AN ALTERNATIVE ALGORITHM FOR DISCRETE-TIME FILTERING

THESIS

Frederick H. Zeitz III
Major, USAF

AFIT/GE/ENG 91J-06

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(1)

AFIT/GE/ENG 91J-06

AN ALTERNATIVE ALGORITHM FOR DISCRETE-TIME FILTERING

THESIS

Frederick H. Zeitz III
Major, USAF

AFIT/GE/ENG 91J-06

Approved for public release; distribution unlimited

Acknowledgments

This thesis is an investigation of the properties of the influence diagram, when applied to discrete-time filtering. The thesis was accomplished during a three-year, part-time Electrical Engineering Masters Degree program.

This thesis is the result of the time and efforts of Maj Joe Tatman, who dedicated an entire quarter of the school year to the task of teaching me about influence diagrams. There are immeasurable contributions by Dr. Pete Maybeck, who suggested the topic, taught me the necessary tools, and inspired me to treat this topic with the discipline it deserves. There are few truly gifted instructors, and Pete Maybeck is one of them.

Many students helped me on the way to completion of this thesis. The spirit of teamwork and cooperation at AFIT will remain with me always. I give special thanks to Capt Mark Gallagher for the value of his time and experience.

There is no way this thesis would have been possible without the understanding, motivation, and participation of my family. My wife Cheryl, and my daughters Candice and Angela, were incredible during this effort. This thesis is a monument to your love. I am truly fortunate to have such a family. I love you all.

This thesis is dedicated to the memory of Julius Becsey. I will treasure your memory. In many ways, you made this possible.

Frederick H. Zeitz III

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1991	Master's Thesis

4. TITLE AND SUBTITLE
An Alternative Algorithm for Discrete-Time Filtering

5. FUNDING NUMBERS

6. AUTHOR(S)
Frederick H. Zeitz III, Major, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/91J-06

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The discrete-time Kalman filter is a conditional mean estimator of the states of a linear stochastic process, conditioned on the previous state and current measurements. It assumes that the states and noise inputs can be represented as jointly Gaussian random variables.

The influence diagram is a decision analysis tool. Under certain conditions, it can represent continuous, jointly Gaussian random variables. The conditioning order of the random variables may be changed (using Bayes' rule) so that any random variable can be conditioned on any other subset of random variables in the diagram. Under these conditions, an influence diagram can represent the states, measurements, and initial conditions of a linear stochastic process. It too can be a conditional mean estimator of the states of a linear stochastic process, and is an alternative algorithm for the Kalman filter. The influence diagram algorithm for the Kalman filter uses a factored covariance matrix. It is similar to other factored forms of the Kalman filter such as the U-D filter. It can be faster than other factored forms of the Kalman filter, but retain their improved numerical properties.

14. SUBJECT TERMS
Kalman filter, influence diagram, LDU Decomposition

15. NUMBER OF PAGES
157

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to **stay within the lines to meet optical scanning requirements.**

Block 1. Agency Use Only (Leave Blank)

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Names(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency. Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of ..., To be published in .... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availablity Statement. Denote public availability or limitation. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR)

DOD	- See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	- See authorities
NASA	- See Handbook NHB 2200.2.
NTIS	- Leave blank.

Block 12b. Distribution Code.

DOD	- DOD - Leave blank
DOE	- DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports
NASA	- NASA - Leave blank
NTIS	- NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

Table of Contents

Page

Acknowledgments ii

Table of Contents iii

Abstract vi

I. Introduction 1

1.1 Summary of Research 2

1.2 Thesis Objectives 7

1.3 Thesis Overview 8

II. Influence Diagrams 9

2.1 Discrete Random Variables 9

2.2 Example of Discrete Random Variables 13

2.3 Continuous Random Variables 18

2.4 Example of the Gaussian Influence Diagram 23

2.5 Matrix Representation of the Influence Diagram 28

2.6 The Influence Diagram for Discrete-Time Filtering 31

2.6.1 Numerical Example of the Discrete-Time Kalman Filter. 45

2.6.2 Numerical Example of Discrete-Time Influence Diagram Filter. . 53

2.6.3 Chapter Summary. 57

III. Implementing the Influence Diagram 71

3.1 The Recursive Algorithm 74

3.2 Vector Measurement Update 76

3.3 Efficient Implementation Form 82

3.4 Operations Count 87

Page

3.4.1 Pipeline Processing. 88

3.5 Chapter Summary 92

IV. Equivalent Operations 93

4.1 Equivalence to Matrix Operations 93

4.1.1 Exchanging a Node with a Predecessor. 94

4.1.2 Exchanging a Node with a Successor. 98

4.1.3 Matrix Representation of Node Reversal. 103

4.1.4 Comparison with Kalman Filtering. 105

4.1.5 Comparison with U-D Filtering. 108

4.2 Chapter Summary 110

V. Numerical Properties 112

5.1 Matrix Operations 112

5.1.1 Stability of Calculations. 114

5.2 Error Analysis 117

5.3 Numeric Examples 119

5.3.1 Scalar Update Example. 119

5.3.2 Bierman's Example. 122

5.3.3 Potential Errors in the U-D Filter. 130

5.3.4 Numeric Analysis Generalizations. 132

5.4 Chapter Conclusions 133

VI. Summary 134

6.1 Conclusions 134

6.2 Other Applications 135

6.2.1 Fault Detection/Hypothesis Testing. 135

6.2.2 Inverse Covariance. 135

6.2.3 Optimal Control. 136

Page

6.2.4 Parameter Estimation. 137

6.2.5 Discrete Probability Applications. 137

6.3 Recommendations 137

Bibliography 139

Appendix A. Appendix A: Operations Count 140

Appendix B. Appendix B: Operations Count, Parallel Processing 146

Vita 149

Abstract

The influence diagram is a decision analysis tool. It shows how random variables and decisions affect (or influence) a value function. The influence diagram can also be used for probabilistic analysis by ignoring decision variables and the value function. In this case, the influence diagram represents a joint distribution function of the random variables.

The influence diagram represents a joint distribution function in conditional form, with one random variable in unconditional form, a second random variable conditioned on the first, a third conditioned on the first two, and so on. If a random variable is conditioned on another, then the conditioning order can be reversed with Bayes' rule. In this manner, any random variable can eventually be expressed in unconditional form, or as being conditioned on any other subset of random variables.

Under certain conditions, the influence diagram can represent continuous random variables. When the random variables are continuous and jointly Gaussian, then each random variable is completely specified by its mean and variance. The influence diagram calculates the conditional mean and conditional variance of a given random variable, given a subset of the remaining random variables The conditional mean and variance are sufficient to completely specify the conditional density function. The conditioning order of the random variables may be changed (again using Bayes' rule) so that any random variable can be conditioned on any other subset of random variables in the diagram.

The discrete-time Kalman filter is a conditional mean estimator of the states of a linear stochastic process, conditioned on the previous state and current measurements An influence diagram can represent the states, measurements, and initial conditions of a linear stochastic process. Under these conditions, the influence diagram is also a conditional mean estimator of the states of

a linear stochastic process. The influence diagram algorithm becomes an alternative algorithm for discrete-time filtering.

Two important characteristics of any algorithm are its speed and numerical properties. The speed of the algorithm is determined by the number and type of mathematical operations required by a digital computer implementing the algorithm. The numerical properties of an algorithm depend upon how it handles the roundoff errors that are inherent in a digital computer. These roundoff errors cause the computed values to deviate from the true values that would be computed on a machine with no roundoff errors (infinite wordlength). One measure of numerical properties is the stability of an algorithm. The errors caused by a stable algorithm are related to rounding errors in the original values input to the algorithm.

The speed of the Kalman filter is well known. The numerical properties are known also, but can be unsatisfactory. The conventional Kalman filter algorithm is not a stable algorithm and has numerical precision problems under certain conditions. One solution to the numerical problems of the conventional Kalman filter has been to use factored forms of the covariance matrix as the basis for computation. These factored forms of the Kalman filter are slower than the conventional form, but allow better numerical properties. The U-D factored form of the filter offers the best numeric properties with reasonable speed.

This research focused on the numeric properties and the speed of the influence diagram. It revealed that the influence diagram algorithm for discrete-time filtering uses a factored form of the covariance matrix. This factored form is essentially a mirror-image of the factorization used by the U-D filter.

Previous research showed that the speed of the influence diagram was equivalent to the U-D filter. However, this research revealed circumstances that allow significant savings in the number of operations required by the influence diagram. These savings do not make the influence diagram faster than the Kalman filter, but they do make it faster than other factored forms.

This research also showed a link between the influence diagram algorithm and matrix operations. These matrix operations are known to be stable and demonstrate the stability of the influence diagram algorithm. The errors caused by the influence diagram are directly related to the types of errors caused by the inversion of a unit triangular matrix. These errors are very similar to the errors that occur in the U-D filter.

The influence diagram allows a pictorial view of the conditional mean estimator. It can be used for discrete-time filtering, resulting in excellent numeric properties. Although the influence diagram is not as fast as the Kalman filter, it appears to offer the best trade of speed for numeric precision of any known discrete-time filtering algorithm.

AN ALTERNATIVE ALGORITHM FOR DISCRETE-TIME FILTERING

I. Introduction

The influence diagram is a tool for decision analysis [10, 11, 12, 14]. It is a pictorial description of conditional relationships between a given set of random variables. If one random variable is conditioned on another, then the conditioning order can be reversed by using Bayes' rule. The influence diagram can be used to show changes in the relationships between the random variables when the conditioning order changes.

Associated with the influence diagram are the conditional and unconditional probability distributions for each of the random variables in the diagram. Taken together, the conditioning relationships and the associated distributions are sufficient to describe a joint distribution function for the given set of random variables.

If all the random variables of a set are continuous and jointly Gaussian, then the joint density function can be fully specified by a mean vector and a covariance matrix. For jointly Gaussian continuous random variables, the influence diagram can be used as an alternative expression for the joint density function. Instead of a covariance matrix, the influence diagram specifies the unconditional and conditional variances of the random variables, and the conditional relationships between them.

The discrete-time Kalman filter is an optimal estimator for the states of a linear, stochastic system [7]. It can be derived by assuming that state estimates can be expressed as jointly Gaussian random variables. Using linear operations and Bayes' rule, the estimate of the state at any time can be calculated as a vector of conditional means and a conditional covariance matrix, conditioned on current and previous measurements.

The linear operations of a linear stochastic system can also be expressed in influence diagram form. Furthermore, the influence diagram can be used to represent the conditional vector of means and conditional covariance matrix, conditioned on previous and current measurements, or equivalently, the most recent previous state estimates and the current measurements. Thus, the influence diagram affords an alternative algorithm for discrete-time filtering.

One limitation of the Kalman filter is that, under certain circumstances, it can have problems with numeric accuracy due to fixed computer wordlength and roundoff error [2, 7]. There are various ways of addressing this type of problem. A simple, but perhaps costly way is to increase the wordlength of the computer. Another way is to use alternative algorithms which have better numerical properties than Kalman's original equations, with an associated increase in computational requirements. One alternative algorithm which offers a large benefit in numeric accuracy, at a moderate increase in computational burden, is the U-D factored form of the Kalman filter [2]. Because of these properties, the U-D filter will be a baseline for comparison in the remainder of this thesis.

1.1 Summary of Research

The Gaussian influence diagram draws from different fields of expertise [1, 6, 8, 9, 10, 11, 12, 17]. It is a graphical method of showing the conditional means and variances of jointly Gaussian random variables. It represents the relationship between such variables as the conditioning of one on another. The algorithm for manipulating the Gaussian influence diagram takes advantage of the special characteristics of the joint Gaussian distribution.

The theory behind the continuous random variable form of the influence diagram dates from early work in statistical analysis. Some of the notation for the influence diagram can be traced to Yule's notation for partial regression coefficients of multiple variables [17]. The influence diagram depicts the relationships between conditional random variables and those random variables upon which they are conditioned. Either Mood and Graybill [8:pp. 198–215] or Anderson [1:pp. 5–34]

present a thorough summary of the characteristics of conditional Gaussian random variables. Some of the applicable properties are repeated here for continuity.

If $\mathbf{x}$ is a Gaussian random vector that has components $x_1, x_2, \ldots, x_p$, and the components x_1 through x_q are conditioned on the components x_{q+1} through x_p, then the mean of any of the conditioned variables can be expressed as a linear function of the realizations of the conditioning variables. This linear relationship can be expressed in terms of a matrix of regression coefficients. If x_i is a component of the conditioned set of variables, and x_j is a component of the conditioning set of variables, then the i,jth element of this matrix is $\beta_{ij.q+1,q+2,\ldots,j-1,j+1,\ldots,p}$, where the subscript implies that this scalar is the partial derivative of x_i's conditional mean with respect to x_j's value. The subscript terms after the period are the conditioning variables for the coefficient.

The matrix of regression coefficients is closely related to the covariance matrix. If a Gaussian random vector is partitioned into two random vectors $\mathbf{x}$ and $\mathbf{y}$, then the covariance matrix can be partitioned as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xy} \\ \mathbf{P}_{yx} & \mathbf{P}_{yy} \end{bmatrix} \tag{1}$$

The matrix of regression coefficients of $\mathbf{x}$ on $\mathbf{y}$ is $\mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}$. If $\mathbf{m}_x$ represents the unconditional mean vector for $\mathbf{x}$, and $\mathbf{m}_y$ represents the unconditional mean vector for $\mathbf{y}$, then the conditional mean of $\mathbf{x}$ given $\mathbf{y}$ is given by the equation (also called the regression function):

$$\mathbf{m}_{x|y} = \mathbf{m}_x + \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}(\rho - \mathbf{m}_y) \tag{2}$$

where ρ is a dummy variable for the realization of the random vector $\mathbf{y}$. Furthermore, the conditional variance of $\mathbf{x}$ given $\mathbf{y}$ is:

$$\mathbf{P}_{x|y} = \mathbf{P}_{xx} - \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx} \tag{3}$$

These equations show that, for a given set of conditioning variables, the matrix of regression

coefficients and the conditional covariance matrix are fixed. Furthermore, the conditional mean is a linear function of the realizations of the conditioning random variables [1:pp. 27–30].

