5)r +$ $(1.5n + 2)r^2 +$ $0.3r^3$	$n - 1 +$ $(n + 0.5)r +$ $0.5r^2$	0
SQUARE ROOT COVARIANCE	$1.7n^3 + 2n^2 + 0.3n - 1 +$ $(3n^2 + 3n + 0.5)r +$ $(n + 4.5)r^2 +$ r^3	$1.7n^3 + 2n^2 + 0.3n - 1$ $(3n^2 + 4n + 1.5)r +$ $(n + 6.5)r^2 +$ r^3	$n + r$	$n + r$

Table 2-6 WEIGHTED OPERATION COUNTS FOR TIME UPDATE WITH COLORED PROCESS NOISE

ALGORITHM	WEIGHTED OPERATION COUNTS
INFLUENCE DIAGRAM	$2.8n^3 + 3.95n^2 - 11.55n + 10 + (6n^2 + 9.9n - 9.15)r + (7.2n - 1.95)r^2 + 2.4r^3$
CONVENTIONAL KALMAN	$3.6n^3 + 3.6n^2 + (6n^2 + 5n + 3.8)\ r + (2.4n + 1.4)r^2$
U-D COVARIANCE	$3.6n^3 + 4n^2 + 3.1n - 4.5 + (4.8n^2 + 7.5n + 9)r + (3.6n + 4.6)r^2 + 0.8r^3$
SQUARE ROOT COVARIANCE	$4n^3 + 4.8n^2 + 26.7n - 2.4 + (7.2n^2 + 8.6n + 27.5)r + (2.4n + 13.6)r^2 + 2.4r^3$

algorithms in Table 2-6, influence diagram processing is the most efficient algorithm when n is larger than r.

2.11 Conclusion

We have represented the complex discrete-time filtering problem with an intuitive graphical tool. Modelling issues and implementation approaches can be discussed using the influence diagram. In addition, computational efficiency of algorithms derived from graphical manipulation compares quite favorably with efficiency of current algorithms for discrete-time filtering.

APPENDIX 2.A

ALGORITHMS FOR REMOVAL AND REVERSAL BETWEEN VECTOR NODES

This appendix presents detailed algorithms for reversal of arcs between vector nodes and removal of a vector node into another vector node. Referring to Fig. 2-4, assume the vector nodes x_0, x_1, and x_2 are ordered with arc coefficient matrix

$$B = \begin{bmatrix} B_{00} & B_{01} & B_{02} \\ 0 & B_{11} & B_{12} \\ 0 & 0 & B_{22} \end{bmatrix}$$

The number of components of vector node x_i is n_i for $i = 0,1,2$; and the dimension of B_{ij} is $n_i \times n_j$ for $i = 0,1,2$ and $j = 0,1,2$. The components of each x_i are ordered such that the matrices B_{ii} for $i = 0,1,2$ are strictly upper triangular.

Let $v^T = (v_0^T, v_1^T, v_2^T)$ be the $(n_0+n_1+n_2)$-dimensional vector of conditional variances of $x^T = (x_0^T, x_1^T, x_2^T)$. The procedure for reversal of x_1 with x_2 is as follows:

```
Procedure Vreverse(B,v,n0,n1,n2)
Do i = n0+n1, n0+1, -1          ; loop through x1 in reverse order
  i' = i - n0                   ; element number within x1
  Do j = n0+n1+1, n0+n1+n2      ; reverse with {x2(1),...,x2(n2)}
    j' = j - (n0 + n1)          ; element number within x2
    If (B(i,j) = 0) Skip to next j
    Do k = 1, n0                ; update arcs from x0 to x1
```

```
  If (B(k,i) = 0) Skip to next k
  B(k,j) = B(k,j) + B(k,i) * B(i,j)
End Do
Do k = n0+1, i-1          ; update arcs from {x1(1),...,x1(i'-1)}
                            to x2(j')
  If (B(k,i) = 0) Skip to next k
  B(k,j) = B(k,j) + B(k,i) * B(i,j)
End Do
Do k = n0+n1+1, j-1       ; update arcs from {x2(1),...,x2(j'-1)}
                            to x1(i')
  If (B(k,i) = 0) Skip to next k
  B(k,j) = B(k,j) + B(k,i) * B(i,j)
End Do
If (v(i) = 0)             ; x1(i') deterministic
  B(j,i) = 0
Else
  If ((v(i) ≠ ∞) and (v(j) ≠ ∞))  ; standard distributions
    If (v(j) = 0)         ; x2(j') deterministic
      v(j) = B(i,j) * B(i,j) * v(i)
      v(i) = 0
      B(j,i) = 1 / B(i,j)
    Else        ; both nodes probabilistic
      vjold = v(j)
      v(j) = v(j) + B(i,j) * B(i,j) * v(i)
      vratio = v(i) / v(j)
      v(i) = vjold * vratio
```

```
            B(j,i) = B(i,j) * vratio
          End If
        Else                          ; noninformative distributions
          B(j,i) = 1 / B(i,j)
          If ((v(i) = ∞) and (v(j) ≠ ∞))
              v(i) = v(j) * B(j,i) * B(j,i)
            End If
            v(j) = ∞
          End If
        End If
        B(i,j) = 0
        Do k = 1, n0                  ; update arcs from x0 to x1
            If (B(k,j) = 0) Skip to next k
            B(k,i) = B(k,i) - B(k,j) * B(j,i)
        End Do
        Do k = n0+1, i-1              ; update arcs from {x1(1),...,
                                        x1(i'-1)} to x1(i')
          If (B(k,j) = 0) Skip to next k
          B(k,i) = B(k,i) - B(k,j) * B(j,i)
        End Do
        Do k = n0+n1+1, j-1           ; update arcs from {x2(1),...,
                                        x2(j'-1)} to x1(i')
        If (B(k,j) = 0) Skip to next k
        B(k,i) = B(k,i) - B(k,j) * B(j,i)
      End Do
    End Do
  End Do
```

Removal of x_1 into x_2 is performed by first reversing x_1 with elements of $\{x_2(1),\ldots,x_2(n_2-1)\}$ and then removing each component of x_1 into $x_2(n_2)$ as follows:

```
Procedure Vremove(B,v,n0,n1,n2)
Call Vreverse (B,v,n0,n1,n2-1); reverse x1 with {x2(1),...,x2(n2-1)}
N = n0+n1+n2                     ; index of x2(n2)
Do i = n0+n1, n0+1, -1           ; remove {x1(n1),...,x1(1)} into x2(n2)
i' = i - n0                      ; element number within x1
  If (B(i,N) = 0) Skip to next i
  Do j = 1, n0                   ; update arcs from x0 to x2(n2)
    If (B(j,i) = 0) Skip to next j
    B(j,N) = B(j,N) + B(j,i) * B(i,N)
  End Do
  Do j = n0+1, i-1               ; update arcs from {x1(1),...,x1(i'-1)}
                                   to x2(n2)
  If (B(j,i) = 0) Skip to next j
    B(j,N) = B(j,N) + B(j,i) * B(i,N)
  End Do
  Do j = n0+n1+1, N-1            ; update arcs from {x2(1),...,x2(n2-1)}
                                   to x2(n2)
    If (B(j,i) = 0) Skip to next j
    B(j,N) = B(j,N) + B(j,i) * B(i,N)
  End Do
  If (v(i) = 0) Skip to next i
```

```
  If ((v(i) ≠ ∞) and (v(N) ≠ ∞))
    v(N) = v(N) + B(i,N) * B(i,N) * v(i) ; update conditional
                                           variance of x2(n2)
  Else
    v(N) = ∞
  End If
End Do
Do i = 1, N                    ; zero all arcs to and from x1
  Do j = n0+1, n0+n2
    B(i,j) = 0
    B(j,i) = 0
  End Do
End Do
```

APPENDIX 2.B

DETAILED OPERATION COUNTS FOR TIME UPDATE ALGORITHMS

This appendix presents detailed operation counts for time update algorithms. Tables 2.B-1 and 2.B-2 contain results for the time update algorithm in Section 2.6. Tables 2.B-3 and 2.B-4 contain results for the colored noise time update in Section 2.7.

Table 2.B-1 DETAILED OPERATION COUNTS FOR REMOVAL OF x(k) INTO x(k+1) WITH WHITE NOISE

	Additions	Multiplications	Divisions
Do i = 1, n			
$v(i) = B_{12}(i,1) * \mu(1)$	0	n	0
Do j = 2, n			
$v(i) = v(i) + B_{12}(i,j) * \mu(j)$	n^2-n	n^2-n	0
End Do			
End Do			
Do i = n, 1, -1			
Do j = 1, n-1			
If $(j \neq 1)$			
Do k = 1, r			
$B_{02}(k,j) = B_{02}(k,j) + B_{01}(k,i) * B_{12}(i,j)$	$(n^2-2n)r$	$(n^2-2n)r$	0
End Do			
End If			
Do k = 1, i-1			
$B_{12}(k,j) = B_{12}(k,j) + B_{11}(k,i) * B_{12}(i,j)$	$0.5n^3-n^2+0.5n$	$0.5n^3-n^2+0.5n$	0
End Do			
If $(j <= n+1-i)$			
Do k = 1, j-1			
$B_{22}(k,j) = B_{22}(k,j) + B_{21}(k,i) * B_{12}(i,j)$	$0.17n^3-1.17n^2+1$	$0.17n^3-1.17n^2+1$	0
End Do			
Else			
Do k = 1, n+1-i			
$B_{22}(k,j) = B_{22}(k,j) + B_{21}(k,i) * B_{12}(i,j)$	$0.17n^3-0.5n^2+0.3n$	$0.17n^3-0.5n^2+0.3n$	0
End Do			
End If			
If $(j > n+1-i)$			
$B_{21}(j,i) = 0$			
Else			
If $(j = n+1-i)$			
$v(j) = B_{12}(i,j) * B_{12}(i,j) * v(i)$	0	$2n-2$	0
$v(i) = 0$			
$B_{21}(j,i) = 1 / B_{12}(i,j)$	0	0	$n-1$
Else			
vjold = v(j)			
$v(j) = vjold + B_{12}(i,j) * B_{12}(i,j) * v(i)$	$0.5n^2-1.5n+1$	n^2-3n+2	0
vratio = v(i) / v(j)	0	0	$0.5n^2-1.5n+1$
v(i) = vjold * vratio	0	$0.5n^2-1.5n+1$	0
$B_{21}(j,i) = B_{12}(i,j) * vratio$	0	$0.5n^2-1.5n+1$	0
End If			
Do k = 1, r			
$B_{01}(k,i) = B_{01}(k,i) - B_{02}(k,j) * B_{21}(j,i)$	$(0.5n^2+0.5n-1)r$	$(0.5n^2+0.5n-1)r$	0
End Do			
Do k = 1, i-1			
$B_{11}(k,i) = B_{11}(k,i) - B_{12}(k,j) * B_{21}(j,i)$	$0.17n^3-0.17n$	$0.17n^3-0.17n$	0
End Do			
Do k = 1, j-1			
$B_{21}(k,i) = B_{21}(k,i) - B_{22}(k,j) * B_{21}(j,i)$	$0.17n^3-1.17n+1$	$0.17n^3-1.17n+1$	0
End Do			
End If			
$B_{12}(i,j) = 0$			
End Do			
End Do			
Do i = n, 1, -1			
Do j = 1, r			
$B_{02}(j,n) = B_{02}(j,n) + B_{01}(j,i) * B_{12}(i,n)$	nr	nr	0
End Do			
Do j = 1, i-1			
$B_{12}(j,n) = B_{12}(j,n) + B_{11}(j,i) * B_{12}(i,n)$	$0.5n^2-0.5n$	$0.5n^2-0.5n$	0
End Do			
If (i = 1)			
Do j = 1, n-1			
$B_{22}(j,n) = B_{22}(j,n) + B_{21}(j,i) * B_{12}(i,n)$	$n-1$	$n-1$	0
End Do			
Else			
Do j = 1, n+1-i			
$B_{22}(j,n) = B_{22}(j,n) + B_{21}(j,i) * B_{12}(i,n)$	$0.5n^2-0.5n$	$0.5n^2-0.5n$	0
End Do			
End If			
If (i = 1)			
$v(n) = v(n) + B_{12}(i,n) * B_{12}(i,n) * v(i)$	1	2	0
End If			
End Do			
Do i = 1, n			
Do j = 1, n			
$B_{11}(i,j) = 0$			
$B_{12}(i,j) = 0$			
$B_{21}(i,j) = 0$			
End Do			
End Do			
TOTALS	$1.7n^3-0.17n^2-3n$ $+(1.5n^2-0.5n-1)r$	$1.17n^3+1.3n^2-4.5n+5$ $+(1.5n^2-0.5n-1)r$	$0.5n^2-0.5n$

Table 2.B-2 DETAILED OPERATION COUNTS FOR REMOVAL OF w(k) INTO x(k+1) WITH WHITE NOISE

	Additions	Multiplications	Divisions
Do i = r, 1, -1			
Do j = 1, n-1			
If(j ≠ 1)			
Do k = 1, i-1			
$B_{12}(k,j) = B_{12}(k,j) + B_{11}(k,i) * B_{12}(i,j)$	$(0.5n-1)r^2-(0.5n-1)r$	$(0.5n-1)r^2-(0.5n-1)r$	0
End Do			
End If			
Do k = 1, j-1			
$B_{22}(k,j) = B_{22}(k,j) + B_{21}(k,i) * B_{12}(i,j)$	$(0.5n^2-1.5n+1)r$	$(0.5n^2-1.5n+1)r$	0
End Do			
vjold = v(j)			
$v(j) = vjold + B_{12}(i,j) * B_{12}(i,j) * v(i)$	$(n-1)r$	$(2n-2)r$	0
vratio = v(i) / v(j)	0	0	$(n-1)r$
v(i) = vjold * vratio	0	$(n-1)r$	0
$B_{21}(j,i) = B_{12}(i,j) * vratio$	0	$(n-1)r$	0
Do k = 1, i-1			
$B_{11}(k,i) = B_{11}(k,i) - B_{12}(k,j) * B_{21}(j,i)$	$(0.5n-0.5)r^2-(0.5n-0.5)r$	$(0.5n-1)r^2-(0.5n-0.5)r$	0
End Do			
Do k = 1, j-1			
$B_{21}(k,i) = B_{21}(k,i) - B_{22}(k,j) * B_{21}(j,i)$	$(0.5n^2-1.5n+1)r$	$(0.5n^2-1.5n+1)r$	0
End Do			
End Do			
End Do			
Do i = r, 1, -1			
Do j = 1, i-1			
$B_{12}(j,n) = B_{12}(j,n) + B_{11}(j,i) * B_{12}(i,n)$	$0.5r^2-0.5r$	$0.5r^2-0.5r$	0
End Do			
Do j = 1, n-1			
$B_{22}(j,n) = B_{22}(j,n) + B_{21}(j,i) * B_{12}(i,n)$	$(n-1)r$	$(n-1)r$	0
End Do			
$v(n) = v(n) + B_{12}(i,n) * B_{12}(i,n) * v(i)$	r	2r	0
End Do			
Do i = 1, r			
Do j = 1, r			
$B_{11}(i,j) = 0$			
End Do			
Do j = 1, n			
$B_{12}(i,j) = 0$			
$B_{21}(j,i) = 0$			
End Do			
End Do			
TOTALS	$(n^2-2n+2)r+(n-1)r^2$	$(n^2+n)r+(n-1)r^2$	$(n-1)r$