In his 1986 doctoral dissertation, Kenley proved that the covariance matrix for a Gaussian random vector could be expressed in terms of a matrix of regression coefficients and a matrix of conditional variances [6:pp. 29–31]. Kenley's regression coefficient matrix is not constructed in the same way as the matrix $\mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}$ given above. Instead of using the same conditioning variables (the random vector $\mathbf{y}$), for all the regression coefficients, Kenley constructed his matrix in the following way.

Assume one variable is chosen from the random vector. The choice may be arbitrary, but for the purposes of this explanation, it will be labeled x_1. A second variable, perhaps arbitrary also, is chosen and expressed in conditional form on the first variable. This second variable will be labeled x_2. In order to express this second variable in conditional form, it is sufficient to calculate the regression coefficient $\beta_{x_2x_1}$ and the conditional variance of x_2 given x_1. A third variable is chosen and conditioned on the first two using the regression coefficients $\beta_{x_3x_1}$ and $\beta_{x_3x_2.x_1}$. The variance of the third variable is also expressed as a conditional variance, conditioned on the first two random variables. The process continues until all random variables in the vector have been chosen and conditioned on the previously chosen variables.

The regression coefficients just calculated can be expressed in matrix form as a strictly upper triangular matrix $\mathbf{B}$. Kenley uses conventional row-column notation for this matrix. Kenley's new notation is more convenient for influence diagrams because, as will be shown later, there is no need to keep track of the conditioning variables with subscripts. The influence diagram graphically depicts all conditioning variables. In addition, the order of the first two numbers in the subscript is reversed to make the notation compatible with matrix operations. The $\mathbf{B}$ matrix can be expressed

using either notation as:

$$\mathbf{B} = \begin{bmatrix} 0 & \beta_{21} & \beta_{31} & \beta_{41} & \cdots & \beta_{n1} \\ & 0 & \beta_{32.1} & \beta_{42.1} & \cdots & \beta_{n2.1} \\ & & 0 & \beta_{43.21} & \cdots & \beta_{n3.21} \\ & 0 & & 0 & & \vdots \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} = \begin{bmatrix} 0 & b_{12} & b_{13} & b_{14} & \cdots & b_{1n} \\ & 0 & b_{23} & b_{24} & \cdots & b_{2n} \\ & & 0 & b_{34} & \cdots & b_{3n} \\ & 0 & & 0 & & \vdots \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \tag{4}$$

The conditional variances can also be expressed in matrix form. This matrix is diagonal, ordered as the variables were chosen earlier, $x_1, x_2, x_3, \ldots, x_n$. The conditioning variables for any conditional variance lie above it on the diagonal. Since x_1 is first, it is not conditioned on any other variables and the variance is in unconditional form. The notation for the variance of the ith random variable is v_i, whether the variable is expressed in conditional or unconditional form. The diagonal matrix **D** is expressed as:

$$\mathbf{D} = \begin{bmatrix} v_1 & & & & \\ & v_2 & & 0 & \\ & & v_3 & & \\ & 0 & & \ddots & \\ & & & & v_n \end{bmatrix} \tag{5}$$

The influence diagram was introduced in decision theory for visualizing the "influence" of random variables on decisions [10, 11]. It graphically depicts the relationships between random variables, deterministic variables, decisions, and value functions. For this research however, only random and deterministic variables will be considered. Influence diagrams are relatively new to the field of decision theory and are not widely used outside of the field. In order to make the rest of this

research easier to understand, Chapter 2 is an introduction to influence diagrams as specifically applied to probabilistic analysis and discrete-time filtering.

Initially, the random variables in influence diagrams were assumed to be discrete random variables or deterministic variables, each with discrete probability distribution functions [10]. Kenley later derived the mathematics for influence diagrams with continuous random variables. He specifically applied influence diagrams to the case of jointly Gaussian random variables, although he also extended the analysis to other continuous random variables [6:pp. 12–34].

Kenley also demonstrated that the influence diagram can be used for discrete-time filtering, and is equivalent to the discrete-time Kalman filter [6:pp. 52–106]. He showed that, when the influence diagram is used for discrete-time filtering, it requires about the same number of floating point operations as the U-D factored form of the Kalman filter [6:pp. 89–106].

The discrete-time Kalman filter is known to have problems with roundoff error and numeric instability when implemented on fixed-wordlength computers [2, 4, 5]. Furthermore, the U-D covariance factorization algorithm has been shown to have better numerical properties than the conventional Kalman filter algorithm [2, 4]. However, to the author's knowledge, there has been no research into the numeric properties of the influence diagram. Specifically, there has been no comparison of the numeric properties of the influence diagram algorithm for discrete-time filtering to the conventional Kalman filter or to the U-D factored form of the Kalman filter.

This research will investigate properties of the influence diagram algorithm and compare them to the properties of the U-D filter. Specifically, this research will compare the efficiency and numeric properties of these two algorithms. The numeric properties of the influence diagram algorithm will be shown by relating them to matrix operations. The numerical stability, properties, and error bounds of these matrix operations come from Wilkinson [15, 16].

1.2 Thesis Objectives

This thesis is intended to build upon the research of Kenley. It will explore the properties of the influence diagram, specifically as applied to discrete-time filtering. Additionally, the influence diagram algorithm will be compared to both the conventional Kalman filter algorithm and the U-D filter. The results will be a comparison of the advantages and disadvantages of each filter algorithm.

There are two reasons for using the U-D filter for comparison. One reason is the similarity between the U-D filter and the influence diagram. This similarity will be a topic later in the thesis. The other reason is that the U-D filter represents a standard in terms of numerical accuracy and computational loading. Its properties are well researched and documented [2, 3, 4, 5].

This analysis of the influence diagram will begin with an overview of its principles and purpose. Even though the influence diagram can be used for decision analysis and maximizing (or minimizing) multivariate value functions, these aspects will not be discussed. Instead, this thesis will address only the probabilistic applications of the influence diagram. The discussion will be a review of the work of Schacter, Kenley, and Tatman [10, 11, 6, 12, 14].

The main application of the influence diagram, for the purposes of this research, is an alternative algorithm for discrete-time Kalman filtering. For this reason, there will be a brief discussion of the theory of the Kalman filter, paralleled with an explanation of how the influence diagram implements the same operations. Again, this explanation will be a review of previous work by Kenley [6:pp. 52–106].

The main body of this research will follow the review and will focus on three major areas. The first topic will be a discussion of how the influence diagram can be solved efficiently. It will be a demonstration of practical methods for implementation. A result of this demonstration will be a tally of the mathematical operation needed for the complete filter. It will be shown that, under certain conditions, the operation count can be reduced from previous estimates [6:pp. 89–106].

More importantly, the algorithm lends itself to a pipeline architecture which can lead to increased calculation speed.

The second major topic will be a comparison of the influence diagram with matrix operations. The matrix operations add insight into the mathematics involved and are easier to analyze. This relationship to matrix operations forms the basis for the third major topic, numerical properties.

To this author's knowledge, there has been no previous literature on the numerical properties of the influence diagram. Neither has the influence diagram been included in any experimental comparisons of different Kalman filter algorithms. No such experimental comparisons were made in this research. Instead, this research explores the theoretical numerical properties, based on the known numerical properties of the matrix operations.

There are some important results from the theoretical research in this thesis. First, it will be shown that the influence diagram algorithm, like the U-D factored filter, uses a stable algorithm to calculate the means and conditional variances of random variables. Second, the numeric properties of this algorithm can be compared to the properties of the U-D filter. Third will be an insight into the conditions that do cause numerical problems for the influence diagram.

1.3 Thesis Overview

This chapter reviewed some of the existing research in influence diagrams and their use as an algorithm for discrete-time filtering. It also indicated the outline for the rest of this thesis. Chapter 2 will be a tutorial approach to influence diagrams. It will attempt to explain the applications and operation of the influence diagram, specifically its application to discrete-time filtering. Chapter 3 will demonstrate efficient implementation of the influence diagram, including operation counts. Chapter 4 will demonstrate the relationship of influence diagram operations to matrix operations. This will lead in to Chapter 5, which will be a description of the numerical properties. Finally, Chapter 6 will conclude and make recommendations for future research.

II. Influence Diagrams

The influence diagram is a recent tool in decision analysis and is not well known beyond this field. This chapter is a short description of the influence diagram, concentrating on its use for probabilistic analysis. For a more rigorous description of influence diagrams, refer to Tatman, Schacter, or Kenley [6, 10, 11, 14].

An influence diagram represents deterministic, random, decision, and value function variables by using a node for each variable. For probabilistic analysis, only two types of nodes are necessary. A single circle node represents a random variable. A double circle node represents a deterministic variable or, in other words, a variable that can be calculated exactly as a function of other variables.

In general, a random variable can have a discrete probability distribution, a continuous probability density, or a combination of the two. Initially, influence diagrams were only used for random variables with discrete probability distributions; therefore, this discussion will begin with this case also.

2.1 Discrete Random Variables

Assume two random variables, x and y, each with discrete probability distributions. The variable x can take on discrete values $x_1, x_2, x_3, \ldots, x_n$ with probabilities $p(x_1), p(x2), p(x_3), \ldots, p(x_n)$ such that $P[x = x_i] = p(x_i)$. The probabilities for the random variable y can be similarly defined as $P[y = y_j] = p(y_j)$ where the random variable y can take on the discrete values $y_1, y_2, y_3, \ldots, y_m$.

For two such random variables, a joint distribution can be defined. This joint distribution assigns a probability to each discrete pair of values (realizations) of the random variables. The two random variables can be associated with a random vector where the elements of the vector are the individual random variables. The joint distribution for these random variables becomes the probability distribution for the two dimensional random vector. For each pair of discrete values

x_i and y_j, the probability P[x = x_i, y = y_j] = $p(x_i, y_j)$ is the probability that x and y take on the values x_i and y_j.

A joint distribution function of two variables can also be expressed in terms of the conditional distribution of one random variable, conditioned on the outcome of the other random variable. Bayes' rule demonstrates the mathematics associated with this expression. One form of Bayes' rule is $p(x_i, y_j) = p(x_i|y_j)p(y_j)$, where $p(x_i|y_j)$ means the probability that x takes on the value x_i, given that y has taken on the value y_j. The joint probability distribution can also be expressed, again using Bayes' rule, in terms of the conditional distribution of y given a value of x. This is written as $p(x_i, y_j) = p(y_j|x_i)p(x_i)$.

The probability $p(y_j)$ can be calculated by summing the probabilities of $p(x_i, y_j)$ over all possible values of x_i. This distribution is also referred to as the marginal or unconditional probability distribution of y and can be represented in equation form as:

$$p(y_j) = \sum_{x_i} p(x_i, y_j) = \sum_{x_i} p(x_i|y_j)p(y_j) = \sum_{x_i} p(y_j|x_i)p(x_i) \tag{6}$$

The marginal probability distribution of x could be computed similarly, except that the summation would be over all possible values of y.

The influence diagram represents the joint probability distribution using the conditional form of Bayes' rule. For example, the joint probability distribution for the previously described two dimensional random vector is shown in the first influence diagram in Figure 1. In this diagram, an arrow points from node y to node x. The arrow implies that y is specified as the marginal distribution $p(y_j)$, and that x is expressed in terms of the conditional distribution $p(x_i|y_j)$. The second influence diagram in Figure 1 depicts the same joint probability distribution in terms of the marginal distribution of x and the conditional distribution of y given x.

If the joint probability distribution is specified as in the first influence diagram in Figure 1, then it can be converted to the second diagram by reversing the arrow. Mathematically, the correct distributions can be calculated with Bayes' rule. These operations are

$$p(x_i) = \sum_{y_j} p(x_i|y_j)p(y_j) \tag{7}$$

$$p(y_j|x_i) = \frac{p(x_i|y_j)p(y_j)}{p(x_i)} \tag{8}$$

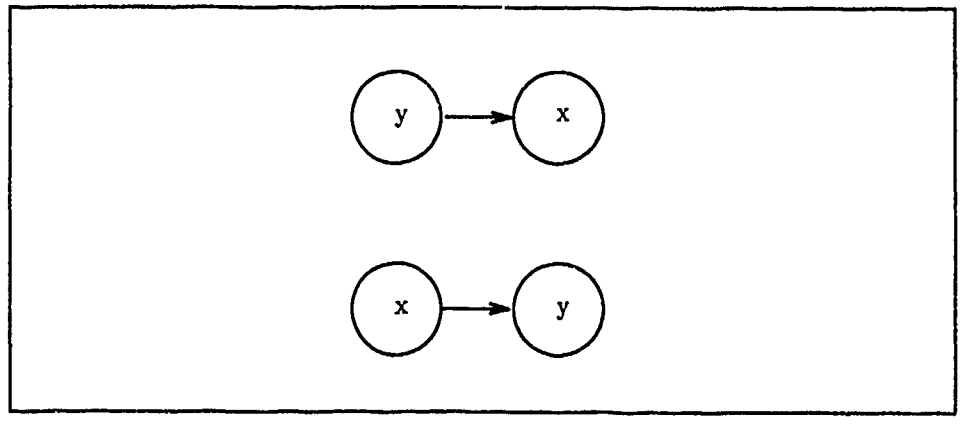


Figure 1. Influence Diagram for the Joint Distribution of $p(x_i, y_j)$

The mathematics in Equations (7) and (8) are not shown explicitly, but are implied by the diagram. This attribute makes the influence diagrams especially attractive for computation by digital computer. The influence diagram is the pictorial representation of the relationships between the variables. The digital computer accomplishes the mathematics needed for computing the underlying distribution functions.

An influence diagram with only two nodes (random variables) is a simplistic case. More commonly, there will be many random variables. Conditional distributions are associated with those nodes having an arrow pointing to them from other nodes. A conditional random variable may be conditioned on any number of the remaining variables in the diagram, with certain restrictions.

Taken together, all of the random variables in a diagram form a random vector. The conditional and unconditional distributions of these random variables represent the joint distribution for the random vector.

As an example, consider the joint distribution of three random variables with probabilities expressed in terms of ordered triples, $P[x = x_i, y = y_i, z = z_i] = p(x_i, y_j, z_k)$. This joint distribution can be expressed in terms of the marginal distribution of one variable, the conditional distribution of a second variable given the first, and the conditional distribution of the third, given the first two. The overall joint distribution remains unchanged, but it is expressed in a different form. This can be shown mathematically as $p(x_i, y_j, z_k) = p(x_i)p(y_j|x_i)p(z_k|x_i, y_j)$ and in influence diagram form in Figure 2.

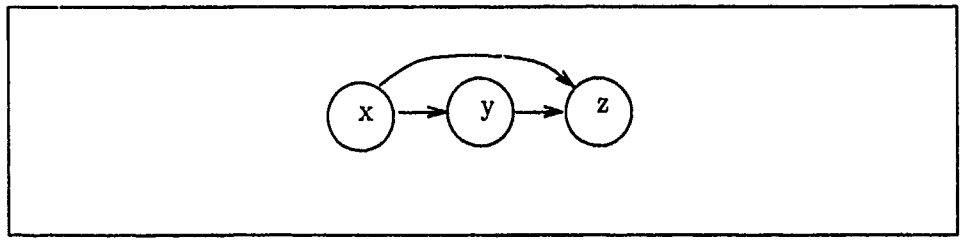


Figure 2. Influence Diagram for Three Variables

The following is some terminology relating to the influence diagram. The nodes which have arrows pointing to another node are called the conditional or direct predecessors of that node. A predecessor node may itself have a predecessor, and that predecessor may have one also. The set of all predecessors of a node, both direct and indirect, is the set of weak predecessors. Similarly, a node with a conditional distribution is called a successor node. A list of nodes is called 'ordered' if none of the weak predecessors of a node follow the node in the list. An influence diagram is defined to be an ordered list of nodes, corresponding to a unique joint distribution function. If a sequence of nodes cannot be ordered, then a cycle is said to exist.

The influence diagram in Figure 2 demonstrates the concept of an ordered list. The nodes can be ordered as x, y, and z. Node y follows node x in the list, and node y is not a predecessor

of x. Similarly, node z follows both nodes x and y, and z is not a predecessor of either. The influence diagram requirement for an ordered list is derived from the conditioning of Bayes' rule. Conceptually, if one node is conditioned on a predecessor node, then that predecessor node may not be conditioned on the successor, either directly or indirectly. To do so would imply circular conditioning.

Transformations can change the appearance of the influence diagram and affect the conditional distributions underlying the diagram. For the purpose of this thesis, these transformations are limited to the arc reversal and the elimination of 'barren' or 'nuisance' nodes. These operations will be explained by example.

2.2 Example of Discrete Random Variables

Assume three random variables described as follows. Random variables x, y, and z represent the probability of failure of components x, y, and z in a machine after a given amount of time. It is known that a failure of either component x or z induces a higher rate of failure in component y. It is possible to determine if components y and z have failed by direct measurement, but component x must be removed, a costly procedure. The problem is to calculate the probability of failure of component x, given the conditions of components y and z. Known probabilities of failure after a given period of time are listed in Table 1.

The influence diagram in Figure 3 depicts this three-variable example. The random variables x and z are expressed as marginal densities. They are not conditioned on any other random variables. The random variable y is conditioned on both x and z. This means that y is conditioned on the random vector (x,z). This diagram also shows x and z to be independent by the lack of an arc between them. The influence diagram does not show the actual distributions of the random variables. It is simply a pictorial method of bookkeeping; the influence diagram shows the conditioning order of the random variables. Both the influence diagram and the tabular data

associated with it are needed to express the overall joint distribution.

Table 1. Initial Distributions for Discrete Influence Diagram Example

$P(z=0)=0.25$	$P(z=1)=0.75$
$P(x=0)=0.6$	$P(x=1)=0.4$
$P(y=0\|z=0,x=0)=0.5$	$P(y=1\|z=0,x=0)=0.5$
$P(y=0\|z=0,x=1)=0.25$	$P(y=1\|z=0,x=1)=0.75$
$P(y=0\|z=1,x=0)=0.2$	$P(y=1\|z=1,x=0)=0.8$
$P(y=0\|z=1,x=1)=0.1$	$P(y=1\|z=1,x=1)=0.9$

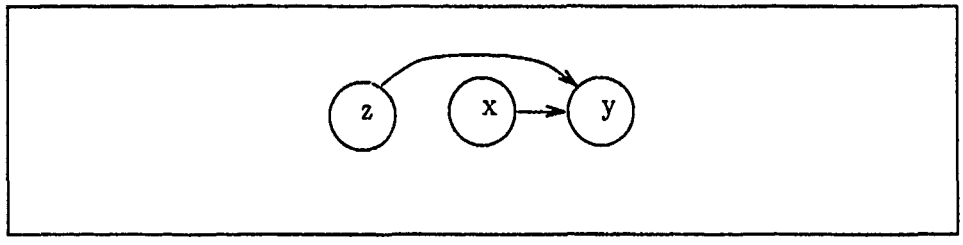


Figure 3. Influence Diagram for Discrete Random Variable Example

The influence diagram arrows in Figure 3 follow the direction of causality. In general, the arrows do not imply causality. In this case however, the causal factors of the process model determine the initial probabilistic relationships. After some mathematical calculations, the probabilistic relationships will be expressed in a different form, and the arrows in the influence diagram will not have any relationship to causality.

The desired distribution is the probability that component x has failed, given the conditions of components y and z. This distribution is given in Table 2 and the influence diagram in Figure 4. The probability distribution of y given z is calculated by taking the summation of y given x and z over all values of x. The probability of x given y and z is calculated by using Bayes' rule. The specific equations are:

$$p(y_j|z_k) = \sum_{x_i} p(y_j|z_k, x_i)p(x_i|z_k) \tag{9}$$

$$p(x_i|y_j, z_k) = \frac{p(y_j|x_i, z_k)p(x_i|z_k)}{p(y_j|z_k)} \tag{10}$$

But because x and z are independent, $p(x_i|z_k) = p(x_i)$. The resultant transformation is an example of arc reversal. The result is the probability of the component x having failed, given the condition of the two observed components. The underlying joint distribution is unchanged.

Table 2. Conditional Distribution of x

$P(z = 0) = 0.25$	$P(z = 1) = 0.75$
$P(y = 0\|z = 0) = 0.4$	$P(y = 1\|z = 0) = 0.6$
$P(y = 0\|z = 1) = 0.16$	$P(y = 1\|z = 1) = 0.84$
$P(x = 0\|z = 0, y = 0) = 0.75$	$P(x = 1\|z = 0, y = 0) = 0.25$
$P(x = 0\|z = 0, y = 1) = 0.5$	$P(x = 1\|z = 0, y = 1) = 0.5$
$P(x = 0\|z = 1, y = 0) = 0.75$	$P(x = 1\|z = 1, y = 0) = 0.25$
$P(x = 0\|z = 1, y = 1) = 0.571429$	$P(x = 1\|z = 1, y = 1) = 0.428571$

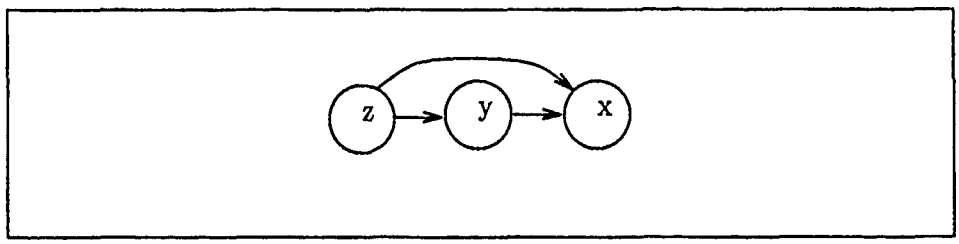


Figure 4. Influence Diagram for Conditional Distribution of x

An important characteristic of arc reversal is that the both nodes involved in the reversal inherit each others direct predecessors. This is a function of Bayes' rule as seen from this example. In this case, node x inherits the predecessors of node y, namely node z. Similarly, if another node had been a predecessor of node x, then it would have become a predecessor of node y after the reversal. This can be seen in the above equations by placing another conditioning variable on x. This additional conditioning variable would have also shown up as a conditioning variable of y in Equation (9).

There is one more point to be made about the resulting diagram in Figure 4. It is not now possible to reverse the arrow from node z to node x. This would result in a cycle in the influence diagram, and the diagram could no longer be ordered. In general, two nodes can be reversed only if they can be placed next to each other in an ordered sequence of nodes. If node z were to be moved

to the end of the ordered sequence, then it must be done without causing a cycle. For the example in Figure 4, nodes y and x would be reversed, resulting in the original diagram in Figure 3. Next, node x and z would be reversed. Since these random variables are independent, no mathematical operations are required for their reversal. Finally, nodes y and z could be reversed to form the diagram in Figure 5 and the distribution in Table 3.

Table 3. Conditional Distribution of z

$P(x = 0) = 0.6$	$P(x = 1) = 0.4$
$P(y = 0\|x = 0) = 0.275$	$P(y = 1\|x = 0) = 0.725$
$P(y = 0\|x = 1) = 0.1375$	$P(y = 1\|x = 1) = 0.8625$
$P(z = 0\|x = 0, y = 0) = 0.454545$	$P(z = 1\|x = 0, y = 0) = 0.545454$
$P(z = 0\|x = 0, y = 1) = 0.172414$	$P(z = 1\|x = 0, y = 1) = 0.827586$
$P(z = 0\|x = 1, y = 0) = 0.454545$	$P(z = 1\|x = 1, y = 0) = 0.545454$
$P(z = 0\|x = 1, y = 1) = 0.217391$	$P(z = 1\|x = 1, y = 1) = 0.782609$

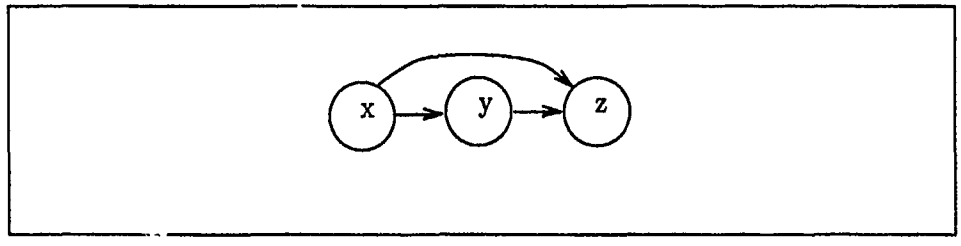


Figure 5. Influence Diagram for Discrete Random Variable Example

A slight change in this second example illustrates another of the transformations, the elimination of barren nodes. Assume this time that the components z and y cannot be tested directly. A decision to replace component y is based solely on the condition of component x. The desired distribution is the probability of y given x. The condition of component z is irrelevant to the decision because it cannot be observed.

The desired distribution is already present in the top part of Table 3. The influence diagram is shown in Figure 6. In this case, node z was barren, meaning that it was not a predecessor of the desired nodes. The probabilistic information in node z was already incorporated into node y by taking the summation as in $p(y_j) = \sum_{z_k} p(y_j|z_k)p(z_k)$. Node z is no longer needed and

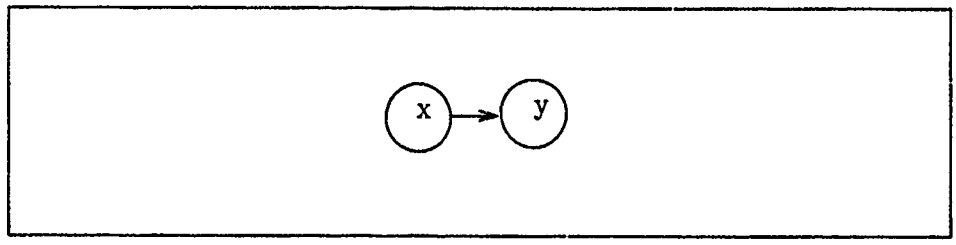


Figure 6. Discrete Distribution Influence Diagram with Node Removed

can be removed. However, the remaining influence diagram does not represent the original joint distribution. It represents the joint distribution of the subset of the original random variables, corresponding to the visible nodes.

To summarize in simple terms, making a node less conditional involves taking the summation of that random variable's conditional distribution over one of its direct predecessors. In the process, the original successor node inherits all conditional predecessors of the original predecessor. On the other hand, making a node more conditional requires an application of Bayes' rule. Specifically, it needs the summation just calculated as the denominator term for Bayes' rule. The new successor will inherit all conditional predecessors of the original successor. If the new successor node is barren, and if it has no meaning in desired result, then it can be removed. In this case, Bayes' rule is not needed because the new distribution of the successor node will be discarded when the node is removed.