Table 2.B-3 DETAILED OPERATION COUNTS FOR REMOVAL OF x(k) INTO x(k+1) WITH COLORED NOISE

	Additions	Multiplications	Divisions
Do i = 1, n			
v(i) = B_{12}(i,1) * μ(1)	0	n	0
Do j = 2, n			
v(i) = v(i) + B_{12}(i,j) * μ(j)	n^2-n	n^2-n	0
End Do			
End Do			
Do i = n, 1, -1			
Do j = 1, n-1			
Do k = 1, r			
B_{02}(k,j) = B_{02}(k,j) + B_{01}(k,i) * B_{12}(i,j)	$(n^2-n)r$	$(n^2-n)r$	0
End Do			
Do k = 1, i-1			
B_{12}(k,j) = B_{12}(k,j) + B_{11}(k,i) * B_{12}(i,j)	$0.5n^3-n^2+0.5n$	$0.5n^3-n^2+0.5n$	0
End Do			
If (j < = n+1-i)			
Do k = 1, j-1			
B_{22}(k,j) = B_{22}(k,j) + B_{21}(k,i) * B_{12}(i,j)	$0.17n^3-1.17n^2+1$	$0.17n^3-1.17n^2+1$	0
End Do			
Else			
Do k = 1, n+1-i			
B_{22}(k,j) = B_{22}(k,j) + B_{21}(k,i) * B_{12}(i,j)	$0.17n^3-0.5n^2+0.3n$	$0.17n^3-0.5n^2+0.3n$	0
End Do			
End if			
If (j> n+1-i)			
B_{21}(j,i) = 0			
Else			
If (j = n+1-i)			
v(j) = B_{12}(i,j) * B_{12}(i,j) * v(i)	0	$2n-2$	0
v(i) = 0			
B_{21}(j,i) = 1 / B_{12}(i,j)	0	0	$n-1$
Else			
vjold = V(j)			
v(j) = vjold + B_{12}(i,j) * B_{12}(i,j) * v(i)	$0.5n^2-1.5n+1$	n^2-3n+2	0
vratio = v(i) / v(j)	0	0	$0.5n^2-1.5n+1$
v(i) = vjold * vratio	0	$0.5n^2-1.5n+1$	0
B_{21}(j,i) = B_{12}(i,j) * vratio	0	$0.5n^2-1.5n+1$	0
End If			
Do k = 1, r			
B_{01}(k,i) = B_{01}(k,i) - B_{02}(k,j) * B_{21}(j,i)	$(0.5n^2+0.5n-1)r$	$(0.5n^2+0.5n-1)r$	0
End Do			
Do k = 1, i-1			
B_{11}(k,i) = B_{11}(k,i) - B_{12}(k,j) * B_{21}(j,i)	$0.17n^3-0.17n$	$0.17n^3-0.17n$	0
End Do			
Do k = 1, j-1			
B_{21}(k,i) = B_{21}(k,i) - B_{22}(k,j) * B_{21}(j,i)	$0.17n^3-1.17n+1$	$0.17n^3-1.17n+1$	0
End Do			
End If			
B_{12}(i,j) = 0			
End Do			
End Do			
Do i = n, 1, -1			
Do j = 1, r			
B_{02}(j,n) = B_{02}(j,n) + B_{01}(j,i) * B_{12}(i,n)	nr	nr	0
End Do			
Do j = 1, i-1			
B_{12}(j,n) = B_{12}(j,n) + B_{11}(j,i) * B_{12}(i,n)	$0.5n^2-0.5n$	$0.5n^2-0.5n$	0
End Do			
If (i = 1)			
Do j = 1, n-1			
B_{22}(j,n) = B_{22}(j,n) + B_{21}(j,i) * B_{12}(i,n)	$n-1$	$n-1$	0
End Do			
Else			
Do j = 1, n+1-i			
B_{22}(j,n) = B_{22}(j,n) + B_{21}(j,i) * B_{12}(i,n)	$0.5n^2-0.5n$	$0.5n^2-0.5n$	0
End Do			
End If			
If (i = 1)			
v(n) = v(n) + B_{12}(i,n) * B_{12}(i,n) * v(i)	1	2	0
End If			
End Do			
Do i = 1, n			
Do j = 1, n			
B_{11}(i,j) = 0			
B_{12}(i,j) = 0			
B_{21}(i,j) = 0			
End Do			
End Do			
TOTALS	$1.7n^3-0.17n^2-3n+3$ $+(1.5n^2+0.5n-1)r$	$1.17n^3+1.3n^2-4.5n+5$ $+(1.5n^2+0.5n-1)r$	$0.5n^2-0.5n$

Table 2.B-4 DETAILED OPERATION COUNTS FOR REMOVAL OF y(k) INTO t(k+1) WITH COLORED NOISE

	Additions	Multiplications	Divisions
Do i = r, 1, -1			
Do j = i, r+n-1			
Do k = 1, i-1			
$B_{12}(k,j) = B_{12}(k,j) + B_{11}(k,i) * B_{12}(i,j)$	$(-0.5n+0.3)r+(0.5n-0.5)r^2+0.17r^3$	$(-0.5n+0.3)r+(0.5n-0.5)r^2+0.17r^3$	0
End Do			
Do k = 1, j-1			
$B_{22}(k,j) = B_{22}(k,j) + B_{21}(k,i) * B_{12}(i,j)$	$(0.5n^2-0.5n+0.3)r+(n-1)r^2+0.3r^3$	$(0.5n^2-0.5n+0.3)r+(n-1)r^2+0.3r^3$	0
End Do			
vjold = v(j)			
$v(j) = vjold + B_{12}(i,j)$	$(n-0.5)r+0.5r^2$	$(2n-1)r+r^2$	0
$vratio = v(i) / v(j) * B_{12}(i,j) * v(i)$	0	0	$(n-0.5)r+0.5r^2$
v(i) = vjold * vratio	0	$(n-0.5)r+0.5r^2$	0
$B_{21}(j,i) = B_{12}(i,j) * vratio$	0	$(n-0.5)r+0.5r^2$	0
Do k = 1, i-1			
$B_{11}(k,i) = B_{11}(k,i) - B_{12}(k,j) * B_{21}(j,i)$	$(-0.5n+0.3)r+(0.5n-0.5)r^2+0.17r^3$	$(-0.5n+0.3)r+(0.5n-0.5)r^2+0.17r^3$	0
End Do			
Do k = 1, j-1			
$B_{21}(k,i) = B_{21}(k,i) - B_{22}(k,j) * B_{21}(j,i)$	$(0.5n^2-0.5n+0.3)r+(n-1)r^2+0.3r^3$	$(0.5n^2-0.5n+0.3)r+(n-1)r^2+0.3r^3$	0
End Do			
End Do			
End Do			
Do i = r, 1, -1			
Do j = 1, i-1			
$B_{12}(j,r+n) = B_{12}(j,r+n) + B_{11}(j,i) * B_{12}(i,r+n)$	$0.5r^2-0.5r$	$0.5r^2-0.5r$	0
End Do			
Do j = 1, n-1			
$B_{22}(j,r+n) = B_{22}(j,r+n) + B_{21}(j,i) * B_{12}(i,r+n)$	$(n-1)r+r^2$	$(n-1)r+r^2$	0
End Do			
$v(r+n) = v(r+n) + B_{12}(i,r+n) * B_{12}(i,r+n) * v(i)$	r	2r	0
End Do			
Do i = 1,r			
Do j = 1,r			
$B_{11}(i,j) = 0$			
End Do			
Do j = 1,n			
$B_{12}(i,j) = 0$			
$B_{21}(j,i) = 0$			
End Do			
End Do			
TOTALS	$(n^2-1)r+(3n-1)r^2+r^3$	$(n^2+3n-1.5)r+(3n-0.5)r^2+r^3$	$(n-0.5)r+0.5r^2$

CHAPTER 3

THE LINEAR-QUADRATIC INFLUENCE DIAGRAM

3.1 Introduction

For the normal influence diagram, the conditional mean of a state vector component is a linear function of its conditioning variables, and the conditional variance is independent of its conditioning variables. These assumptions are sufficient to guarantee that the joint distribution of the state vector is multivariate normal [8]. In this chapter, an influence diagram is developed to analyze decision problems with quadratic value functions and linear deterministic nodes (as in Chapter 1), but we relax the requirement of normally distributed chance nodes. The influence diagram factorization theorem of Appendix 1.A is shown to apply to all covariance matrices, not just to those corresponding to a multivariate normal vector. As a consequence, influence diagram operations detailed in Chapter 1 are valid for linear-quadratic decision models. Probabilistic operations of removal and reversal are interpreted as transformations to the covariance matrix factorization. Chance node removal into the value node also is interpreted, and the implication for decision node removal is discussed.

3.2 Influence Diagram Factorization of Covariance Matrices

To determine the expected value of a quadratic value function, it is sufficient to know μ and Σ, the mean vector and covariance matrix of the state vector. Let the state vector X be indexed by $N = \{1,...,n\}$. The state vector covariance matrix Σ can be factored into a matrix product that is equivalent to the normal influence diagram. For this representation, each scalar node $j \in N$ has an unconditional mean μ_j and a scalar v_j, which we shall call the conditional variance, although it is guaranteed to be the conditional variance only if the distribution of X_N is multivariate normal. For each pair of nodes $(i,j) \subset N \times N$, such that $i < j$, there is a scalar arc coefficient b_{ij}. Let

$$U_j = \begin{bmatrix} I_{j-1} & B_{\cdot j} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{n-j} \end{bmatrix},$$

where

$$B_{\cdot j} = (b_{1j}, \ldots, b_{j-1,j})^T.$$

The covariance matrix is factored as follows:

$$\Sigma = U_n^T \ldots U_1^T \operatorname{diag}(v_1, \ldots, v_n)\, U_1 \ldots U_n.$$

Appendix 3.A contains a detailed proof of this factorization.

3.3 Probabilistic Analysis

The probabilistic operations of removal and reversal presented in Chapter 1 are valid for maintaining the influence diagram factorization of the covariance matrix of the resulting state vector and node ordering. For the covariance matrix, removal of node $i \in N$ is simply elimination of the row i and column i of the matrix. The operation is

$$N \leftarrow N \backslash \{i\}$$

$$\Sigma \leftarrow \Sigma_{NN}.$$

Reversal of nodes i and j is a symmetric permutation of the covariance matrix. Assume the nodes are ordered so that $j = i+1$. Reversal of nodes i and j is as follows:

$$N \leftarrow \{1,\ldots,i-1,j,i,j+1,\ldots,n\}$$

$$\Sigma \leftarrow \Sigma_{NN}.$$

In Appendix 3.B, we prove the validity of using normal influence diagram operations for updating influence diagram factorizations of the updated covariance matrix following removal or reversal.

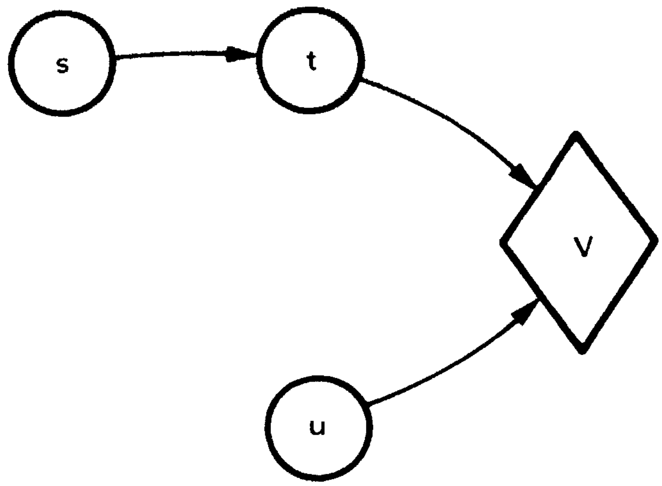
3.4 Decision Analysis

In the normal influence diagram, the decision analysis operation of removal of a vector of chance nodes into the value node maintains the conditional expected value of the value function. If the state vector is not normal, the conditional expected value of the value function is maintained if there are no predecessors to the vector of chance nodes being removed. Referring to Fig. 3-1, let t be the set of chance nodes to be removed; s, the set of direct and indirect predecessors of t; and u, all other nodes. Let the conditional expected value of the value function be:

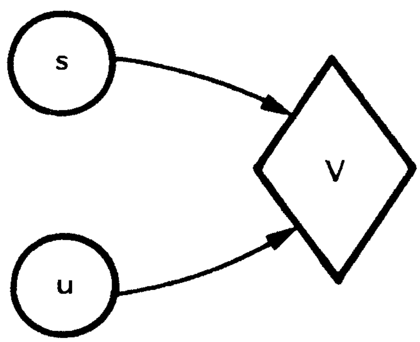
$$E[V(X_s,X_t,X_u)|X_s = x_s,\ X_t = x_t,\ X_u = x_u] = \frac{1}{2}x_s^TQ_{ss}x_s + \frac{1}{2}x_s^TQ_{st}x_t$$

$$+ \frac{1}{2}x_s^TQ_{su}x_u + \frac{1}{2}x_t^TQ_{ts}x_s + \frac{1}{2}x_t^TQ_{tt}x_t + \frac{1}{2}x_t^TQ_{tu}x_u$$

$$+ \frac{1}{2}x_u^TQ_{us}x_s + \frac{1}{2}x_u^TQ_{ut}x_t + \frac{1}{2}x_u^TQ_{uu}x_u + P_s^Tx_s + p_t^Tx_t + p_u^Tx_u + r.$$