A deterministic variable is one that is a function of only the outcome of other variables. The functional relationship between one variable and another is depicted as the conditioning of a deterministic variable upon those variables of which it is a function. For this reason, in the influence diagram, the deterministic variable is usually shown as a conditional variable. If the conditioning variables for a deterministic variable are random, then the deterministic variable can be expressed as a random variable in unconditional form. The distribution can be calculated by taking summations as before.

2.3 Continuous Random Variables

The influence diagram can be extended to the case of continuous random variables. If the random variables are Gaussian, and all joint densities of the variables are Gaussian, then the variables are called jointly Gaussian. Gaussian random vectors are easy to represent because the entire probability density function can be specified with only a vector of the means and a covariance matrix. It is also known that any linear combination of jointly Gaussian random variables is Gaussian. Furthermore, any subset of jointly Gaussian variables is jointly Gaussian itself.

The Gaussian influence diagram is patterned after the discrete influence diagram described earlier. It consists of a set of jointly Gaussian random variables. Each node represents one of the random variables and the entire influence diagram represents the joint density function of a Gaussian random vector. The rules for exchanging nodes and for node removal are the same as for the discrete form of the influence diagram. Again, the influence diagram represents the conditioning relationships while the actual probability density functions are computed separately.

Another example will show the simple two-variable case. Assume two jointly Gaussian random variables x and y make up a random vector. As stated before, the joint density function of the random variables can be specified by a two-dimensional vector of means and a two-by-two covariance matrix. The influence diagram expresses the joint density as the mean and variance of one variable, along with the conditional mean and conditional variance of the second, given the first. This conditional density is Gaussian because any conditional density of jointly Gaussian random variables is Gaussian also. Either representation is sufficient to specify the overall joint density function.

The two Gaussian random variables are x and y. The mean vector and covariance matrix are:

$$\mathbf{m} = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \tag{11}$$

$$\mathbf{P} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} \tag{12}$$

where μ_x and μ_y are the unconditional means of x and y respectively. Also, the notation Σ_{xy} implies the covariance of the variables x and y and is equal to Σ_{yx}. Finally, Σ_{xx} represents the covariance of variable x with itself, or just the variance of x.

The conditional mean of x given y becomes $\mu_{x|y} = \mu_x + \frac{\Sigma_{xy}}{\Sigma_{yy}}(\rho - \mu_y)$, where ρ is a dummy variable representing the possible realizations of the variable y. The quantity in parenthesis is the difference of the predicted (mean) value of the random variable and its realization. This term will be called the residual. The conditional variance of x given y is $\Sigma_{x|y} = \Sigma_{xx} - \frac{\Sigma_{xy}^2}{\Sigma_{yy}}$ [7:pp. 110–111].

The coefficient $\frac{\Sigma_{xy}}{\Sigma_{yy}} = \beta_{xy}$ has special significance. It is called the regression coefficient of x on y. It represents a linear change in the conditional mean of x, given the realization of y. The notation β_{xy} comes from Yule [17]. The order of the subscripts is significant for the regression coefficient so $\beta_{xy} \neq \beta_{yx}$.

The notation in the remainder of this thesis will follow the convention of Kenley rather than Yule. Kenley uses the notation b_{yx} instead of β_{xy}. Also, Kenley does not explicitly write any other conditioning variables in the coefficient as Yule does. Instead, other conditioning variables are implied by the influence diagram [6].

The regression coefficient is related to, but is not the same as the correlation coefficient. The correlation coefficient is given by $r_{xy} = r_{yx} = \frac{\Sigma_{xy}}{\sqrt{\Sigma_{xx}}\sqrt{\Sigma_{yy}}}$. The regression coefficient can be calculated from the correlation coefficient by the relationship $b_{yx} = r_{xy}\frac{\sqrt{\Sigma_{xx}}}{\sqrt{\Sigma_{yy}}}$.

Either random variable could have been chosen to be represented in unconditional form. If the order of conditioning were reversed, then the conditional mean of y given x would be $\mu_{y|x} = \mu_y + \frac{\Sigma_{yx}}{\Sigma_{xx}}(\xi - \mu_x)$, where ξ is now the dummy variable representing the possible realizations of the variable x. The conditional variance of y given x is $\Sigma_{y|x} = \Sigma_{yy} - \frac{\Sigma_{yx}^2}{\Sigma_{xx}}$. The term

$\frac{\Sigma_{yx}}{\Sigma_{xx}} = b_{xy}$ is called the regression coefficient of y on x. It too is related to the correlation coefficient r_{xy}, this time by the relationship $b_{xy} = r_{xy}\frac{\sqrt{\Sigma_{yy}}}{\sqrt{\Sigma_{xx}}}$.

The influence diagram represents the joint density function as the marginal density of one variable and the conditional density of the other. Assume an influence diagram and its underlying density as shown in Figure 7. In this case, the random variable y is given in unconditional form, and the random variable x is expressed in conditional form. Only five values are needed to represent the joint distribution: the unconditional means of the two random variables, the unconditional variance of y, the conditional variance of x, and the regression coefficient of x on y. In this figure, v_i is the variance of the ith node, whether it is in conditional or unconditional form. Because of the simple method for describing the joint density, these values are shown directly on the influence diagram rather than in a separate table. The means and variances are associated with the appropriate node, while the regression coefficients are associated with the arrow between the nodes.

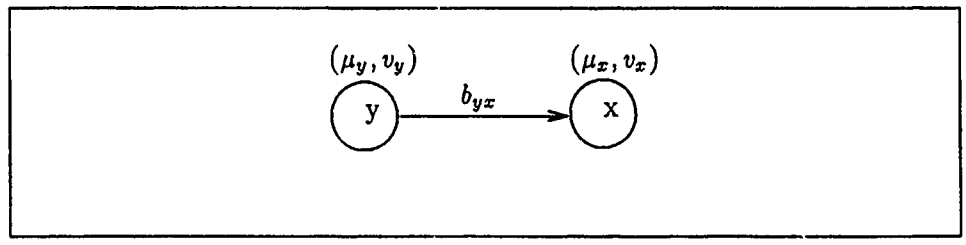


Figure 7. Influence Diagram for Continuous Gaussian Random Variables

As with the discrete form of the influence diagram, it is possible to reverse the arrow between the two nodes. This is equivalent to changing the order of conditioning using Bayes' rule. The mathematics for the Gaussian form of the influence diagram are different however.

If the unconditional density of x is desired, then only the unconditional variance of x must be computed (the unconditional mean already exists). The original calculations for the conditional variance can be used to give $\Sigma_{x|y} = \Sigma_{xx} - \frac{\Sigma_{xy}^2}{\Sigma_{yy}} = \Sigma_{xx} - b_{yx}^2\Sigma_{yy}$. This equation can be rearranged to give

$$\Sigma_{xx} = \Sigma_{x|y} + b_{yx}^2\Sigma_{yy} \tag{13}$$

The second part of the reversal of these two nodes is the calculation of the conditional density of y. The only values that need to be calculated are the conditional variance and the regression coefficient of y on x. The conditional variance of y given x was already shown to be $\Sigma_{y|x} = \Sigma_{yy} - \frac{\Sigma_{yx}^2}{\Sigma_{xx}}$. This equation can be simplified as follows:

$$\begin{aligned}
\Sigma_{y|x} &= \Sigma_{yy} - \frac{\Sigma_{yx}^2}{\Sigma_{xx}} \\
&= \frac{\Sigma_{yy}\Sigma_{xx} - \Sigma_{xy}^2}{\Sigma_{xx}} \\
&= \frac{\Sigma_{yy}(\Sigma_{xx} - \frac{\Sigma_{xy}^2}{\Sigma_{yy}})}{\Sigma_{xx}} \\
\Sigma_{y|x} &= \frac{\Sigma_{yy}\Sigma_{x|y}}{\Sigma_{xx}} \qquad (14)
\end{aligned}$$

Similarly, the regression coefficient of y on x is calculated by

$$b_{xy} = \frac{\Sigma_{xy}}{\Sigma_{xx}} = \frac{b_{yx}\Sigma_{yy}}{\Sigma_{xx}} \qquad (15)$$

The result is that the joint density function is expressed as an unconditional density of x and a conditional density of y. This is shown in the influence diagram in Figure 8.

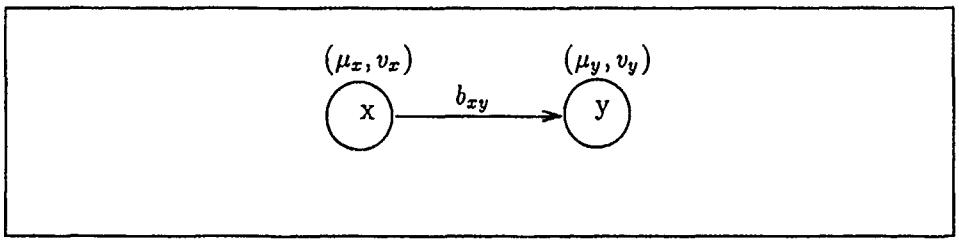


Figure 8. Continuous Gaussian Influence Diagram after Reversal

As in the discrete form of the influence diagram, the two-variable case is simplistic. The calculations become slightly more complicated as the number of variables increases. If there are n random variables in the Gaussian random vector, then one is chosen, perhaps arbitrarily, to be represented in unconditional form. A second random variable is conditioned on the first as in

the previous example. The third random variable is represented as being conditioned on the first two. There will be two regression coefficients associated with this third variable, one from the first variable and one from the second. The process continues with the fourth variable conditioned on the previous three and having three regression coefficients, etc. Eventually, the nth random variable will be represented as a conditional density, conditioned on the previous $n-1$ variables, with a regression coefficient associated with each. The result can be written in the form of an n-dimensional influence diagram. Using simplified notation where f_{x_n} implies $f_{x_n}(\xi_n)$, the diagram shown in Figure 9 can be expressed mathematically as:

$$f_{x_1,x_2,x_3\ldots x_n} = f_{x_1} f_{x_2|x_1} f_{x_3|x_1,x_2} \cdots f_{x_n|x_1,x_2,x_3\ldots x_{n-1}} \tag{16}$$

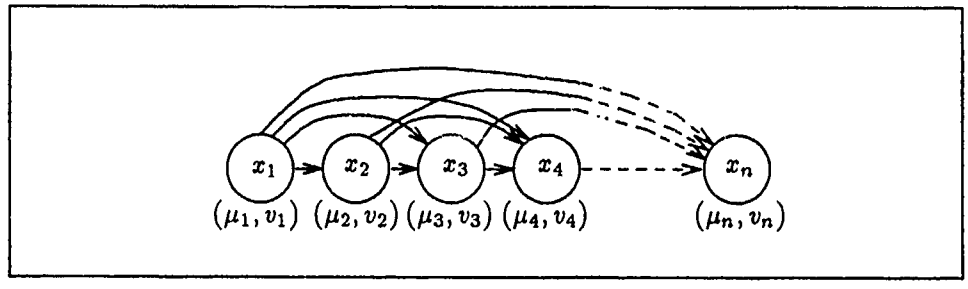


Figure 9. Influence Diagram for N Jointly Gaussian Random Variables

Assume two variables are represented by two nodes in an influence diagram. If the nodes can be placed next to each other in some ordered sequence, then the conditioning order can be reversed. Call the first variable x_i and the second x_j where it is assumed that x_i is a conditional predecessor of x_j. Furthermore, assume that x_K is a set containing the union of all direct predecessor nodes of both x_i and x_j, except for i itself. Also assume that x_k is an arbitrary element of x_K. Nodes x_i and x_j have variance v_i and v_j respectively. The term b_{ij} is the regression coefficient of j on i; the terms b_{ki} and b_{kj} are the regression coefficients of x_i and x_j on x_k.

In the multivariable case, the equations for calculating the new variances and regression coefficients are similar to those in the preceding paragraphs for the two-variable case. Additionally, both nodes will inherit each others direct predecessors, and the regression coefficients from predecessor nodes must be adjusted to reflect the new conditioning order. The equations for calculating the new variances and regression coefficients are as follows, where the prime symbols represent a new value.

For node j, no longer conditioned on node i, the new variance and regression coefficients from predecessors are:

$$v'_j = v_j + b_{ij}^2 v_i \tag{17}$$

$$b'_{kj} = b_{kj} + b_{ki} b_{ij} \tag{18}$$

For node i, conditioned on node j, the new variance and regression coefficients are:

$$v'_i = \frac{v_i v_j}{v'_j} \tag{19}$$

$$b'_{ji} = \frac{v_i b_{ij}}{v'_j} \tag{20}$$

$$b'_{ki} = b_{ki} + b'_{kj} b'_{ji} \tag{21}$$

2.4 Example of the Gaussian Influence Diagram

The following example does not use actual statistics and does not represent any actual relationships. It is purely fictional to demonstrate the use of the influence diagram.

Suppose that an analysis of statistics from a group of professional basketball players reveals a relationship between the height of the player, the percentage of playing time in a game, and the number of points per game. These three statistics are assumed to be jointly Gaussian with a mean

vector and covariance matrix:

$$\mathbf{m} = \begin{bmatrix} \mu_h \\ \mu_p \\ \mu_t \end{bmatrix} = \begin{bmatrix} 82 \\ 20 \\ 75 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \Sigma_{hh} & \Sigma_{hp} & \Sigma_{ht} \\ \Sigma_{ph} & \Sigma_{pp} & \Sigma_{pt} \\ \Sigma_{th} & \Sigma_{tp} & \Sigma_{tt} \end{bmatrix} = \begin{bmatrix} 9 & 2 & 4 \\ 2 & 9 & 15 \\ 4 & 15 & 49 \end{bmatrix}$$

In this example, the subscript h refers to the height in inches. The average (mean) height is 82 inches, with a standard deviation of 3 inches. The subscript p is the number of points scored per game. The average number of points is 20, with a standard deviation of 3 points. The subscript t is the playing time expressed as a percentage of total game time. The average is 75% with a standard deviation of 7%.