INITIAL DIAGRAM



NODE t REMOVED INTO VALUE NODE

Fig. 3-1 Chance Node Removal Into Value Node

The conditional expected value of the value function after removing t into the value node is:

$$E[V(X_s,X_t,X_u)|X_s = X_s,X_u = x_u] = \frac{1}{2} x_s^TQ_{ss}x_s + \frac{1}{2} x_s^TQ_{st}E[X_t|X_s = x_s]$$

$$+ \frac{1}{2} x_s^TQ_{su}x_u + \frac{1}{2} E[X_t|X_s = x_s]^TQ_{ts}x_s$$

$$+ \frac{1}{2} \text{trace}(Q_{tt}\text{Var}[X_t|X_s = x_s]) + \frac{1}{2} E[X_t|X_s = x_s]^TQ_{tt}E[X_t|X_s=x_s]$$

$$+ \frac{1}{2} E[X_t|X_s = x_s]^TQ_{ru}x_u + \frac{1}{2} x_u^TQ_{us}x_s + \frac{1}{2} x_u^TQ_{ut}E[X_t|X_s = x_s]$$

$$+ \frac{1}{2} x_u^TQ_{uu}x_u + p_s^Tx_s + p_t^TE[X_t|X_s = x_s] + p_u^Tx_u + r .$$

If the state vector were normal, we would complete the removal by substituting the linear conditional expectation and constant conditional covariance matrix into the above expression. For non-normal variables, the functional forms for the required conditional expectation and covariance are not simple linear functions or constants. If $q = s \cup t$ is a set of chance nodes and we remove q into the value node in one operation, the conditional expected value of the value function is:

$$E[V(X_q,X_u)|X_u = x_u] = \frac{1}{2} \text{trace}(Q_{qq}\text{Var}[X_q]) + \frac{1}{2} E[X_q]^TQ_{qq}E[X_q]$$

$$+ \frac{1}{2} E[X_q]^TQ_{qu}x_u + \frac{1}{2} x_u^TQ_{uq}E[X_q] + \frac{1}{2} x_u^TQ_{uu}x_u + p_q^TE[X_q] + p_u^Tx_u + r$$

$$= \frac{1}{2} x_u^TQ_{uu}x_u + \frac{1}{2} (p_u + Q_{qu}E[X_q])^Tx_u + r + \frac{1}{2} \text{trace}(Q_{qq}\text{Var}[X_q])$$

$$+ \frac{1}{2} E[X_q]^TQ_{qq}E[X_q] + p_q^TE[X_q] .$$

In this instance, only unconditional expectations and covariances are necessary to remove a vector of chance nodes, since there is no arc from node u to nodes s and t in Fig. 3-1. This guarantees that the result represents the conditional expected value of the value function. In Appendix 3.C, we show that removal of q can be completed by a series of scalar normal influence diagram removal operations, with the value node resulting representing the conditional expected value of the value function.

If the value node represents the conditional expected value of a quadratic value function, a decision node can be removed by normal influence diagram operations. This is because decision node removal is a deterministic maximization operation, and does not depend on the probabilistic structure of the chance nodes.

3.5 Time Series Models

In this section, we use linear-quadratic influence diagrams to represent stochastic time series models used by Box and Jenkins [2].

Figure 3-2 demonstrates the linear filter model, where

$$z_t = \mu + a_t + \Psi_1 a_{t-1} + \Psi_2 a_{t-2} + \cdots ;$$

$$E(a_{t-j}) = 0, \quad \text{for } j = 0,1,\ldots ;$$

$$\mathrm{Var}(a_{t-j}) = \sigma_a^2, \quad \text{for } j = 0,1,\ldots .$$

If $\lim_{t\to\infty} \{z_t | z_0\} = \mu$, almost surely for all z_0, then the process is stationary.

Figure 3-3 shows $AR(p)$, the autoregressive process of order p.

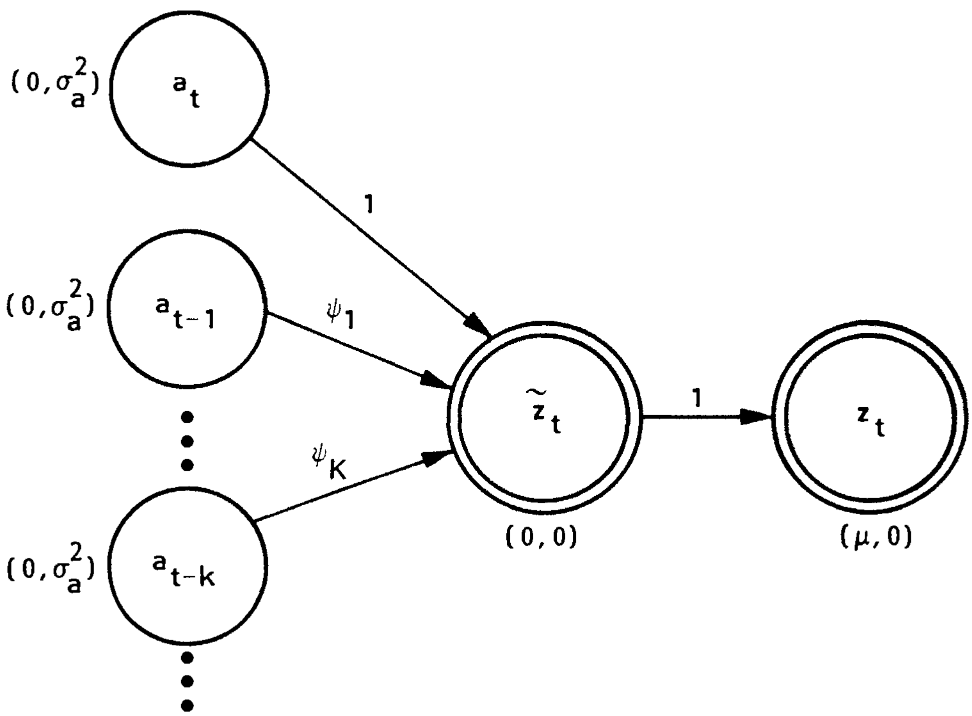


Fig. 3-2 Linear Filter Model

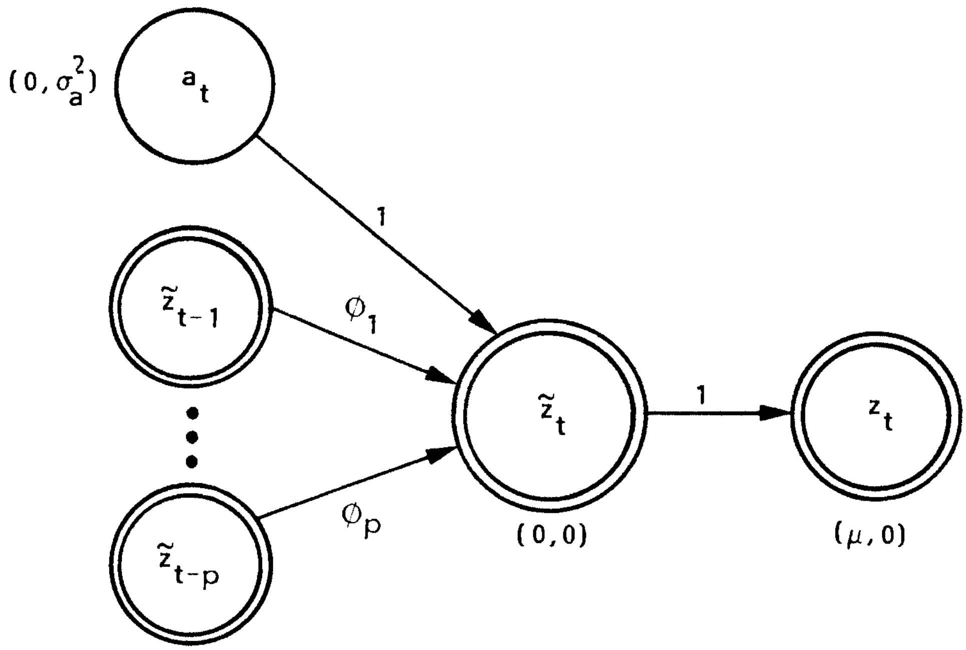


Fig. 3-3 Model for the Autoregressive (AR) Process of Order p

The AR(p) model is:

$$z_t = \mu + \tilde{z}_t$$
$$= \mu + \phi_1 \tilde{z}_{t-1} + \dots + \phi_p \tilde{z}_{t-p} + a_t \ ;$$
$$E(a_t) = 0 \ ;$$
$$Var(a_t) = \sigma_a^2 \ .$$

MA(q), the moving average process of order q, is shown in Fig. 3-4. The MA(q) model is:

$$z_t = \mu + \tilde{z}_t$$
$$= \mu + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \ ;$$
$$E(a_{t-j}) = 0, \quad \text{for } j = 0,\dots,q \ ;$$
$$Var(a_{t-j}) = \sigma_a^2, \quad \text{for } j = 0,\dots,q \ .$$

ARMA(p,q), the mixed autoregressive-moving average process of order (p,q), is a combination of the AR(p) and MA(q) processes, and is shown in Fig. 3-5.

AR(p), MA(q), and ARMA(p,q) processes are usually assumed to be stationary. If the process $z_1,\dots,z_t$ is not stationary, it is possible that the d^{th} difference of the process is stationary. For example, the linear process

$$z_t = a_t + z_{t-1}$$

is not stationary, but the first difference process

$$\nabla z_t = z_t - z_{t-1} = a_t$$

is stationary. Define the d^{th} difference as

$$\nabla^d z_t = \sum_{j=0}^{d} \binom{d}{j} (-1)^j z_j \ .$$

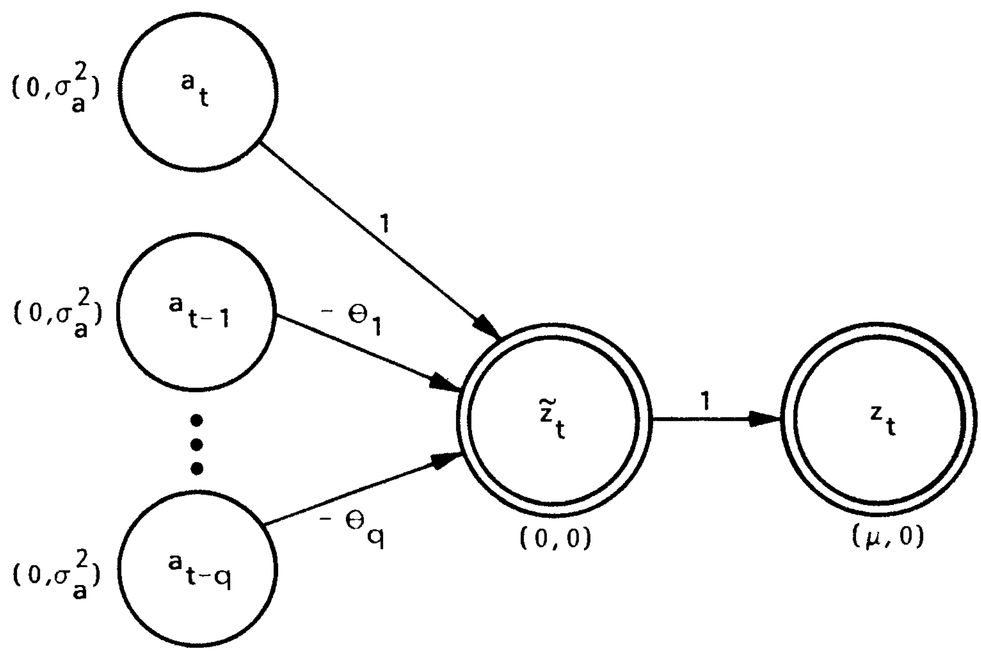


Fig. 3-4 Model for the Moving Average (MA) Process of Order q

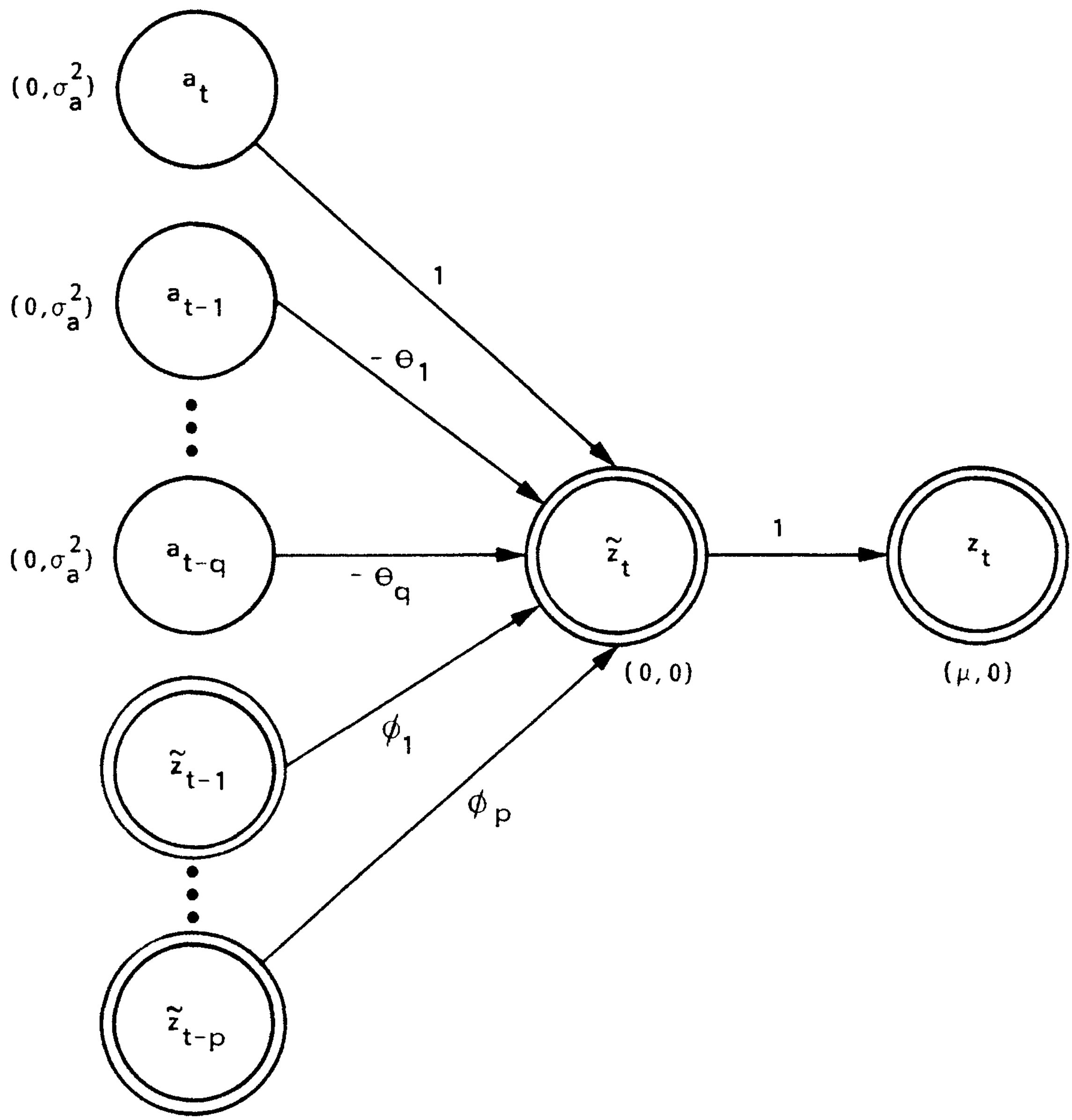


Fig. 3-5 Model for the Mixed Autoregressive – Moving Average (ARMA) Process of Order (p,q)

The stationary ARMA(p,q) model for $\nabla^d z_t$ is the ARIMA(p,d,q), or autoregressive-integrated moving average model of order (p,d,q), shown in Fig. 3-6.

The inverse of the d^{th} difference operator is the d^{th} summation operator

$$S^d z_t = \nabla^{-d} z_t = \sum_{j_d=0}^{\infty} \cdots \sum_{j_1=0}^{j_2} z_{t-j_1} \; .$$

Using the d^{th} summation operator, the nonstationary time series $\{z_1,\ldots,z_t,\ldots\}$ implied by the ARIMA(p,d,q) model is shown in Fig. 3-7.

3.6 Conclusion

With certain restrictions, the linear-quadratic-Gaussian influence diagram operations of Chapter 1 are valid for decision problems with quadratic value functions and non-Gaussian random variables. Sequential decisions are allowed as long as no decision vector has a chance node successor. Referring to Fig. 3-8, nodes in the informational set $I(d_{j-1})$ may not have a successor in $s_j \cup s_{j+1} \ldots \cup s_{T+1}$, for $j = 1,\ldots,T$. Any chance or decision node can be a predecessor to the value node, which represents a quadratic value function. In summary, decisions cannot affect aleatory variables, and observation of an aleatory variable does not permit inference about unobserved aleatory variables.

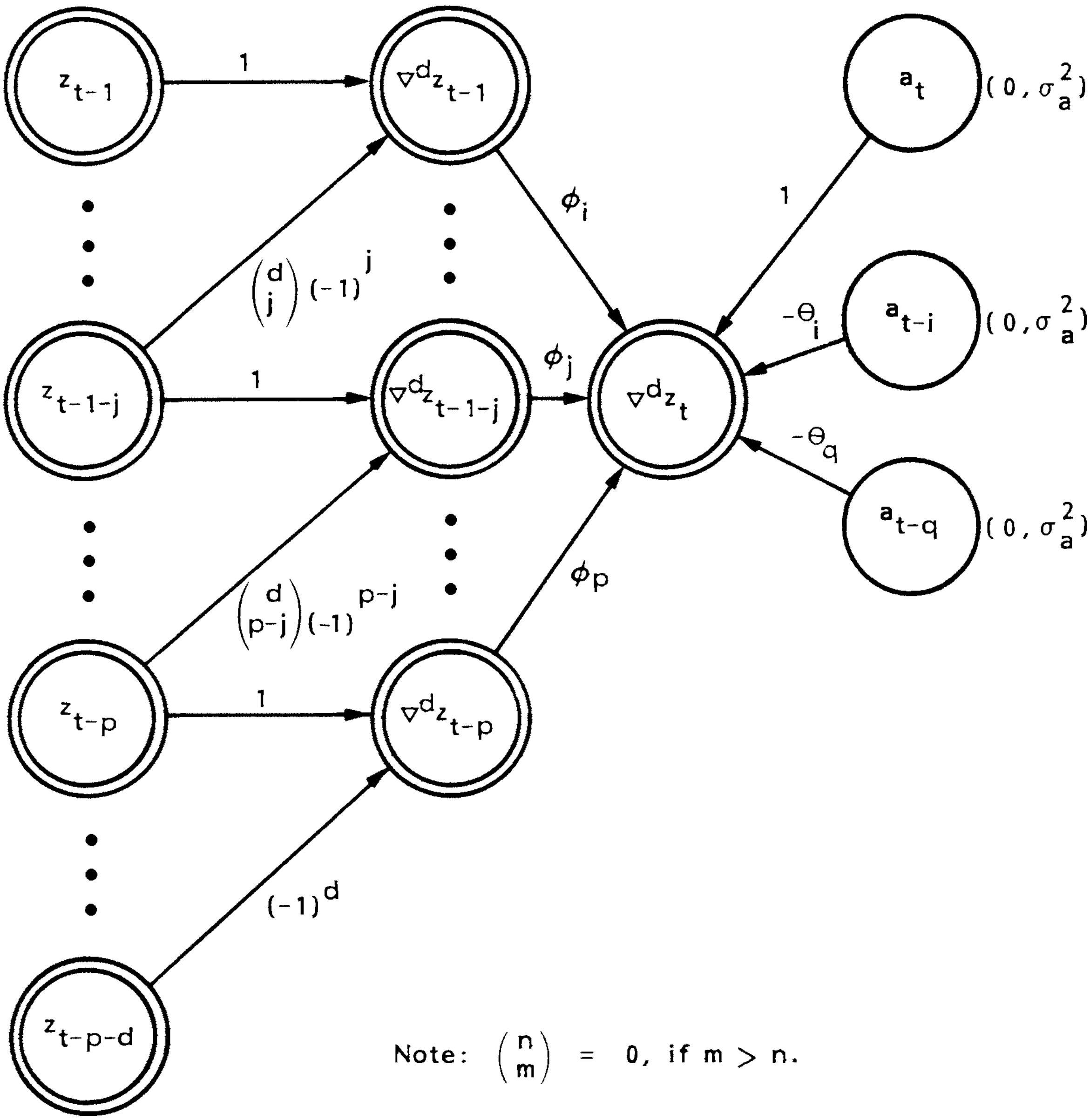


Fig. 3-6 Model for the Autoregressive – Integrated Moving Average (ARIMA) Process of Order (p,d,q)

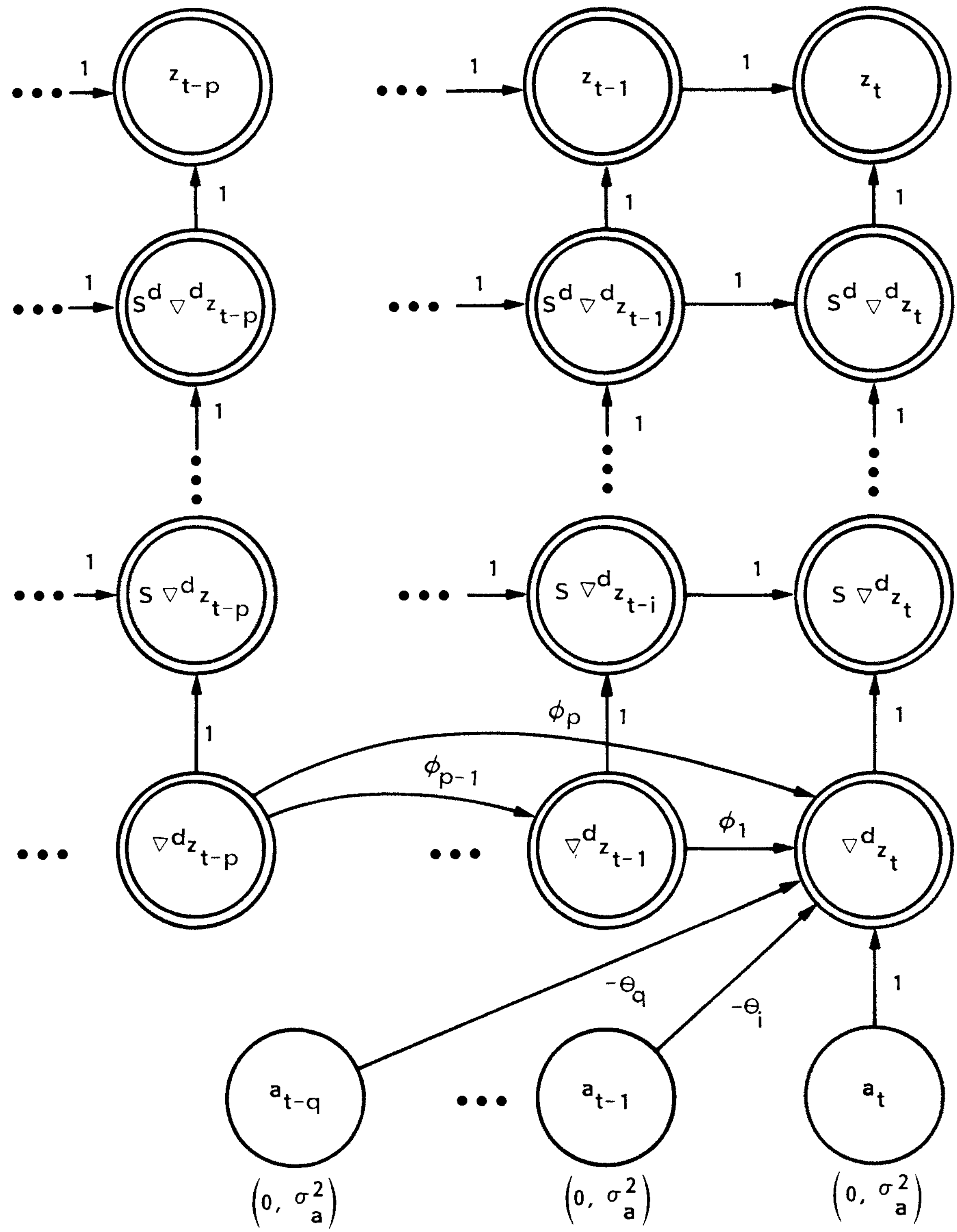


Fig. 3-7 ARIMA (p,d,q) as Nonstationary Summation Filter

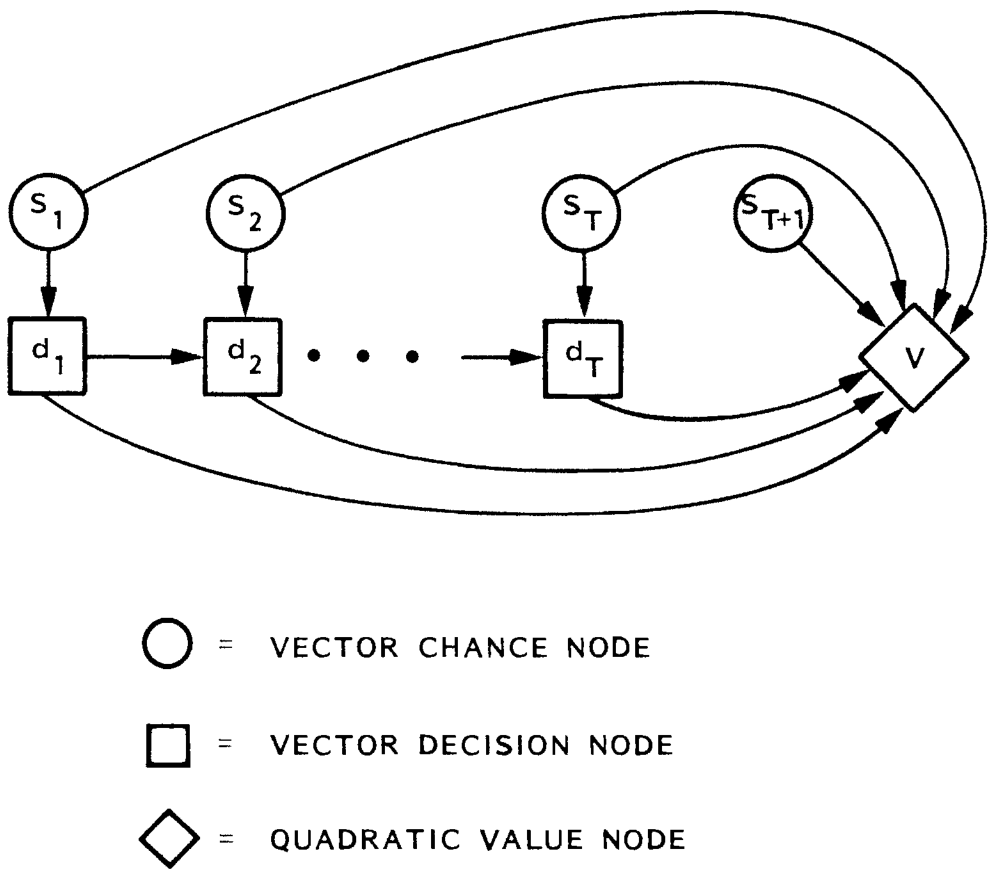


Fig. 3-8 Generalization of Results With Non-Gaussian Variables

APPENDIX 3.A

INFLUENCE DIAGRAM FACTORIZATION OF A COVARIANCE MATRIX

We can factor the covariance matrix of a multivariate normal vector into a $U^T DU$ form that can be read directly from the normal influence diagram representation of the state vector. The normality assumption is used in a key lemma for the proof of Appendix 1.A. This appendix presents a proof that any covariance matrix can be factored in the same way, without assuming normality.