These three variables can be expressed in an influence diagram as shown in Figure 10. The node labeled H refers to the player's height, the node labeled P is the average number points per game, and the node labeled T is the playing time as a percentage. The operations for calculating the influence diagram from a covariance matrix will be discussed briefly in the next section. The most notable feature of this diagram is the relationship between the three variables.

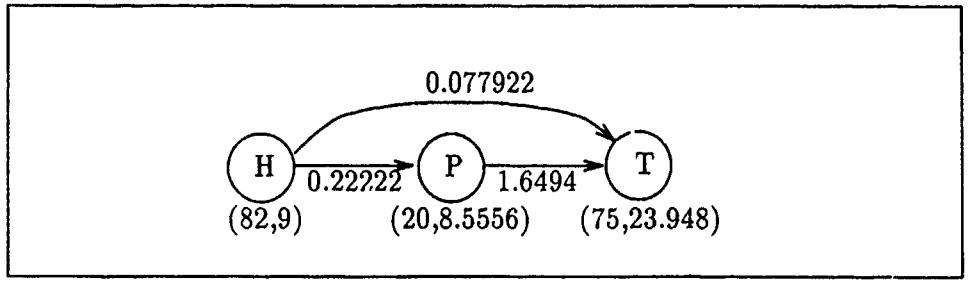


Figure 10. Example of Continuous Gaussian Influence Diagram

The regression coefficient for the number points against the height is approximately 0.22222. This means that each increase in heignt of one inch changes the conditional mean by 0.22222 points

per game. Similarly, an increase of one inch in height changes the conditional mean of the playing time by 0.077922%. An increase of 1 point per game implies an increase in percentage of playing time of 1.6494%. If both the height and the number of points are given, the change in the mean playing time is the sum of the changes caused by the measured height and number of points scored. The effect can be thought of as propagating the residual to all subsequent nodes.

The conditional mean and variance, as specified in this diagram, can be used to infer the average playing time if a player's height is given or if the a player's points per game are given. The total effect of the regression coefficients from the first node to the last one is computed by summing the effects from both paths, both direct and indirect through the second node. The direct effect is the regression coefficient on the path from the first to the third. The indirect effect is the product of the coefficients on the path from the first to the second, and from the second to the third. The change in the mean of the last node due to the realization of the first node is $(0.22222)(1.6494) + (0.077922) = 0.44445$.

As an example, assume an 84-inch player scores 16 points in a game. The conditional mean of the number of points, given the player's height is $20 + 0.22222(84 - 82) = 20.4444$. The conditional mean of the time played, given both the height and the number of points is calculated by first finding the effect due to the player's height, then finding the effect due to the number of points scored. The effect due to the height measurement is $75 + [0.44445(84 - 82)] = 75.8889\%$. At this time, the height node is no longer needed on the influence diagram. The effects of the measurement have been propagated to all subsequent nodes. In the influence diagram, the realization and removal of a random variable is called "instantiation." Instantiated nodes must be at the beginning of the ordered sequence of nodes so that all remaining nodes can be affected by the realization of the random variable.

The second calculation uses the new conditional mean for the number of points scored to calculate the average playing time. The calculations for the average playing time now becomes

$75.8889 + [1.6494(16 - 20.44444)] = 68.558\%$. Again, the node corresponding to the realized random variables can be instantiated. It was at the beginning of the ordered sequence because its predecessor was removed by instantiation. After the second node is instantiated, only the last node remains. The result is the conditional mean estimate of the playing time for this player of 68.558% with a (conditional) standard deviation of 4.8937%. Even though this node has no predecessors, it retains the conditional mean and variance based on instantiated variables. For the remainder of this thesis, a node with no predecessors will be said to be in "unconditional form" even though it may represent a conditional distribution.

The influence diagram can also be rearranged as described in the previous section. In this case, it will be rearranged so that the player's height is last. This takes place in two separate node reversal operations. First, the nodes labeled H and P are reversed to yield the first influence diagram in Figure 11. The calculation involved in calculating the new conditional and unconditional variances are:

$$\begin{aligned} v_p &= 8.5556 + (0.22222)^2(9) \\ &= 9 \\ v_h &= \frac{(9)(8.5556)}{9} \\ &= 8.5556 \\ b_{ph} &= \frac{(9)(0.22222)}{9} \\ &= 0.22222 \end{aligned}$$

Second, the nodes labeled H and T are reversed to yield the influence diagram in Figure 12. Both nodes have node P as a predecessor so the calculations are:

$$
\begin{aligned}
v_t &= 23.948 + (1.6494)^2(8.5556) \\
&= 47.224 \\
b_{pt} &= 0.077922 + (0.22222)(1.6494) \\
&= 0.44445 \\
v_h &= \frac{(8.5556)(23.948)}{47.224} \\
&= 4.3387 \\
b_{th} &= \frac{(8.5556)(1.6494)}{47.224} \\
&= 0.29882 \\
b_{ph} &= 0.22222 - (0.44445)(0.29882) \\
&= 0.089409
\end{aligned}
$$

Again, the conditional mean and variance can be calculated, based on realizations of either or both of the first two random variables in the diagram. In another example, assume a player scores 24 points in a game and plays 95% of the game. The conditional mean of the playing time, given the number of points scored is $75 + 0.44445(24 - 20) = 76.7778\%$. The conditional mean of the

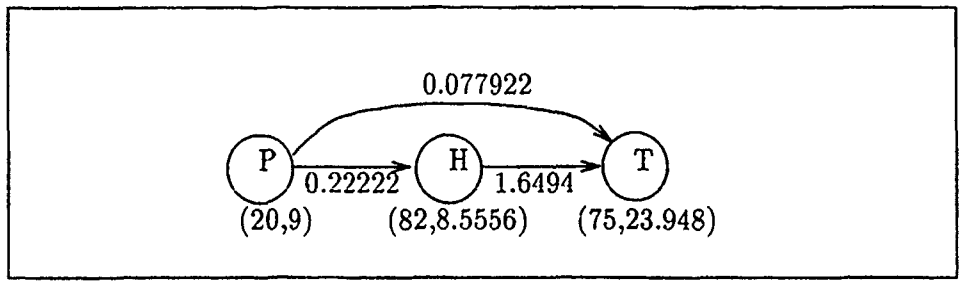


Figure 11. First Reversal of Gaussian Influence Diagram Example

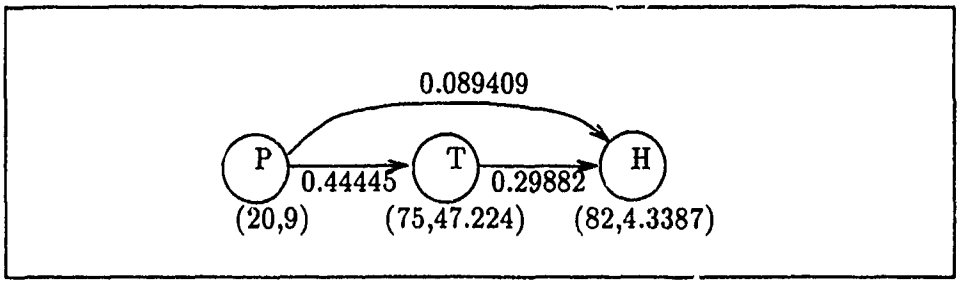


Figure 12. Second Reversal of Gaussian Influence Diagram Example

player's height, given both the number of points scored and the actual playing time is calculated in two steps. First, the total change due to the change in number of points is computed. The total coefficient from the first to the last node is $0.089409 + (0.44445)(0.29882) = 0.22222$. The new average height, due to the realization of the number of points is $82 + 0.22222(24 - 20) = 82.8889$. The first node is then instantiated, leaving only the last two nodes. Next, the change due to the playing time is computed. This is $82.8889 + (0.29882)(95 - 76.7778) = 88.334$. The second node is now instantiated, leaving only the last node. The conditional mean estimate of the player's height is 88.334 inches with a (conditional) standard deviation of 3.0830 inches.

2.5 Matrix Representation of the Influence Diagram

The influence diagram is a convenient tool for calculating the conditional mean and variance of an element of a Gaussian random vector. It is a visual depiction of the equations for the Gaussian conditional mean vector and covariance matrix:

$$\mathbf{P}_{x|y} = \mathbf{P}_{xx} - \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx} \tag{22}$$

$$\mathbf{m}_{x|y} = \mathbf{m}_x + \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}(\rho - \mathbf{m}_y) \tag{23}$$

where ρ is a dummy vector for the realization of components of the $\mathbf{y}$, $\mathbf{P}_{x|y}$ is the conditional variance of $\mathbf{x}$ given $\mathbf{y}$, and $\mathbf{m}_{x|y}$ is the conditional mean of $\mathbf{x}$ given $\mathbf{y}$ [7:pp. 110–111].

In the first of these two equations, the conditional variance of a node in the influence diagram is fixed for a given set of conditioning variables. The influence diagram allows convenient recalculation of the conditional variance when the conditioning variables change.

In the second of these two equations, the conditional mean is seen to be a linear function of the residuals of the conditioning variables. This linear relationship is maintained in the influence diagram by the regression coefficients.

The covariance matrix can be used to calculate both the matrix of regression coefficients and the conditional variances for all random variables in the vector. The calculations are closely related to the Cholesky decomposition of the covariance matrix. For example, let $\mathbf{P}$ be a positive definite, symmetric covariance matrix. A factorization of $\mathbf{P}$ exists in the form $\mathbf{P} = \mathbf{U}^T\mathbf{SSU}$ where $\mathbf{U}$ is a unit upper triangular matrix and $\mathbf{S}$ is a diagonal matrix so that $\mathbf{S}^T = \mathbf{S}$. The matrix $(\mathbf{SU})^T\mathbf{SU}$ is identical to the Cholesky decomposition of the covariance matrix. The matrix $\mathbf{SS} = \mathbf{D}$ is also a diagonal matrix so the the covariance matrix can be represented as $\mathbf{U}^T\mathbf{DU}$.

Let $\mathbf{I}$ be the identity matrix, and $\mathbf{B}_j$ be a strictly upper triangular matrix that is all zeros except for the jth column above the diagonal. The elements of the $\mathbf{B}_j$ matrix are the regression coefficients $b_{1j}, b_{2j}, \ldots, b_{j-1,j}$ where the subscripts indicate the predecessor and the successor node. The subscripts also correspond to the conventional row-column notation for matrix elements. The matrix $\mathbf{U}_j$ is a unit upper triangular matrix defined as $(\mathbf{I} - \mathbf{B}_j)$, and the matrix $\mathbf{B}$ is defined as $\mathbf{B}_1 + \mathbf{B}_2 + \mathbf{B}_3 + \ldots + \mathbf{B}_n$ (where the convention leads to $\mathbf{B}_1 = \mathbf{0}$ and $\mathbf{U}_1 = \mathbf{I}$). Kenley and Schacter showed [12:pp. 547–548]:

$$\mathbf{U} = \mathbf{U}_1\mathbf{U}_2\mathbf{U}_3 \ldots \mathbf{U}_n \tag{24}$$

and

$$\mathbf{U} = (\mathbf{I} - \mathbf{B})^{-1} \tag{25}$$

The $\mathbf{D}$ and the $\mathbf{U}$ matrices can be computed by taking the Cholesky decomposition of the covariance

matrix, then factoring the $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \ldots, \mathbf{U}_n$ matrices from the U matrix. A more efficient method is given in Kenley and Schacter [12:549].

For the example in the previous section, the matrices turn out to be:

$$\mathbf{P} = \begin{bmatrix} 9 & 2 & 4 \\ 2 & 9 & 15 \\ 4 & 15 & 49 \end{bmatrix}$$

$$\mathbf{B}_2 = \begin{bmatrix} 0 & 0.22222 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}_3 = \begin{bmatrix} 0 & 0 & 0.077922 \\ 0 & 0 & 1.6494 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{U} = \mathbf{U}_2\mathbf{U}_3 = \begin{bmatrix} 1 & 0.22222 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0.077922 \\ 0 & 1 & 1.6494 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.22222 & 0.44445 \\ 0 & 1 & 1.6494 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 8.5556 & 0 \\ 0 & 0 & 23.948 \end{bmatrix}$$

From this example, each term of the diagonal matrix **D** is seen to be the conditional variance of the random variable, conditioned on those variables on the diagonal above it. The terms above the diagonal in the $\mathbf{B}_j$ and $\mathbf{U}_j$ matrices are just the regression coefficients. The b_{ij} coefficient indicates the direct linear change in the jth conditional mean due to a change in the mean of the

ith variable. The terms of the U matrix represent the total effect of all regression coefficients, both direct and indirect. In this example, the U_{13} term is 0.44445. This is the value previously calculated as the effect of the measured height on the conditional mean of playing time.

2.6 The Influence Diagram for Discrete-Time Filtering

Kenley proved in his doctoral dissertation that the influence diagram could be used for discrete-time filtering. The first of the following subsections will be a discussion of discrete-time Kalman filtering taken from Maybeck [7:pp. 133–220] and discrete-time filtering taken from Kenley [6:pp. 52–106]. It is intended to present Kenley's theories in parallel with more widely known Kalman filtering algorithms. The second of the following subsections will be a numerical example showing the operation of the influence diagram.

The Kalman filter is a conditional mean estimator of the states of a linear system, under the assumption of linear measurements and Gaussian disturbances. Under these assumptions, it is an optimal estimator.

The Gaussian influence diagram permits the calculation of the conditional density of Gaussian random variables in a jointly Gaussian random vector. If these variables represent the states of a linear system, then it too is a conditional mean estimator for the states. Under these assumptions, it will yield the same state estimates and variances as the Kalman filter.