<u>Definition</u>

Let $B_{\cdot j} = [b_{1j},\ldots,b_{j-1,j}]^T$, and

$$U_j = \begin{bmatrix} I_{j-1} & B_{\cdot j} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{n-j} \end{bmatrix},$$

for $j = 1,\ldots,n$. Let $v(n) = (v_1,\ldots,v_n)^T \geq 0$. The <u>n-dimensional influence diagram factorization</u> is the matrix product

$$U_n^T \ldots U_1^T \operatorname{diag}(v(n))\, U_1 \ldots U_n .$$

<u>Lemma 1</u>

If A has an (n-1)-dimensional influence diagram factorization, then the generalized inverse of A is

$$A^- = U_{n-1}^{-1} \ldots U_1^{-1} \operatorname{diag}(v_1^-,\ldots,v_{n-1}^-)(U_1^T)^{-1} \ldots (U_{n-1}^T)^{-1},$$

where

$$v_j^- = \begin{cases} 1/v_j & \text{for } v_j \neq 0 \\ 0 & \text{for } v_j = 0, \end{cases}$$

for $j = 1,\ldots,n-1$.

Proof

The four necessary and sufficient properties for generalized inverses can be verified by matrix multiplication. They are [4, p.24]:

(1) $AA^-A = A$,

(2) $A^-AA^- = A^-$,

(3) AA^- is symmetric, and

(4) A^-A is symmetric. $\square$

Lemma 2

Let

$$\Sigma = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix}$$

be a positive semi-definite (PSD) matrix, where A is an $(n-1)\times(n-1)$ matrix, b is an $(n-1)$ vector, and c is a scalar. Then

(1) the system of equations $Ax = b$ is consistent,

(2) $x = A^-b$ is a solution of $Ax = b$, and

(3) $c - b^TA^-b \geq 0$.

Proof

(1) For consistency, it must be shown that $z^TA = 0 \Rightarrow z^Tb = 0$. Assume $z^TA = 0$. Σ is PSD; hence, for all scalar y,

$$0 \le [z^T \; y]\Sigma\begin{bmatrix} z \\ y \end{bmatrix} = z^T A z + 2yz^T b + cy^2 = 2yz^T b + cy^2.$$

Let

$$y = \begin{cases} -z^T b/2 & \text{for } c = 0 \\ -z^T b/c & \text{for } c \ne 0. \end{cases}$$

Then

$$0 \le 2yz^T b + cy^2 = -(z^T b)^2 \Rightarrow z^T b = 0.$$

(2) Since (1) is true, there exists x such that $Ax = b$. Let $y = A^- b$. Then

$$Ay = AA^- b = AA^- Ax = Ax = b ;$$

therefore, $y = A^- b$ is also a solution to $Ax = b$.

$$(3) \quad c - b^T A^- b = [-b^T A^- \; 1]\begin{bmatrix} A & b \\ b^T & c \end{bmatrix}\begin{bmatrix} -A^- b \\ 1 \end{bmatrix} \ge 0 ,$$

since Σ is PSD. □

Theorem

Let Σ be a symmetric $n \times n$ PSD matrix. Then an n-dimensional influence diagram factorization for Σ exists.

Proof (by induction on n)

For $n = 1$, let $U_1 = 1$ and $v_1 = \sigma_{11}$.

Assume the induction hypothesis is true for all $(n-1)\times(n-1)$ PSD matrices. Since Σ is symmetric and PSD, there is an $(n-1)\times(n-1)$ symmetric PSD matrix A such that

$$\Sigma = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix}$$

where b is an (n-1)-vector, and c is a scalar. By the induction hypothesis, there exists

$$W_j = \begin{bmatrix} I_{j-1} & B_{.j} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{(n-1)-j} \end{bmatrix}$$

for $j = 1,\ldots,n-1$, and $v(n-1) = (v_1,\ldots,v_{n-1})^T \geq 0$, such that

$$A = W_{n-1}^T \ldots W_1^T \operatorname{diag}(v(n-1))\, W_1 \ldots W_{n-1} .$$

Let

$$U_j = \begin{bmatrix} W_j & 0 \\ 0 & 1 \end{bmatrix},$$

for $j = 1,\ldots,n-1$. Also, let v_n be a scalar. Then

$$U_{n-1}^T \ \ldots\ U_1^T \operatorname{diag}(v(n))\, U_1 \ldots U_{n-1} = \begin{bmatrix} A & 0 \\ 0 & v_n \end{bmatrix}.$$

We must solve for U_n and v_n such that

$$\begin{bmatrix} A & b \\ b^T & c \end{bmatrix} = U_n^T U_{n-1}^T \ldots U_1^T \operatorname{diag}(v(n))\, U_1 \ldots U_{n-1} U_n = U_n^T \begin{bmatrix} A & 0 \\ 0 & v_n \end{bmatrix} U_n$$

$$= \begin{bmatrix} I_{n-1} & 0 \\ B_{.n}^T & 1 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & v_n \end{bmatrix} \begin{bmatrix} I_{n-1} & B_{.n} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} A & AB_{.n} \\ B_{.n}^T A & v_n + B_{.n}^T AB_{.n} \end{bmatrix}.$$

$B_{\cdot n}$ is the solution of $AB_{\cdot n} = b$. By Lemma 2, this is consistent and $B_{\cdot n} = A^{-}b$, where A^{-} is given by Lemma 1. Solving for v_n,

$$
\begin{aligned}
v_n &= c - B_{\cdot n}^T A B_{\cdot n} \\
&= c - b^T (A^{-})^T A A^{-} b \\
&= c - b^T A^{-} A A^{-} b \qquad \text{(since } A^{-} \text{ is symmetric)} \\
&= c - b^T A^{-} b \geq 0, \quad \text{by Lemma 2.} \qquad \square
\end{aligned}
$$

APPENDIX 3.B

PROOFS FOR PROBABILISTIC ANALYSIS

This appendix shows that normal influence diagram removal and reversal operations maintain the influence diagram factorization of the updated covariance matrix.

The following notation is used for the proofs:

$s = \{1,\ldots,i-1\}$;

$j = i + 1$;

$t = \{1,\ldots,i,j\}$;

$u = \{1,\ldots,i-1,j\}$;

$w = \{1,\ldots,i-1,j,i\}$.

<u>Lemma</u>

$$\Sigma_{tt} = \begin{bmatrix} \Sigma_{ss} & \Sigma_{ss}B_{si} & \Sigma_{ss}(B_{sj}+B_{si}b_{ij}) \\ B_{si}^T\Sigma_{ss} & v_i+B_{si}^T\Sigma_{ss}B_{si} & \sigma_{ij} \\ (B_{sj}+B_{si}b_{ij})^T\Sigma_{ss} & \sigma_{ij} & \sigma_{jj} \end{bmatrix},$$

where

$$\sigma_{ij} = B_{si}^T\Sigma_{ss}B_{sj} + b_{ij}(v_i + B_{si}^T\Sigma_{ss}B_{si}),$$

and

$$\sigma_{jj} = (B_{sj} + B_{si}b_{ij})^T\Sigma_{ss}B_{sj} + b_{ij}B_{si}^T\Sigma_{ss}B_{sj} + b_{ij}^2(v_i + B_{si}^T\Sigma_{ss}B_{si}) + v_j.$$

Proof

Apply the influence diagram factorization theorem of Appendix 3.A and matrix multiplication. □

Theorem (Removal)

$$\Sigma_{uu} = \begin{bmatrix} I_{j-1} & 0 \\ (B_{sj}+B_{si}b_{ij})^T & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{ss} & 0 \\ 0 & v_j+b_{ij}^2 v_i \end{bmatrix} \begin{bmatrix} I_{j-1} & B_{sj}+B_{si}b_{ij} \\ 0 & 1 \end{bmatrix} .$$

Proof

Obtain Σ_{uu} by striking row i and column i from the matrix of the Lemma. Apply matrix multiplication to the factorization given for Σ_{uu} to verify the identity. □

Theorem (Reversal)

$$\Sigma_{ww} = \begin{bmatrix} I_{i-1} & 0 & 0 \\ (B_{sj}+B_{si}b_{ij})^T & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_{i-1} & 0 & 0 \\ 0 & 1 & 0 \\ B_{si}^T-b_{ji}(B_{sj}+B_{si}b_{ij})^T & b_{ji} & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} \Sigma_{ss} & 0 & 0 \\ 0 & v_j+b_{ij}^2 v_i & 0 \\ 0 & 0 & v_i' \end{bmatrix} \begin{bmatrix} I_{i-1} & B_{si}-b_{ji}(B_{sj}+B_{si}b_{ij}) & 0 \\ 0 & b_{ji} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} I_{j-1} & B_{sj}+B_{si}b_{ij} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ,$$

where

$$b_{ji} = b_{ij}v_i/(v_j + b_{ij}^2 v_i),$$

and

$$v_i' = v_j v_i/(v_j + b_{ij}^2 v_i).$$

Proof

Obtain Σ_{ww} permuting the (i,j) rows and columns of the matrix of the Lemma. Apply matrix multiplication to the factorization given for Σ_{ww} to verify the identity. □

APPENDIX 3.C

PROOFS FOR DECISION ANALYSIS

This appendix shows that removal of a scalar chance node into a quadratic value node, using normal influence diagram operations, preserves the expectation of the value function.

The following notation is used:

$$N = \{1,\ldots,n\} \;;$$

$$s = \{1,\ldots,n-1\} \;;$$

$$X_n = (X_1,\ldots,X_n)^T \;;$$

$$\mu = \mu_N = E[X_N] \;;$$

$$\Sigma = \Sigma_{NN} = E[X_N X_N^T] - \mu_N \mu_N^T \,.$$

<u>Lemma 1</u>

$$\frac{1}{2}\,\text{trace}\{Q\Sigma\} = \frac{1}{2}\,\text{trace}\{Q_{new}\Sigma_{ss}\} + \frac{1}{2}\,q_{nn}v_n,$$

where

$$Q_{new} = Q_{ss} + B_{sn}Q_{ns} + Q_{sn}B_{sn}^T + B_{sn}^T q_{nn} B_{sn}.$$

<u>Proof</u>

$$\frac{1}{2}\,\text{trace}\{Q\Sigma\} = \frac{1}{2}\,\text{trace}\left\{\begin{bmatrix} Q_{ss} & Q_{sn} \\ Q_{ns} & q_{nn} \end{bmatrix}\begin{bmatrix} I_{n-1} & 0 \\ B_{sn}^T & 1 \end{bmatrix}\begin{bmatrix} \Sigma_{ss} & 0 \\ 0 & v_n \end{bmatrix}\begin{bmatrix} I_{n-1} & B_{sn} \\ 0 & 1 \end{bmatrix}\right\}$$

$$= \frac{1}{2}\,\text{trace}\left\{\begin{bmatrix} I_{n-1} & B_{sn} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} Q_{ss} & Q_{sn} \\ Q_{ns} & q_{nn} \end{bmatrix}\begin{bmatrix} I_{n-1} & 0 \\ B_{sn}^T & 1 \end{bmatrix}\begin{bmatrix} \Sigma_{ss} & 0 \\ 0 & v_n \end{bmatrix}\right\}$$

$$= \frac{1}{2}\ \text{trace}\left\{\begin{bmatrix} Q_{ss}+B_{sn}Q_{ns} & Q_{sn}+B_{sn}q_{nn} \\ Q_{ns} & q_{nn} \end{bmatrix}\begin{bmatrix} \Sigma_{ss} & 0 \\ B^T_{sn}\Sigma_{ss} & v_n \end{bmatrix}\right\}$$

$$= \frac{1}{2}\ \text{trace}\left\{\begin{bmatrix} (Q_{ss}+B_{sn}Q_{ns})\Sigma_{ss}+(Q_{sn}+B_{sn}q_{nn})B^T_{sn}\Sigma_{ss} & (Q_{sn}+B_{sn}q_{nn})v_n \\ Q_{ns}\Sigma_{ss}+q_{nn}B^T_{sn}\Sigma_{ss} & q_{nn}v_n \end{bmatrix}\right\}$$

$$= \frac{1}{2}\ \text{trace}\{(Q_{ss}+B_{sn}Q_{ns})\Sigma_{ss}+(Q_{sn}+B_{sn}q_{nn})B^T_{sn}\Sigma_{ss}\} + \frac{1}{2}\ q_{nn}v_n$$

$$= \frac{1}{2}\ \text{trace}\{(Q_{ss}+B_{sn}Q_{ns}+Q_{sn}B^T_{sn}+B_{sn}q_{nn}B^T_{sn})\Sigma_{ss}\} + \frac{1}{2}\ q_{nn}v_n\ . \qquad \square$$

Lemma 2

$$p^T\mu = p^T_{new}\ \mu_s - (\mu_n - B^T_{sn}\mu_s)(Q_{sn} + B_{sn}q_{nn})^T\mu_s + p_n(\mu_n - B^T_{sn}\mu_s),$$

where

$$p_{new} = p_s + (\mu_n - B^T_{sn}\mu_s)(Q_{sn} + B_{sn}q_{nn})^T\mu_s + B_{sn}p_n\ .$$

Proof

$$p^T_{new}\mu_s - (\mu_n-B^T_{sn}\mu_s)(Q_{sn}+B_{sn}q_{nn})^T\mu_s + p_n(\mu_n-B^T_{sn}\mu_s)$$

$$= [p_s + (\mu_n-B^T_{sn}\mu_s)(Q_{sn}+B_{sn}q_{nn})^T\mu_s + B_{sn}p_n]^T\mu_s$$

$$- (\mu_n-B^T_{sn}\mu_s)(Q_{sn}+B_{sn}q_{nn})^T\mu_s + p_n(\mu_n-B^T_{sn}\mu_s)$$

$$= p^T_s\mu_s + p_n\mu_n$$

$$= p^T\mu\ . \qquad \square$$

Lemma 3

$$\frac{1}{2}\mu^T Q \mu = \frac{1}{2}\mu_s^T Q_{new}\mu_s + (\mu_n - B_{sn}^T\mu_s)^2 q_{nn}$$

$$+ (\mu_n - B_{sn}^T\mu_s)(Q_{sn} + B_{sn}q_{nn})^T\mu_s ,$$

where

$$Q_{new} = Q_{ss} + B_{sn}Q_{ns} + Q_{sn}B_{sn}^T + B_{sn}^T q_{nn} B_{sn} .$$

Proof

$$\frac{1}{2}\mu_s^T Q_{new}\mu_s + (\mu_n - B_{sn}^T\mu_s)^2 q_{nn} + (\mu_n - B_{sn}^T\mu_s)(Q_{sn} + B_{sn}q_{nn})^T\mu_s$$

$$= \frac{1}{2}\mu_s^T(Q_{ss} + B_{sn}Q_{ns} + Q_{sn}B_{sn}^T + B_{sn}^T q_{nn}B_{sn})\mu_s + \frac{1}{2}\mu_n^2 q_{nn}$$

$$- \mu_n B_{sn}^T\mu_s q_{nn} + \frac{1}{2}(B_{sn}^T\mu_s)^2 q_{nn} + \mu_n Q_{ns}\mu_s + \mu_n q_{nn} B_{sn}^T\mu_s$$

$$- B_{sn}^T\mu_s Q_{ns}\mu_s - (B_{sn}^T\mu_s)^2 q_{nn}$$

$$= \frac{1}{2}\mu_s^T Q_{ss}\mu_s + \mu_n Q_{ns}\mu_s + \frac{1}{2}\mu_n^2 q_{nn}$$

$$= \frac{1}{2}\mu^T Q \mu . \qquad \square$$

Theorem (Chance Node Removal)

$$\frac{1}{2}\operatorname{trace}\{Q\Sigma\} + \frac{1}{2}\mu^T Q\mu + p^T\mu + r = \frac{1}{2}\operatorname{trace}\{Q_{new}\Sigma_{ss}\}$$

$$+ \frac{1}{2}\mu_s^T Q_{new}\mu_s + p_{new}^T\mu_s + r_{new} ,$$

where

$$Q_{new} = Q_{ss} + B_{sn}Q_{ns} + Q_{sn}B_{sn}^T + B_{sn}^T q_{nn} B_{sn} ,$$

$$p_{new} = p_s + (\mu_n - B_{sn}^T \mu_s)(Q_{sn} + B_{sn} q_{nn})^T \mu_s + B_{sn} p_n ,$$

and

$$r_{new} = r + \frac{1}{2} q_{nn} v_n + \frac{1}{2} (\mu_n - B_{sn}^T \mu_s)^2 q_{nn} + p_n(\mu_n - B_{sn}^T \mu_s) .$$

<u>Proof</u>

From Lemmas 1, 2, and 3,

$$\frac{1}{2} \operatorname{trace}\{Q\Sigma\} + \frac{1}{2} \mu^T Q \mu + p^T \mu + r$$

$$= \frac{1}{2} \operatorname{trace} \{Q_{new}\Sigma_{ss}\} + \frac{1}{2} q_{nn} v_n + \frac{1}{2} \mu_s^T Q_{new} \mu_s$$

$$+ \frac{1}{2}(\mu_n - B_{sn}^T \mu_s)^2 q_{nn} + (\mu_n - B_{sn}^T \mu_s)(Q_{sn} + B_{sn} q_{nn})^T \mu_s$$

$$+ p_{new}^T \mu_s - (\mu_n - B_{sn}^T \mu_s)(Q_{sn} + B_{sn} q_{nn})^T \mu_s$$

$$+ p_n(\mu_n - B_{sn}^T \mu_s) + r$$

$$= \frac{1}{2} \operatorname{trace}\{Q_{new}\Sigma_{ss}\} + \frac{1}{2} \mu_s^T Q_{new} \mu_s + p_{new}^T \mu_s + r_{new} . \qquad \square$$

CHAPTER 4

PROXIMAL DECISION ANALYSIS

4.1 Introduction

Proximal decision analysis is a method proposed by Howard [51 for analyzing the effect of uncertainty on large decision problems. Proximal analysis assumes

(1) the value function is approximately quadratic in decision and state variables,

(2) the state variables do not depend on decision variables, and

(3) either no state variables are observed prior to selection of decision variables or, for calculating the expected value of perfect information, all variables are observed prior to selection of decision variables.

Under these conditions, the mean vector and covariance matrix of the state variables are sufficient to analyze the effect of uncer tainty. The linear-quadratic influence diagram developed in Chapter 3 can be used for proximal analysis. This is done using the quadratic approximation as the value node deterministic function. In this chapter, influence diagram techniques are developed for proximal decision analysis, and Howard's entrepreneur's problem is revisited.

4.2 Proximal Analysis Model

For proximal analysis, we begin with a deterministic value function $V(X_s,X_d)$ that depends on a random state vector X that is not observed prior to selecting the decision vector X_d. We select X_d to maximize the expected value of the value function. We also assume that the probability distribution of X_s does not depend on X_d. If the state vector takes on its mean value μ_s, the reference optimal deterministic decision is

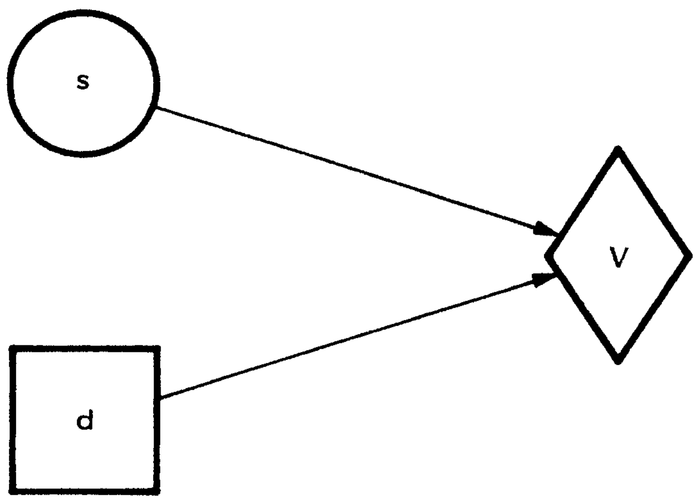
$$\mu_d = \max_{X_d}{}^{-1} V(\mu_s, X_d).$$

Letting $x^T = (x_s^T, x_d^T)$, we approximate the value function V by a second order Taylor series expansion about $\mu^T = (\mu_s^T, \mu_d^T)$.

$$\begin{aligned}
V(x) &\approx V(\mu) + \nabla^T V(\mu)(x-\mu) + \frac{1}{2}(x-\mu)^T \nabla^2 V(\mu)(x-\mu) \\
&= [V(\mu) - \nabla^T V(\mu)\mu + \frac{1}{2}\mu^T \nabla^2 V(\mu)\mu] + [\nabla V(\mu) - \nabla^2 V(\mu)\mu]^T x \\
&\quad + \frac{1}{2} x^T \nabla^2 V(\mu) x \\
&= r + p^T x + \frac{1}{2} x^T Q x \\
&= r + \begin{bmatrix} p_s \\ p_d \end{bmatrix}^T \begin{bmatrix} x_s \\ x_d \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x_s \\ x_d \end{bmatrix}^T \begin{bmatrix} Q_{ss} & Q_{sd} \\ Q_{ds} & Q_{dd} \end{bmatrix} \begin{bmatrix} x_s \\ x_d \end{bmatrix}.
\end{aligned}$$

To analyze the effect of uncertainty on this quadratic approximation to the value function, it is sufficient to know μ_s and Σ_{ss}, the mean and covariance matrix of the state vector.

The influence diagram representation of proximal analysis is shown in Fig. 4-1. Let X_s be indexed by $s = \{1,\ldots,n\}$. The state vector covariance matrix Σ_{ss} is factored and mapped onto a linear-quadratic influence diagram of scalar nodes similar to the normal influence diagram. For this representation, each scalar node $j \in s$ has an unconditional mean μ_j and a scalar v_j, which we shall call the



$$X = \begin{bmatrix} X_s \\ X_d \end{bmatrix}$$

○ = VECTOR CHANCE NODE
□ = VECTOR DECISION NODE

$$V(x) \approx 1/2\ x^T Q x + p^T x + r$$

Fig. 4-1 Proximal Analysis Influence Diagram conditional variance, although it is guaranteed to be the conditional variance only if the distribution of X_s is multivariate normal. For each pair of nodes $(i,j) \subset s \times s$, such that $i < j$, there is a scalar arc coefficient b_{ij}. Let

$$U_j = \begin{bmatrix} I_{j-1} & B_{\cdot j} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I_{n-j} \end{bmatrix} ,$$

where

$$B_{\cdot j} = [b_{1j}, \ldots, b_{j-1,j}]^T .$$

The covariance matrix factorization is as follows:

$$\Sigma_{ss} = U_n^T \ldots U_1^T \operatorname{diag}(v_1, \ldots, v_n)\, U_1 \ldots U_n .$$

Associated with each scalar decision node is its scalar component of μ_d, the reference optimal deterministic decision vector, as shown in Fig. 4-2, where μ_d is an m-vector, indexed by $d = \{n+1, \ldots, n+m\}$.

For proximal analysis, we want to maintain the covariance matrix of a non-normal distribution while using the normal influence diagram factorization and operations. Chapter 3 showed that a linear-quadratic influence diagram does maintain the covariance matrix, as long as the structure of the decision is a version of Fig. 3-8. For proximal analysis, $T = 1$, $s_1 = \emptyset$, $d_1 = d$, and $s_2 = s$ in Fig. 3-8.

4.3 <u>Open-Loop Analysis</u>

Deterministic sensitivity with respect to a scalar component X_j, where $j \in s \cup d$, may be performed by setting all the influence diagram

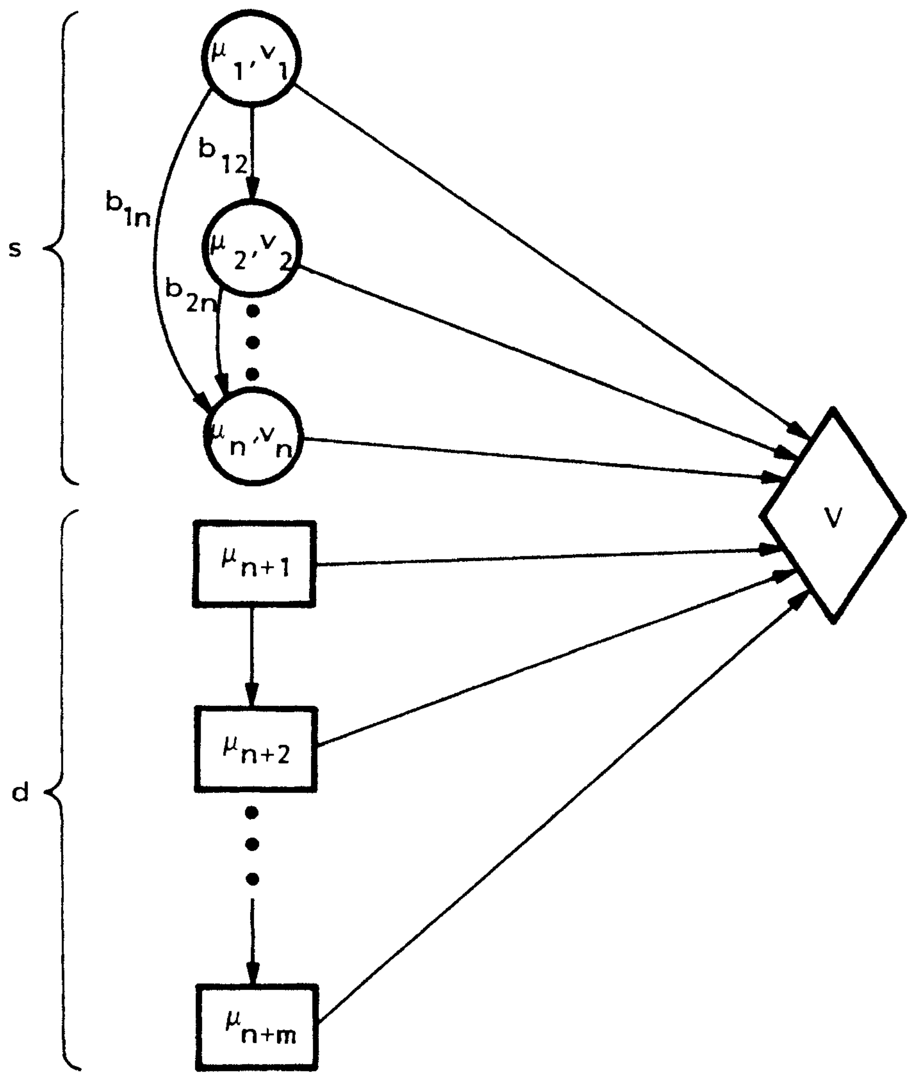


Fig. 4-2 Influence Diagram Factorization for Proximal Analysis

arc coefficients B_{ss} and conditional variances v_s to zero, removing all chance nodes (except X_j, if $j \in s$) into the value node, and removing all decision nodes (except X_j, if $j \in d$) into the value node. The remaining scalar values of Q, p, and r are the parameters for the quadratic sensitivity of the value to X_j given by

$$V(x_j) = r + px_j + \frac{1}{2} Qx_j^2.$$

If we also remove X_j, the value of r will be the deterministic maximum expected value of the reference decision choice of $X_d = \mu_d$.

For analyzing the open-loop effect of uncertainty on the expected value of the reference decision, we use the influence diagram factorization of the covariance matrix. The effect is obtained by first removing all the chance nodes and then removing all the decision nodes. The value of r should differ from the preceding deterministic open-loop calculation, if there is an effect of uncertainty on the expected value.

4.4 Closed-Loop Analysis

If we are permitted to observe the state vector prior to making our decisions, we select x_d as a function of x_s that maximizes $V(x_s,x_d)$. For our quadratic approximation to the value function, the optimal x_d is a linear function of x_s. The influence diagram representation of this closed-loop decision model is shown in Fig. 4-3, which is the same as Fig. 4-1, except that an informational arc has been added from s to d. This is a version of Fig. 3-8 with $T = 1$, $s_1 = s$, $d_1 = d$, and $s_2 = \emptyset$.