Assume a linear system of the form:

$$\mathbf{x}(t_i) = \Phi(t_i, t_{i-1})\mathbf{x}(t_{i-1}) + \mathbf{G}_d(t_{i-1})\mathbf{w}_d(t_{i-1}) \tag{26}$$

where $\mathbf{x}(t_i)$ and $\mathbf{x}(t_{i-1})$ are n-dimensional state vectors at times t_i and t_{i-1} respectively. $\Phi(t_i, t_{i-1})$ is the state transition matrix which describes the propagation of the states from time t_{i-1} to time t_i. The term $\mathbf{w}_d(t_{i-1})$ represents the process noise and is assumed to be a discrete-time zero-mean

white Gaussian noise sequence with covariance kernel

$$E\{\mathbf{w}_d(t_i)\mathbf{w}_d^T(t_j)\} = \begin{cases} \mathbf{Q}_d(t_i), & t_i = t_j, \\ 0, & t_i \neq t_j \end{cases} \tag{27}$$

The matrix $\mathbf{G}_d(t_{i-1})$ represents a linear operation which describes how the discrete-time noise enters the system at time t_{i-1}. The vectors $\mathbf{x}(t_{i-1})$ and $\mathbf{w}_d(t_{i-1})$ are assumed to be independent.

Now assume that the state vector at time t_{i-1} is a Gaussian random vector. The conditional mean of this vector, based on a priori information and measurements up to time t_{i-1} is defined to be:

$$\hat{\mathbf{x}}(t_{i-1}^+) \equiv E\{\mathbf{x}(t_{i-1})|\mathbf{Z}(t_{i-1})\} \tag{28}$$

where:

$$\mathbf{Z}(t_{i-1}) = \begin{bmatrix} \mathbf{z}(t_1) \\ \mathbf{z}(t_2) \\ . \\ . \\ \mathbf{z}(t_{i-1}) \end{bmatrix} \tag{29}$$

In this equation, the measurement vector available at time t_j is called $\mathbf{z}(t_j)$, and $\mathbf{Z}(t_{i-1})$ is the history of measurements through time t_{i-1}. At t_0, the state estimate is defined to be:

$$\hat{\mathbf{x}}(t_0) \equiv E\{\mathbf{x}(t_0)\} \tag{30}$$

The conditional covariance of the state estimate at time t_{i-1} is defined by:

$$\mathbf{P}(t_{i-1}^+) \equiv E\{\left[\mathbf{x}(t_{i-1}) - \hat{\mathbf{x}}(t_{i-1}^+)\right]\left[\mathbf{x}(t_{i-1}) - \hat{\mathbf{x}}(t_{i-1}^+)\right]^T |\mathbf{Z}(t_{i-1})\} \tag{31}$$

and the unconditional covariance of the estimate at time t_0 is

$$\mathbf{P}(t_0) \equiv E\{[\mathbf{x}(t_0) - \hat{\mathbf{x}}(t_0)][\mathbf{x}(t_0) - \hat{\mathbf{x}}(t_0)]^T\} \tag{32}$$

Because the state estimate at time t_{i-1} is a Gaussian random vector, its density can be fully described by the conditional mean $\hat{\mathbf{x}}(t_{i-1}^+)$ and the covariance matrix $\mathbf{P}(t_{i-1}^+)$.

The state estimate at time t_{i-1} can also be described in influence diagram form. The n-dimensional random vector is depicted as n random variable nodes. The means of the nodes are the respective conditional means of the vector $\hat{\mathbf{x}}(t_{i-1}^+)$. The regression coefficients and the conditional variances of the individual nodes are equivalent to the covariance matrix $\mathbf{P}(t_{i-1}^+)$. Such an n-dimensional influence diagram is shown in Figure 13. In this influence diagram, the first node is in unconditional form, but it represents the conditional mean and variance, conditioned on prior measurements. The prior measurements do not need to be shown because they represent random variables that have been realized and instantiated. The remaining nodes represent the conditional mean and variance, based on the instantiated nodes. The realizations of the instantiated nodes do not affect the conditional variances, but they do affect the conditional means. In this way, the conditional mean vector $\hat{\mathbf{x}}(t_{i-1}^+)$ becomes a sufficient statistic for the measurement history $\mathbf{Z}(t_{i-1})$.

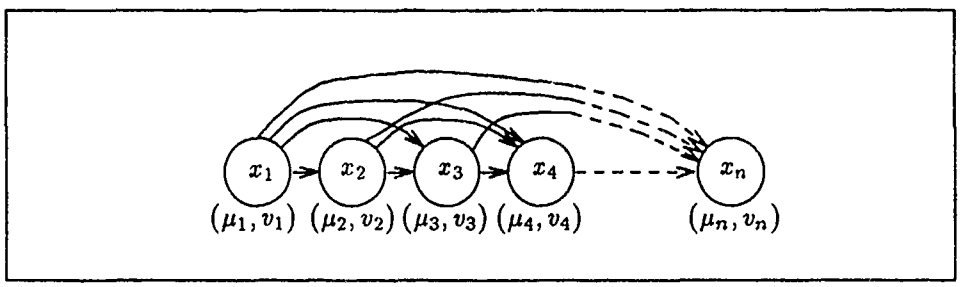


Figure 13. Influence Diagram for N Jointly Gaussian Random Variables

The linear model in Equation (26) describes the propagation of the states from time t_{i-1} to time t_i. The estimate at time t_i is also Gaussian with a conditional density function of the states

at time t_i, given the measurements through time t_{i-1} defined by the conditional mean

$$\hat{\mathbf{x}}(t_i^-) \equiv E\{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})\} \tag{33}$$

and the conditional covariance matrix

$$\mathbf{P}(t_i^-) \equiv E\{\left[\mathbf{x}(t_i) - \hat{\mathbf{x}}(t_i^-)\right]\left[\mathbf{x}(t_i) - \hat{\mathbf{x}}(t_i^-)\right]^T |\mathbf{Z}(t_{i-1})\} \tag{34}$$

The conditional mean and covariance at time t_i can be calculated by the Kalman filter propagation equations:

$$\hat{\mathbf{x}}(t_i^-) = \Phi(t_i, t_{i-1})\hat{\mathbf{x}}(t_{i-1}^+) \tag{35}$$

$$\mathbf{P}(t_i^-) = \Phi(t_i, t_{i-1})\mathbf{P}(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + \mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1}) \tag{36}$$

The influence diagram can depict the linear model of Equation (26). As shown earlier, the conditional density of the state estimate at time t_{i-1} corresponds to an influence diagram with n nodes. The conditional means of the nodes are $\hat{\mathbf{x}}(t_{i-1}^+)$ while the conditional covariance $\mathbf{P}(t_{i-1}^+)$ is factored into the conditional variances of the respective nodes and the regression coefficients between them.

The r-dimensional, discrete-time, zero mean random noise vector $\mathbf{w}_d(t_{i-1})$ is depicted as r nodes in influence diagram form. In Figure 14, the nodes of $\mathbf{w}_d(t_{i-1})$ have no arrows drawn between them, implying that the covariance matrix $\mathbf{Q}_d(t_{i-1})$ is diagonal. The mean of all nodes of $\mathbf{w}_d(t_{i-1})$ is zero.

The vector $\mathbf{x}(t_i)$ is depicted by a set of n independent deterministic nodes. This is because it is a deterministic function of two independent Gaussian random vectors, $\mathbf{x}(t_{i-1})$ and $\mathbf{w}_d(t_{i-1})$. The linear relationship between $\mathbf{x}(t_i)$ and $\mathbf{x}(t_{i-1})$ is represented by the regression coefficients on the arrows from the nodes of $\mathbf{x}(t_{i-1})$ to the nodes of $\mathbf{x}(t_i)$. From Equation (26), this linear relationship

is the matrix $\Phi(t_i, t_{i-1})$. Consequently, the regression coefficients on the arrows between the two vectors are the elements of the state transition matrix. The coefficient on the arrow between the kth node of $\mathbf{x}(t_{i-1})$ and the jth node of $\mathbf{x}(t_i)$ corresponds to the (j,k) element of $\Phi(t_i, t_{i-1})$.

In this example, $\mathbf{Q}_d(t_{i-1})$ is assumed to be diagonal and the linear operation corresponding to $\mathbf{G}_d(t_{i-1})$ is depicted by arrows from the nodes of $\mathbf{w}_d(t_{i-1})$ to $\mathbf{x}(t_i)$. The coefficient on the arrow from the kth node of $\mathbf{w}_d(t_{i-1})$ to the jth node of $\mathbf{x}(t_i)$ is the (j,k) element of $\mathbf{G}_d(t_{i-1})$. The influence diagram in Figure 14 shows the vectors and matrices as just described.

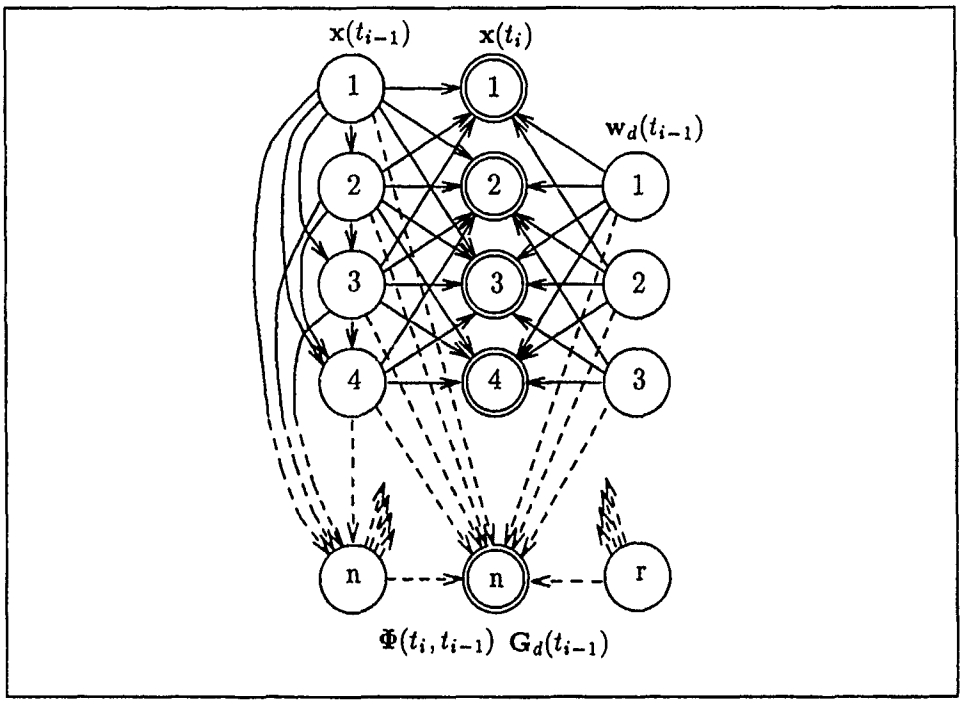


Figure 14. Influence Diagram Depiction of $\mathbf{x}(t_i) = \Phi(t_i, t_{i-1})\mathbf{x}(t_{i-1}) + \mathbf{G}_d(t_{i-1})\mathbf{w}_d(t_{i-1})$

As will be shown in Chapter 3, the matrix product $\mathbf{G}_d(t_{i-1})\mathbf{w}_d(t_{i-1})$ could have been factored such that $\mathbf{G}_d(t_{i-1})$ is the identity matrix and $\mathbf{Q}_d(t_{i-1})$ is an n-dimensional matrix which is not diagonal. In such a case, $\mathbf{w}_d(t_{i-1})$ will be depicted as n nodes with arcs between them corresponding to the influence diagram factorization of the non-diagonal matrix $\mathbf{Q}_d(t_{i-1})$. The identity matrix $\mathbf{G}_d(t_{i-1})$ will be depicted as n arrows. Each node of $\mathbf{w}_d(t_{i-1})$ will have only one arrow leading

from it to a single corresponding node of $\mathbf{x}_d(t_i)$. The regression coefficients on these arrows will be unity.

The entire influence diagram in Figure 14 can be thought of as $2n+r$ jointly Gaussian random variables making up a Gaussian random vector. It represents the joint distribution of $\mathbf{x}(t_{i-1})$, $\mathbf{x}(t_i)$, and $\mathbf{w}_d(t_{i-1})$. Because $\mathbf{w}_d(t_{i-1})$ is assumed to be independent of previous states and measurements, the density function $f_{\mathbf{w}_d(t_i)|\mathbf{x}(t_{i-1}),\mathbf{Z}(t_{i-1})}$ simplifies to $f_{\mathbf{w}_d(t_i)}$. The joint distribution of $\mathbf{x}(t_{i-1})$, $\mathbf{x}(t_i)$, and $\mathbf{w}_d(t_{i-1})$ can then be written in conditional form as:

$$f_{\mathbf{x}(t_i),\mathbf{x}(t_{i-1}),\mathbf{w}_d(t_{i-1})|\mathbf{Z}(t_{i-1})} = f_{\mathbf{x}(t_{i-1})|\mathbf{Z}(t_{i-1})}\ f_{\mathbf{w}_d(t_{i-1})}\ f_{\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}),\mathbf{w}_d(t_{i-1}),\mathbf{Z}(t_{i-1})} \tag{37}$$

The desired density function is $f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})}$. The nodes which represent this density function are those in the center column in Figure 14, labeled $\mathbf{x}(t_i)$. Using the the influence diagram to calculate this density function, the objective is to reverse the arcs in the diagram until the desired nodes are in unconditional form, i.e., at the beginning of the ordered sequence of nodes. This operation is equivalent to calculating

$$f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})} = \frac{f_{\mathbf{x}(t_{i-1})|\mathbf{Z}(t_{i-1})}\ f_{\mathbf{w}_d(t_{i-1})}\ f_{\mathbf{x}(t_i)|\mathbf{x}(t_{i-1}),\mathbf{w}_d(t_{i-1}),\mathbf{Z}(t_{i-1})}}{f_{\mathbf{x}(t_{i-1}),\mathbf{w}_d(t_{i-1})|\mathbf{x}(t_i),\mathbf{Z}(t_{i-1})}} \tag{38}$$

Equation (38) can be verified by multiplying both sides by the denominator of the right side. The result is Equation (37).