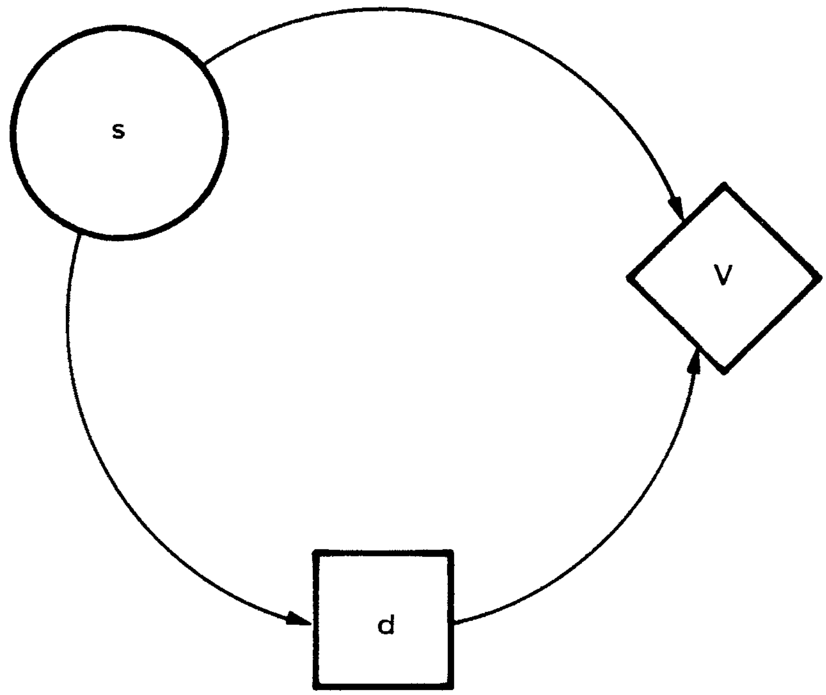


Fig. 4-3 Closed-Loop Influence Diagram

Deterministic sensitivity with respect to a scalar component X_j is performed by setting all the influence diagram arc coefficients and conditional variances equal to zero, removing all decision nodes (except X_j, if $j \in d$) into the value node, and removing all chance nodes (except X_j, if $j \in s$) into the value node. The remaining scalar values of Q, p, and r are the parameters for the quadratic sensitivity of the value to X_j. If we also remove X_j, the value of r will be the deterministic maximum expected value of the closed-loop decision.

For analyzing the effect of uncertainty on the expected value of the closed-loop decision, we use the influence diagram factorization of the covariance matrix. The effect of uncertainty is obtained by first removing all the decision nodes and then removing all the chance nodes. The value of r should differ from the preceding deterministic closed-loop calculation, if there is an effect on the expected value.

The expected value of clairvoyance (perfect information about the state vector) is the difference between the expected value of the open-loop optimal decision under uncertainty and expected value of the closed-loop optimal decision under uncertainty.

4.5 Wizardry

Wizardry is the ability to control a random state variable. Suppose we wish to analyze wizardry with respect to a scalar state variable X_j, where $j \in \{1,\ldots,n\}$. Letting $a = \{1,\ldots,j-1\}$, and $b = \{j+1,\ldots,n\}$, our open-loop influence diagram can be drawn as in Fig. 4-4. The arcs from X_j to X_b can be reversed so that X_j is a successor to X_a and X_b (Fig. 4-5). The new arc coefficients and conditional variances for X_a and X_b represent an influence diagram factorization of Σ_{cc},

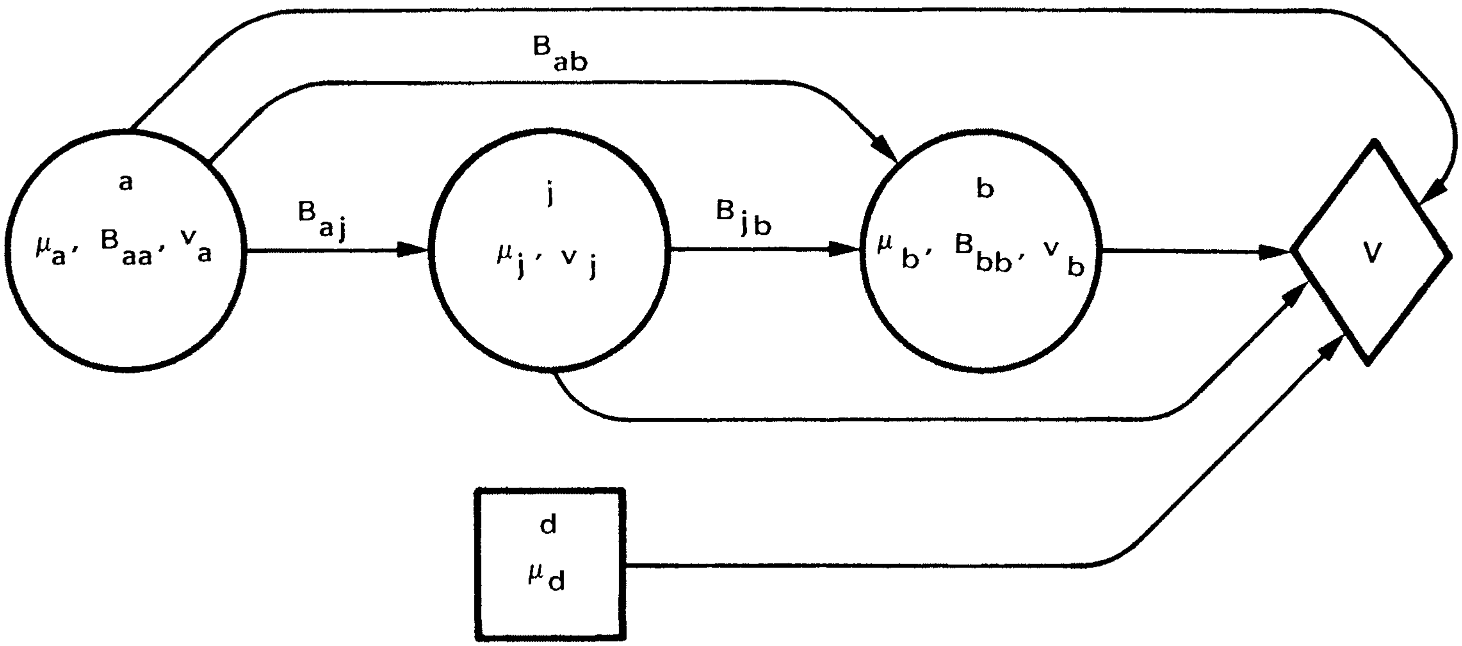


Fig. 4-4 Isolation of X_j for Wizardry

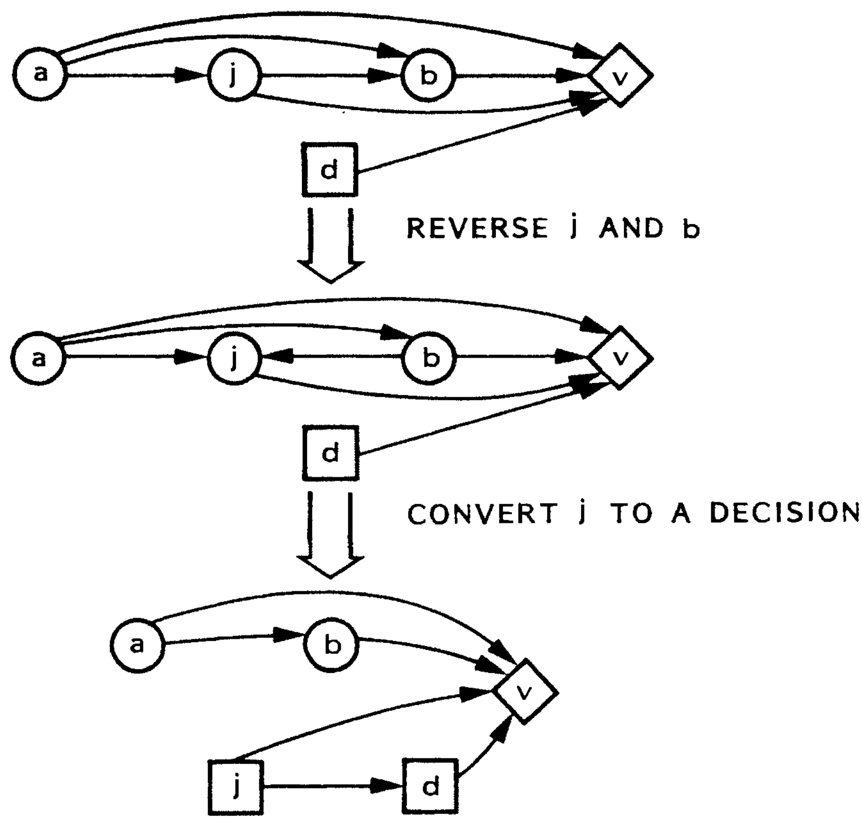


Fig. 4-5 Transformation for Wizardry on X_j

where $c = a \cup b$. To analyze wizardry on X_j, we drop all arcs from X_a and X_b to X_j and convert X_j to a decision node. For open-loop wizardry, we first remove $X_{d \cup \{j\}}$ into the value node and then remove X_b and X_a into the value node. For closed-loop wizardry, we remove X_b and X_a first and $X_{d \cup \{j\}}$ next.

4.6 Example

In this section, we present Howard's entrepreneur's problem [5] to demonstrate the use of influence diagrams in proximal decision analysis. An entrepreneur is deciding on a price for his product. The quantity sold at a given price P is $Q = q(P) + \Delta q$, where his demand curve is

$$q(P) = 80\ \ell n(50/P),$$

and Δq is a random variable with zero mean. His cost is $C = c(Q) + \Delta c$, where his cost function is

$$c(Q) = 700 + 4Q + 400(1 - \exp\{-Q/50\}),$$

and Δc is a random variable with zero mean and independent of Δq. His profit is

$$\pi(Q,C,P) = PQ - C.$$

An influence diagram representation of the entrepreneur's problem is shown in Fig. 4-6. The nodes q, Q, c, and C can be removed into the profit node by deterministic operations to reduce the entrepreneur's profit model to

$$\pi(\Delta q,\Delta c,P) = P[q(P) + \Delta q] - c(q(P) + \Delta q) - \Delta c.$$

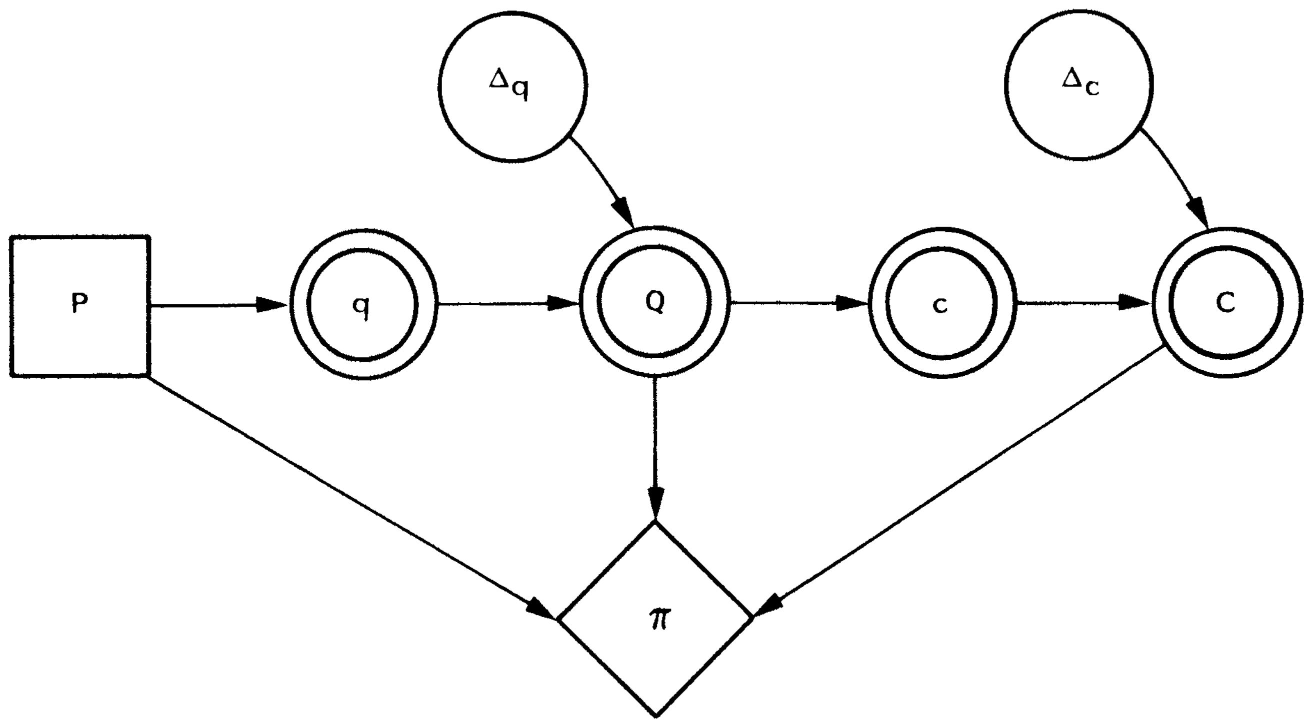


Fig. 4-6 The Entrepreneur's Problem

Letting $X_s = (\Delta q\ \Delta c)^T$, $\mu_s = (0\ 0)^T$, and $X_d = P$, the optimal deterministic decison is $\mu_d = P^* = 24.1$. The profit function at the optimum is

$$\pi(\mu) = 198.$$

The gradient is

$$\nabla\pi(\mu) = [17.58\ -1\ 0]^T,$$

and the Hessian is

$$\nabla^2\pi(\mu) = \begin{bmatrix} 0.0497 & 0 & 0.835 \\ 0 & 0 & 0 \\ 0.835 & 0 & -3.67 \end{bmatrix}.$$

Letting $\Delta p = P - \mu_d$, and redefining $X_d = \Delta p$, the sensitivity of profit as a function of $X^T = (\Delta q, \Delta c, \Delta p)$ is

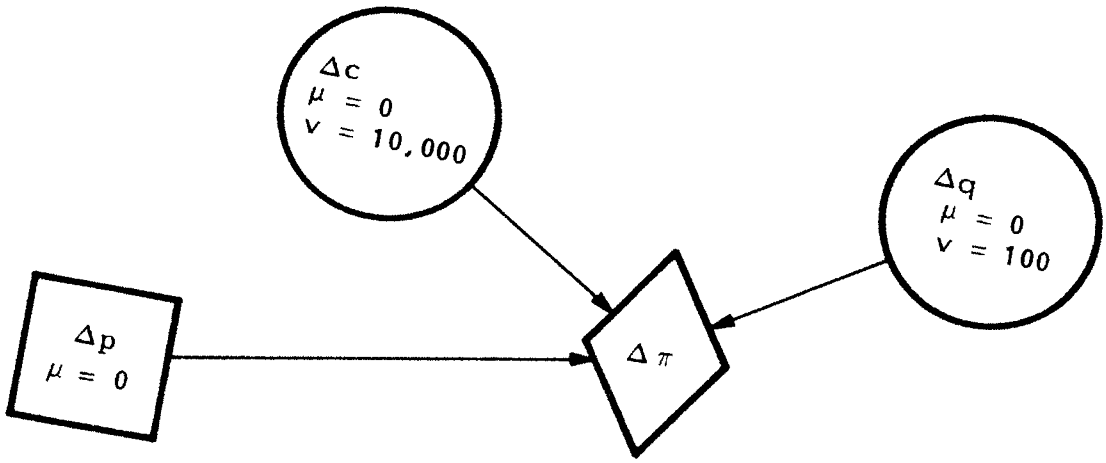
$$\begin{aligned} \Delta\pi(\Delta q, \Delta c, \Delta p) &= \pi(\mu) + [\nabla\pi(\mu)]^T x + \frac{1}{2} x^T \nabla^2 \pi(\mu) x \\ &= r + p^T x + \frac{1}{2} x^T Q x. \end{aligned}$$

An influence diagram of this sensitivity model is shown in Fig. 4-7.

To calculate deterministic sensitivity of $\Delta\pi$ to Δq, we remove Δc and Δp into $\Delta\pi$ using normal influence diagram rules (Fig. 4-8), assuming the conditional variance $v(\Delta c) = 0$. Following removal, the sensitivity is

$$\Delta\pi(\Delta q, 0, 0) = 0.0249(\Delta q)^2 + 17.58\Delta q.$$

Similar removals can be performed for deterministic sensitivity of $\Delta\pi$ to Δc and Δp.



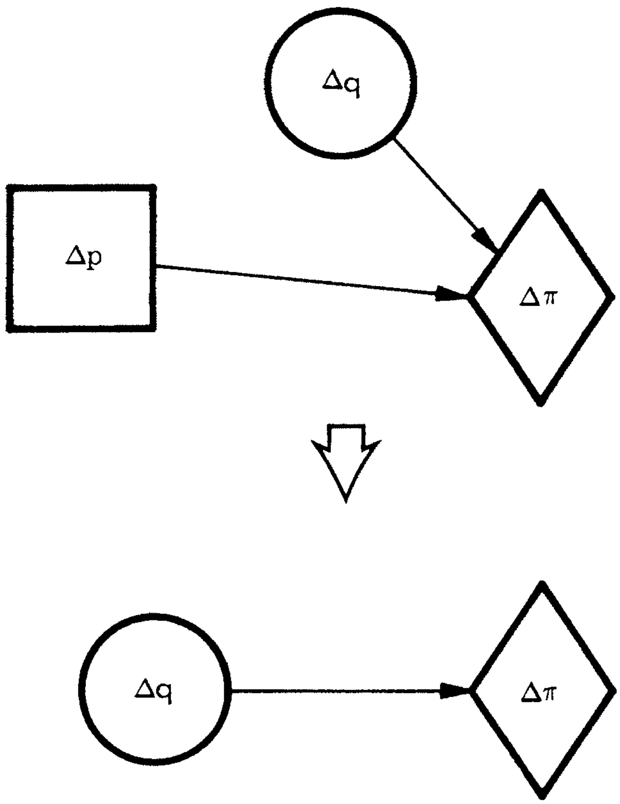
$$x = \begin{bmatrix} \Delta q \\ \Delta c \\ \Delta p \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.0497 & 0 & 0.835 \\ 0 & 0 & 0 \\ 0.835 & 0 & -3.67 \end{bmatrix}$$

$$p = \begin{bmatrix} 17.58 \\ -1 \\ 0 \end{bmatrix}$$

$$r = 0$$

Fig. 4-7 Sensitivity Analysis Diagram



$$Q = \begin{bmatrix} 0.0498 & 0.835 \\ 0.835 & -3.67 \end{bmatrix}$$

$$p = \begin{bmatrix} 17.58 \\ 0 \end{bmatrix}$$

$$r = 0$$

$$Q = 0.0498$$

$$p = 17.58$$

$$r = 0$$

Fig. 4-8 Sensitivity to Δq

The effect of uncertainty on the open-loop decision is calculated by removing Δc, Δq, and Δp into $\Delta\pi$, with $v(\Delta c) = 10000$ and $v(\Delta q) = 100$. This is shown in Fig. 4-9. Upon completion of the removals, $r = 2.49$ is the effect of uncertainty on the entrepreneur's expected profit.

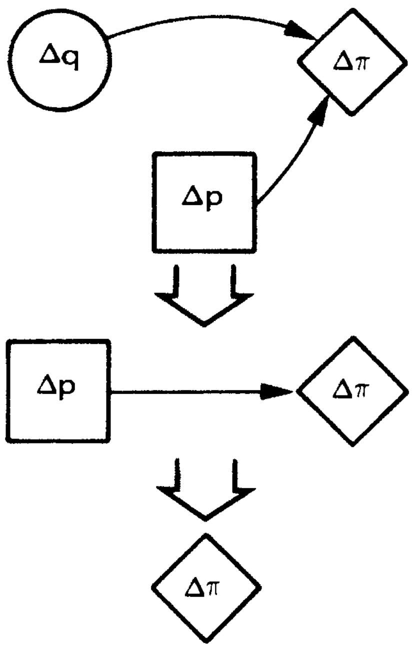
Closed-loop sensitivity of Δp to Δq and Δc is calculated by removing Δp into $\Delta\pi$, as in Fig. 4-10. Note there is no sensitivity of Δp to Δc, as shown by Howard. Removing Δc and Δq into $\Delta\pi$ next, the expected profit of the closed-loop decision is $r = 11.99$. The expected value of clairvoyance on Δq and Δc is 9.5, the difference of the closed-loop (Fig. 4-10) and open-loop (Fig. 4-9) values of r after all nodes have been removed into the value node.

4.7 Conclusion

Proximal decision analysis is a straightforward extension of the linear quadratic influence diagram presented in Chapter 3. As such, extension of proximal analysis to sequential decisions is limited to the structure shown in Fig. 3-8.

Additional insight into assumptions and analysis techniques are provided by the influence diagram. This includes clarification of the distinction between open-loop and closed-loop analysis, demonstration of sensitivity analysis techniques, and illustration of assumptions required to analyze wizardry.

An area for further investigation is combining iterative nonlinear programming and proximal decision analysis using influence diagrams.



$$Q = \begin{bmatrix} 0.0498 & 0.835 \\ 0.835 & -3.67 \end{bmatrix}$$

$$p = \begin{bmatrix} 17.58 \\ 0 \end{bmatrix}$$

$$r = 0$$

$Q = -3.67$
$p = 0$
$r = 2.49$

$r = 2.49$

Fig. 4-9 Open-Loop Effect of Uncertainty

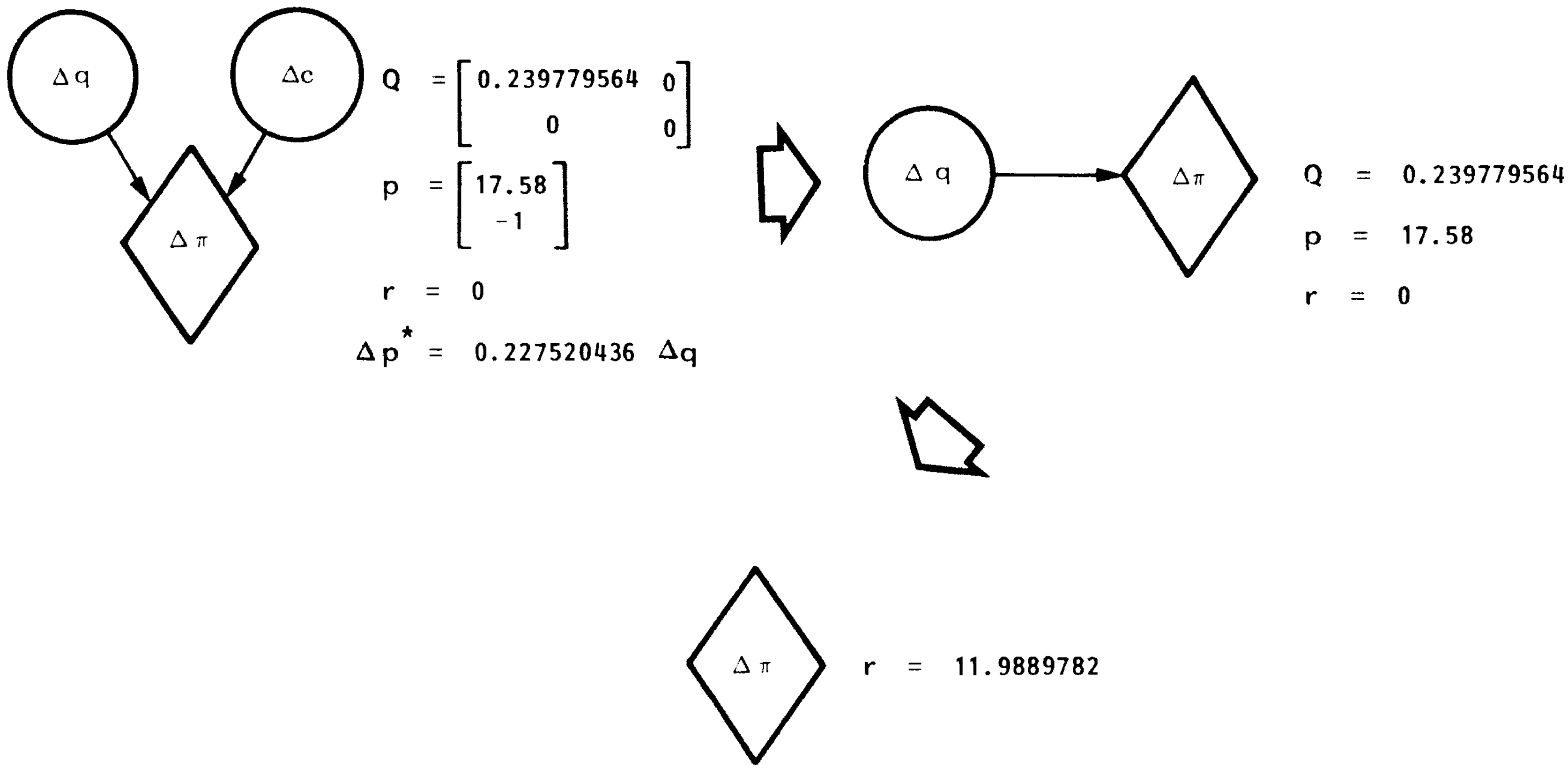


Fig. 4-10 Closed-Loop Analysis

BIBLIOGRAPHY

[1] Bierman, Gerald J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 1977.

[2] Box, George E. P. and Gwilym L Jenkins, Time Series Analysis: Forecasting and Control, Revised Edition, Holden-Day, San Francisco, Ca., 1976.

[3] Bryson, Arthur E and Yui-Chi Ho, Applied Optimal Control Optimization Estimation and Control, Revised Printing, Hemisphere Publishing, Washington, D.C., 1975.

[4] Graybill, Franklin A., Theory and Application of the Linear Model, Duxbury Press, North Scituate, Mass., 1976.

[5] Howard, Ronald A., Proximal Decision Analysis, Management Science Vol. 17, No. 9, May 1971, pp. 507-541.

[6] Howard, Ronald A. and James E. Matheson, "Influence Diagrams," in Readings on the Principles and Applications of Decision Analysis, Vol. 2, ed. by Ronald A. Howard and James E. Matheson, Strategic Decisions Group, Menlo Park, Ca., 1984, pp. 719-762.

[7] Kalman, R E., A New Approach to Linear Filtering and Prediction Problems, Transactions of the American Society of Mechanical Engineers, Series D, Vol. 82, March 1960, pp. 35-50.

[8] Khatri, C. G., "Characterizations of Multivariate Normality II. Through Linear Regressions, Journal of Multivariate Analysis, Vol. 9, 1979, pp. 589-598.

[9] Oren, S. S., "A Conjugate Directions Method for Subjective Assessment of Normal Distributions, Journal of Optimization Theory and Applications, Vol. 33, No. 1, January 1981, pp. 25-36.

[10] Pratt, J., "Risk Aversion in the Small and Large," Econometrica, Vol. 32, 1964, pp. 123-136.

[11] Searle, S. R., Linear Models, John Wiley & Sons, New York, 1971.

[12] Shachter, Ross D., "Probabilistic Inference and Influence Diagrams," Department of Engineering-Economic Systems, Stanford University, submitted to Operations Research, March 1986.

[13] Shachter, Ross D., "Evaluating Influence Diagrams," Department of Engineering-Economic Systems, Stanford University, to appear in Operations Research, 1986

[14) Strang, Gilbert, Linear Algebra and Its Applications, Second Edition, Academic Press, New York, 1980.

[15] Thornton, C. L. and G. J. Bierman, 'tram-Schmidt Algorithms for Covariance Propagation," Proceedings of the 1975 IEEE Conference on Decision and Control, Houston, 1975, pp 489-498

[16] Wright, Sewall, "The Method of Path Coefficients," The Annals of Mathematical Statistics, Vol. 5, No. 3, September 1934, pp. 161-215.

[17] Yule, G. Udny, "On the Theory of Correlation for Any Number of Variables, Treated by a New System of Notation," Proceedings of the Royal Society of London, Series A, Vol. 79, September 1907, pp. 182-193.