The nodes of $\mathbf{w}_d(t_{i-1})$ and $\mathbf{x}(t_{i-1})$ will be moved to the end of the ordered sequence of nodes where they can be removed as "nuisance" variables. An efficient way of removing these nodes is to first make the nodes of $\mathbf{w}_d(t_{i-1})$ conditioned upon the nodes of $\mathbf{x}(t_i)$. This process starts with the most conditional node of $\mathbf{w}_d(t_{i-1})$ and reverses arrows until it is conditioned on the entire vector $\mathbf{x}(t_i)$ as well. When this node is at the end of the ordered sequence (at the end of $\mathbf{x}(t_i)$) it can be removed. The process continues by reversing and removing the most conditional of the remaining

nodes of $\mathbf{w}_d(t_{i-1})$, one at a time. This results in a diagram with only $\mathbf{x}(t_{i-1})$ and $\mathbf{x}(t_i)$ as the first diagram in Figure 15.

Now the nodes of $\mathbf{x}(t_{i-1})$ can be conditioned upon the nodes of $\mathbf{x}(t_i)$ in the same manner, starting with the most conditional node. Alternatively, the least conditional (top) node of $\mathbf{x}(t_i)$ can be made less conditional by reversing arrows with the nodes in $\mathbf{x}(t_{i-1})$ until it is in unconditional form. Then the second node of $\mathbf{x}(t_i)$ can be moved until it is conditioned only on the first. The process repeats until the last node of $\mathbf{x}(t_i)$ is moved up. As each node of $\mathbf{x}(t_{i-1})$ successively becomes a "nuisance" variable, it is removed. Both methods are equally efficient and result in the second diagram of Figure 15.

This remaining vector of nodes represents the desired density function $f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})}$. The conditional mean of this vector is $\hat{\mathbf{x}}(t_i^-)$ while the regression coefficients and conditional variances of the nodes represent a factored form of $\mathbf{P}(t_i^-)$. The influence diagram algorithm renders the same conditional mean estimate and conditional covariance matrix as the Kalman filter propagation Equations (35) and (36).

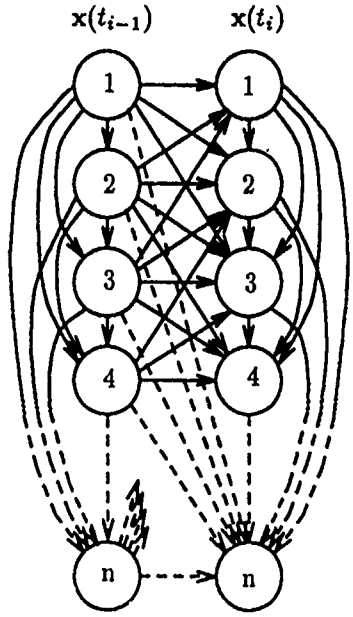
The second part of the Kalman filter assumes a linear measurement model with additive discrete-time Gaussian disturbances. The model is represented in the form:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{39}$$

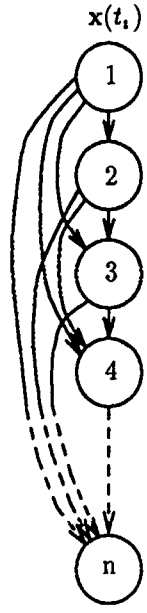
or, as will be used later:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{I}\mathbf{v}(t_i) \tag{40}$$

where $\mathbf{I}$ is the identity matrix of appropriate dimension. The measurement vector, $\mathbf{z}(t_i)$ is a linear combination of the states plus a discrete-time Gaussian noise vector. The term $\mathbf{H}(t_i)$ is the matrix which describes the linear combination of states. The term $\mathbf{v}(t_i)$ is the zero-mean, discrete-time,



Influence diagram after $\mathbf{w}_d(t_{i-1})$ removed



Influence diagram after $\mathbf{x}(t_{i-1})$ removed

Figure 15. Results of Removing the Vectors $\mathbf{w}_d(t_{i-1})$ and $\mathbf{x}(t_{i-1})$

Gaussian noise with covariance kernel:

$$E\{\mathbf{v}(t_i)\mathbf{v}^T(t_j)\} = \begin{cases} \mathbf{R}(t_i), & t_i = t_j, \\ 0, & t_i \neq t_j \end{cases} \tag{41}$$

The new conditional density function of the states, given all prior measurements through time t_i and a priori knowledge, is a Gaussian random vector. Because it is Gaussian, only the conditional mean and covariance matrix are needed to define the density function. The conditional mean of the state vector, conditioned on the measurements through time t_i, is $\hat{\mathbf{x}}(t_i^+)$ where:

$$\hat{\mathbf{x}}(t_i^+) \equiv E\{\mathbf{x}(t_i)|\mathbf{Z}(t_i)\} \tag{42}$$

and the new covariance matrix of the state estimate at time t_i, conditioned on measurements through time t_i, is defined by:

$$\mathbf{P}(t_i^+) \equiv E\{\left[\mathbf{x}(t_i) - \hat{\mathbf{x}}(t_i^+)\right]\left[\mathbf{x}(t_i) - \hat{\mathbf{x}}(t_i^+)\right]^T |\mathbf{Z}(t_i)\} \tag{43}$$

The Kalman filter calculates the updated conditional density function by first calculating a Kalman gain matrix $\mathbf{K}(t_i)$, then using it to calculate both the conditional mean and covariance. The conditional mean is the estimate of the state vector. The applicable equations are:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T(t_i)\left[\mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^T(t_i) + \mathbf{R}(t_i)\right]^{-1} \tag{44}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\left[\mathbf{z}_i - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)\right] \tag{45}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \tag{46}$$

Figure 16 shows how the influence diagram can depict the linear measurement model. The p-dimensional measurement vector $\mathbf{z}(t_i)$ is a deterministic function of the n-dimensional random vector $\mathbf{x}(t_i)$ and the p-dimensional random vector $\mathbf{v}(t_i)$. In this influence diagram, assume that the n-vector of nodes $\mathbf{x}(t_i)$ is the same as the n-vector of nodes in Figure 13. As such, it represents the conditional density given in Equation (38). The vector $\mathbf{v}(t_i)$ is a zero mean, discrete-time, Gaussian noise. The covariance matrix $\mathbf{R}(t_i)$ is represented in factored form by the variances of the nodes of $\mathbf{v}(t_i)$. The lack of arcs between the nodes of $\mathbf{v}(t_i)$ implies that $\mathbf{R}(t_i)$ is a diagonal matrix.

The measurement matrix $\mathbf{H}(t_i)$ is represented by the arrows from the nodes of $\mathbf{x}(t_i)$ to $\mathbf{z}(t_i)$. The regression coefficients on these arrows are the elements of $\mathbf{H}(t_i)$. They represent the linear combination of states that makes up the measurement vector. The identity matrix in Equation (40) is represented by the arrows from the nodes of $\mathbf{v}(t_i)$ to the nodes of $\mathbf{z}(t_i)$. The regression coefficients on these arrows from are all equal to one, corresponding to the ones in the identity matrix.

The desired density function is the vector $\mathbf{x}(t_i)$ conditioned on the vector $\mathbf{z}(t_i)$ (as well as on $\mathbf{Z}(t_{i-1})$). In the influence diagram, this means that the nodes of $\mathbf{z}(t_i)$ will be first in the ordered sequence, followed by the nodes of $\mathbf{x}(t_i)$. The nodes of $\mathbf{v}(t_i)$ are not needed and can be removed as nuisance variables. The required operations can take place in two steps.

The first step is to remove the nodes of $\mathbf{v}(t_i)$. The arrows between $\mathbf{v}(t_i)$ and $\mathbf{z}(t_i)$ are reversed until each of the nodes of $\mathbf{v}(t_i)$ are conditioned on $\mathbf{z}(t_i)$ and removed as nuisance variables. If $\mathbf{R}(t_i)$ is diagonal as shown in Figure 16, then each deterministic node of $\mathbf{z}(t_i)$ takes on the variance of the associated node of $\mathbf{v}(t_i)$. This operation is shown in the first influence diagram in in Figure 17.

If $\mathbf{R}(t_i)$ is not diagonal, then there are two options. One option is a transformation of variables from $\mathbf{z}(t_i)$ to $\mathbf{z}^*(t_i)$, yielding a new $\mathbf{H}^*(t_i)$ and $\mathbf{R}^*(t_i)$ such that $\mathbf{R}^*(t_i)$ is diagonal [7:pp. 375–377]. The other option is to factor $\mathbf{R}(t_i)$ into influence diagram form and remove the nodes one at a time. With the second option, the nodes of $\mathbf{z}(t_i)$ will have arrows between them after $\mathbf{v}(t_i)$ is removed, just as the nodes of $\mathbf{x}(t_i)$ had arrows between them in the top influence diagram in Figure 15.

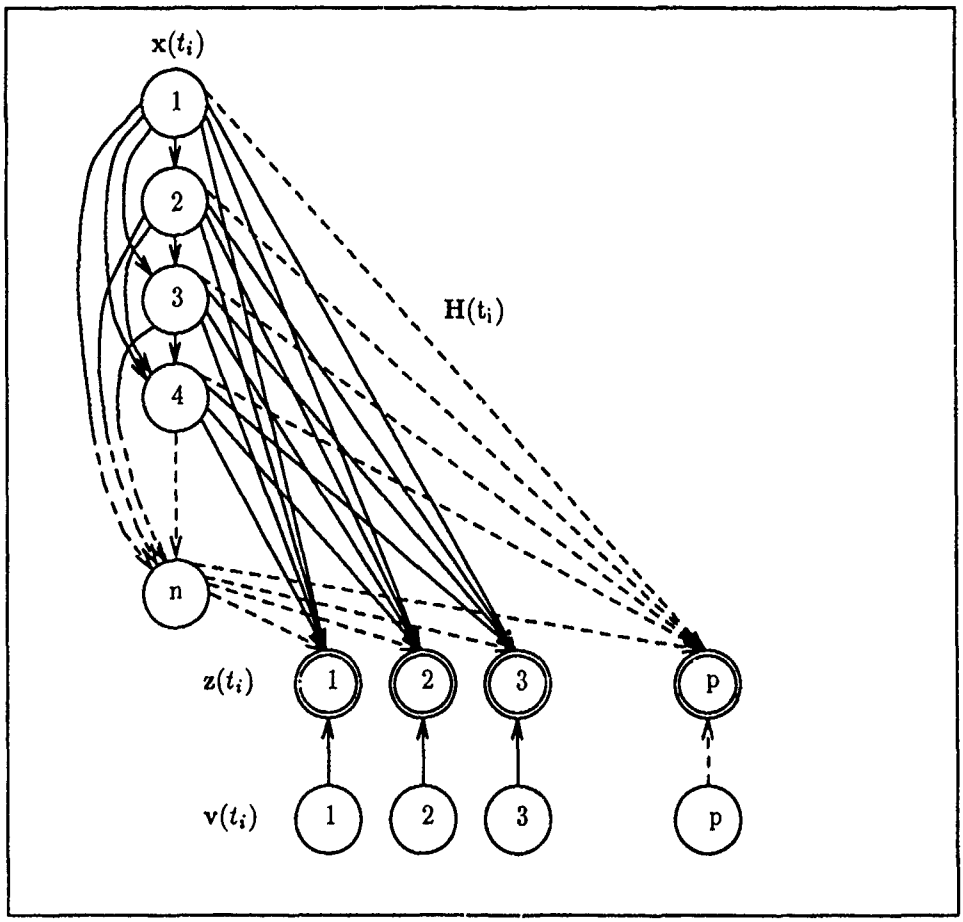


Figure 16. Measurement Update Model for $\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i)$

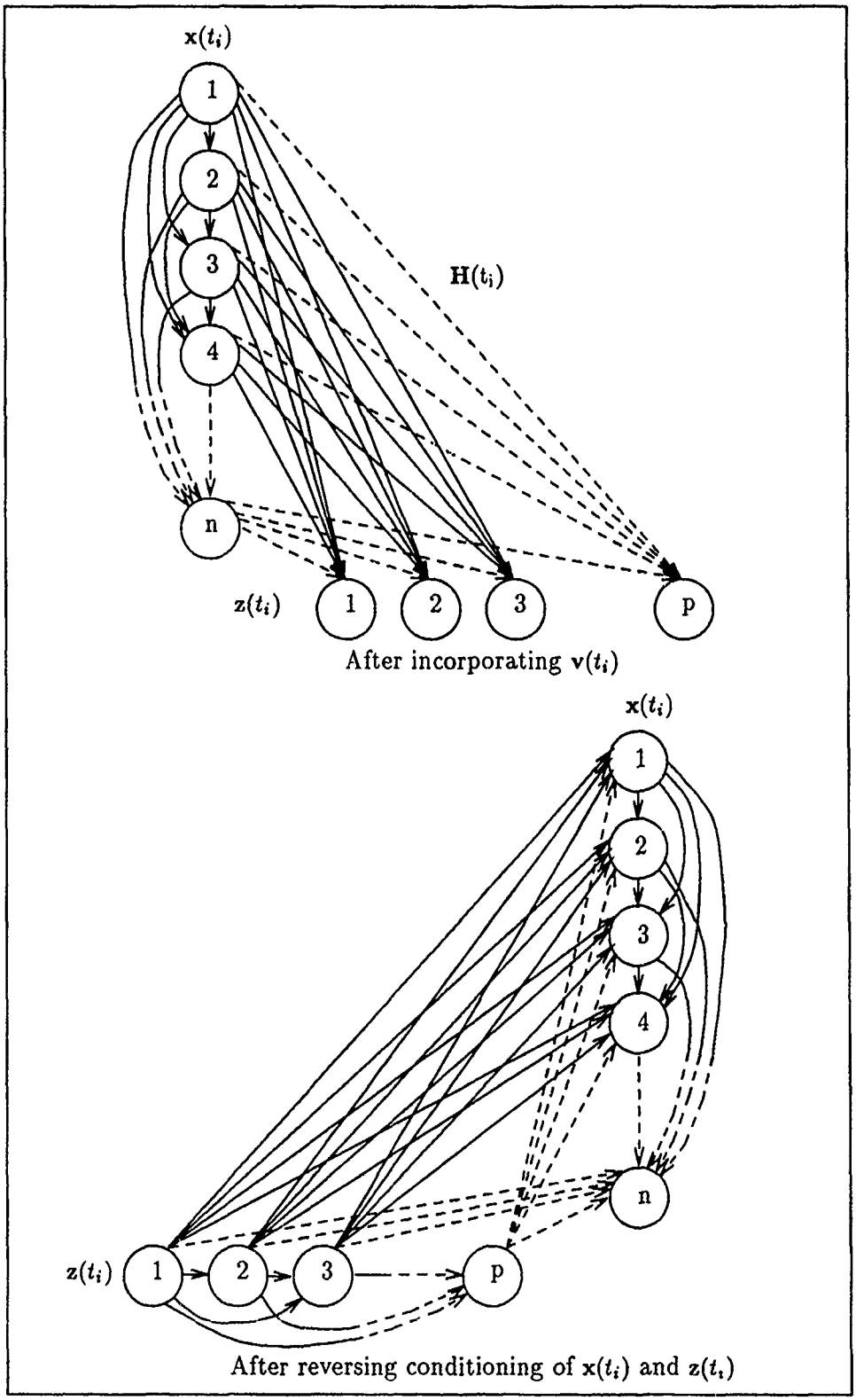


Figure 17. Results of Measurement Update Cycle

The second step in the process is to reverse the arrows between the new $\mathbf{z}(t_i)$ and $\mathbf{x}(t_i)$. The distribution of the vector $\mathbf{x}(t_i)$ is now conditioned on the vector $\mathbf{z}(t_i)$. These operations are shown in the second influence diagram in Figure 17.

The linear operation depicted by the arrows from $\mathbf{z}(t_i)$ to $\mathbf{x}(t_i)$ in Figure 17 represent the Kalman gain matrix. However, the influence diagram in Figure 17 does not use the notation $\mathbf{K}(t_i)$. This is because the regression coefficients on the arrows from $\mathbf{z}(t_i)$ to $\mathbf{x}(t_i)$ are not identical to the elements of the Kalman gain matrix. On the other hand, the total effect of all regression coefficients from node i of $\mathbf{z}(t_i)$ to node j of $\mathbf{x}(t_i)$ must be the (j,i) element in the Kalman gain matrix. This total effect includes the direct effects of the regression coefficients between the two vectors, as well as the effects between components within each vector. This relationship will be demonstrated in the numeric example of the following subsection.

The final calculation is to instantiate the nodes of $\mathbf{z}(t_i)$ by realizing the random variables in order, and removing the nodes. With each realization, the difference between the mean and the realized value of the random variable (the residual) is propagated, via the regression coefficients, to all subsequent nodes. The remaining vector $\mathbf{x}(t_i)$ is now conditioned on the measurements $\mathbf{z}(t_i)$ and $\mathbf{Z}(t_{i-1})$, or simply $\mathbf{Z}(t_i)$

Similar to the prior example, the entire influence diagram in Figure 16 can be thought of as $2p + n$ jointly Gaussian random variables making up a Gaussian random vector. It represents the joint distribution of $\mathbf{x}(t_i)$, $\mathbf{z}(t_i)$, and $\mathbf{v}(t_i)$. Since $\mathbf{v}(t_i)$ is independent of previous measurements and state estimates, $f_{\mathbf{v}(t_i)|\mathbf{x}(t_i),\mathbf{Z}(t_{i-1})} = f_{\mathbf{v}(t_i)}$. The joint distribution of $\mathbf{x}(t_i)$, $\mathbf{z}(t_i)$, and $\mathbf{v}(t_i)$ can then be written in conditional form as:

$$f_{\mathbf{x}(t_i),\mathbf{z}(t_i),\mathbf{v}(t_i)|\mathbf{Z}(t_{i-1})} = f_{\mathbf{v}(t_i)}\ f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})}\ f_{\mathbf{z}(t_i)|\mathbf{x}(t_i),\mathbf{v}(t_i),\mathbf{Z}(t_{i-1})} \tag{47}$$

The desired density function is $f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})\mathbf{z}(t_i)} = f_{\mathbf{x}(t_i)|\mathbf{Z}(t_i)}$. This density function is represented by the arrangement of nodes in the second influence diagram of Figure 17. In this diagram, the nodes of $\mathbf{x}(t_i)$ are conditioned upon the nodes of $\mathbf{z}(t_i)$. Because the nodes of $\mathbf{x}(t_i)$ were already conditioned on $\mathbf{Z}(t_{i-1})$, the resultant density function is conditioned on all measurements through time t_i, represented as $\mathbf{Z}(t_i)$.

The objective of the influence diagram manipulations is to reverse arcs as necessary to achieve the desired form of the influence diagram. This operation is done in two steps. First, the top diagram in Figure 17 is the joint density of $\mathbf{x}(t_i)$ and $\mathbf{z}(t_i)$ expressed in influence diagram form as the marginal density of one vector and the conditional density of the second vector, given the first. The influence diagram operations represent the equations:

$$f_{\mathbf{x}(t_i),\mathbf{z}(t_i)|\mathbf{Z}(t_{i-1})} = \frac{f_{\mathbf{x}(t_i),\mathbf{z}(t_i),\mathbf{v}(t_i)|\mathbf{Z}(t_{i-1})}}{f_{\mathbf{v}(t_i)|\mathbf{z}(t_i)\mathbf{x}(t_i)}} \tag{48}$$

$$= \frac{f_{\mathbf{v}(t_i)}\ f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})}\ f_{\mathbf{z}(t_i)|\mathbf{x}(t_i),\mathbf{v}(t_i)\mathbf{Z}(t_{i-1})}}{f_{\mathbf{v}(t_i)|\mathbf{z}(t_i),\mathbf{x}(t_i)}} \tag{49}$$

This equation can be verified by multiplying both sides by the denominator.

The top diagram of Figure 17 is further changed to the bottom diagram by rearranging the joint density of $\mathbf{x}(t_i)$ and $\mathbf{z}(t_i)$ into a conditional form with $\mathbf{x}(t_i)$ conditioned on $\mathbf{z}(t_i)$. The influence diagram operations represent the equations:

$$f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})\mathbf{z}(t_i)} = \frac{f_{\mathbf{x}(t_i),\mathbf{z}(t_i)|\mathbf{Z}(t_{i-1})}}{f_{\mathbf{z}(t_i)|\mathbf{Z}(t_{i-1})}} \tag{50}$$

$$= \frac{f_{\mathbf{x}(t_i)|\mathbf{Z}(t_{i-1})}\ f_{\mathbf{z}(t_i)|\mathbf{x}(t_i)\mathbf{Z}(t_{i-1})}}{f_{\mathbf{z}(t_i)|\mathbf{Z}(t_{i-1})}} \tag{51}$$

At this time, the regression coefficients and conditional variances of $\mathbf{x}(t_i)$ represent the conditional covariance matrix of $f_{\mathbf{x}(t_i)|\mathbf{Z}(t_i)}$. However, the influence diagram represents the conditional mean in equation form as a function of the realization of the components of $\mathbf{z}(t_i)$. To complete the

operation, the components of $\mathbf{z}(t_i)$ are realized one at a time, in order, and the appropriate nodes are instantiated. When all nodes of $\mathbf{z}(t_i)$ are removed, then the remaining nodes of $\mathbf{x}(t_i)$ represent the conditional density $f_{\mathbf{x}(t_i)|\mathbf{Z}(t_i)}$ in numerical form.

Because the remaining density is Gaussian, it can be represented as the conditional mean vector and the conditional covariance matrix. Kenley showed that the conditional mean vector of $\mathbf{x}(t_i)$ calculated by the influence diagram is identical to the Kalman filter estimate of the states, $\hat{\mathbf{x}}(t_i^+)$. Furthermore, the conditional covariance matrix represented by the influence diagram is equivalent to $\mathbf{P}(t_i^+)$.

The influence diagram representation of the discrete-time system models of Equations (26) and (40) over one time interval can be depicted as one large influence diagram. Such an influence diagram is shown in Figure 18.

Normally, a single time interval is one of many, perhaps infinite, intervals. The influence diagram can depict multiple time intervals, but it is messy. One simplification assumes each node represents an entire random vector. This simplification allows depiction of a series of time intervals in a more orderly manner. Figure 19 depicts the first three time intervals of a linear system as given in Equations (26) and (40) in vector influence diagram form. Each node depicts an influence diagram for an entire random vector. Arrows between nodes represents a matrix linear operation between two vectors.

One practical problem with implementing the influence diagram over many intervals is that when a measurement is made, the residual will be propagated to all successor nodes. In reality, there may be many successors. Using the influence diagram to propagate the new conditional means to all successors may not be practical. Kenley's original work did not address the infinite successor problem, but there is a simple solution, as will be shown later.

2.6.1 Numerical Example of the Discrete-Time Kalman Filter. Assume a model of a one-dimensional target tracking problem. In this problem, the target's position is the integral of the

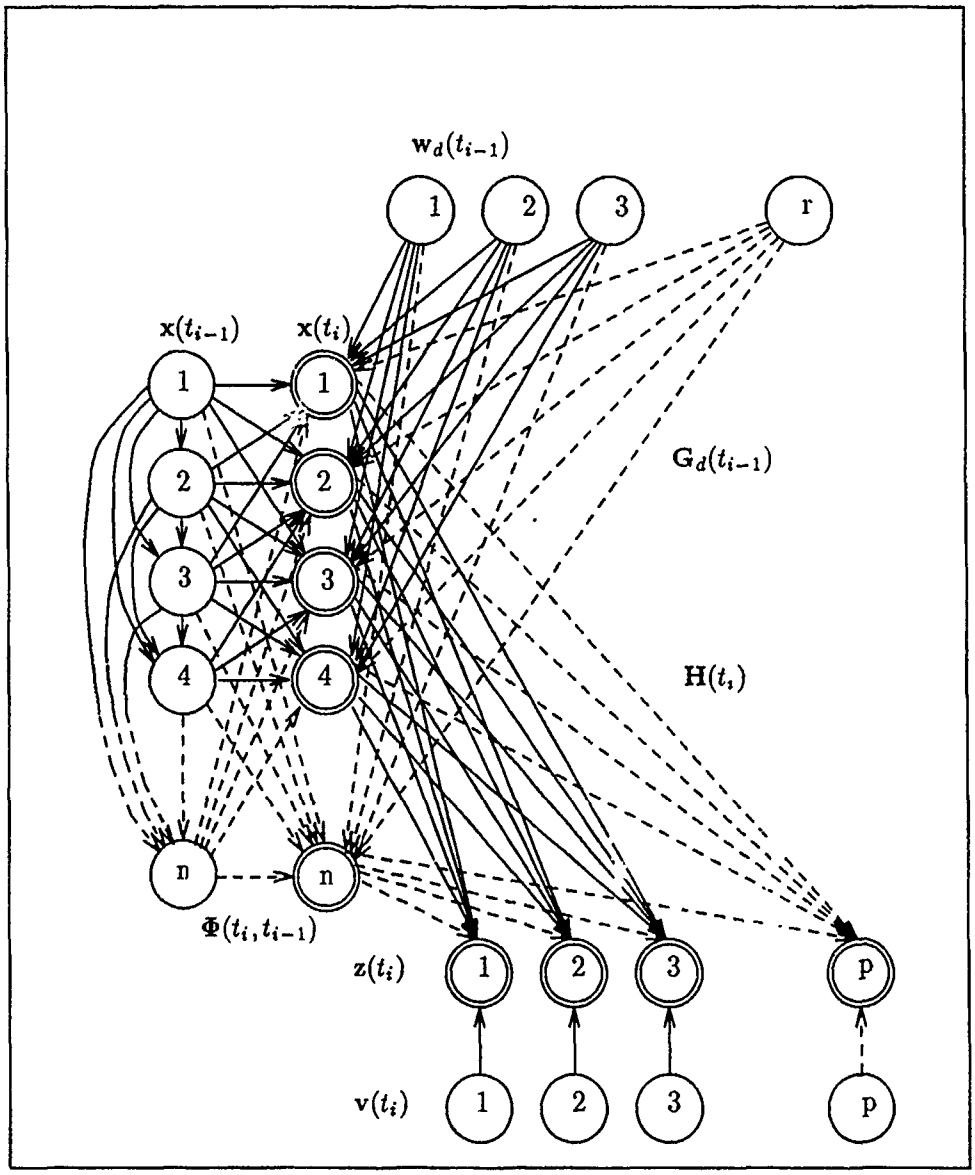


Figure 18. Influence Diagram Representation of Discrete-Time System Model

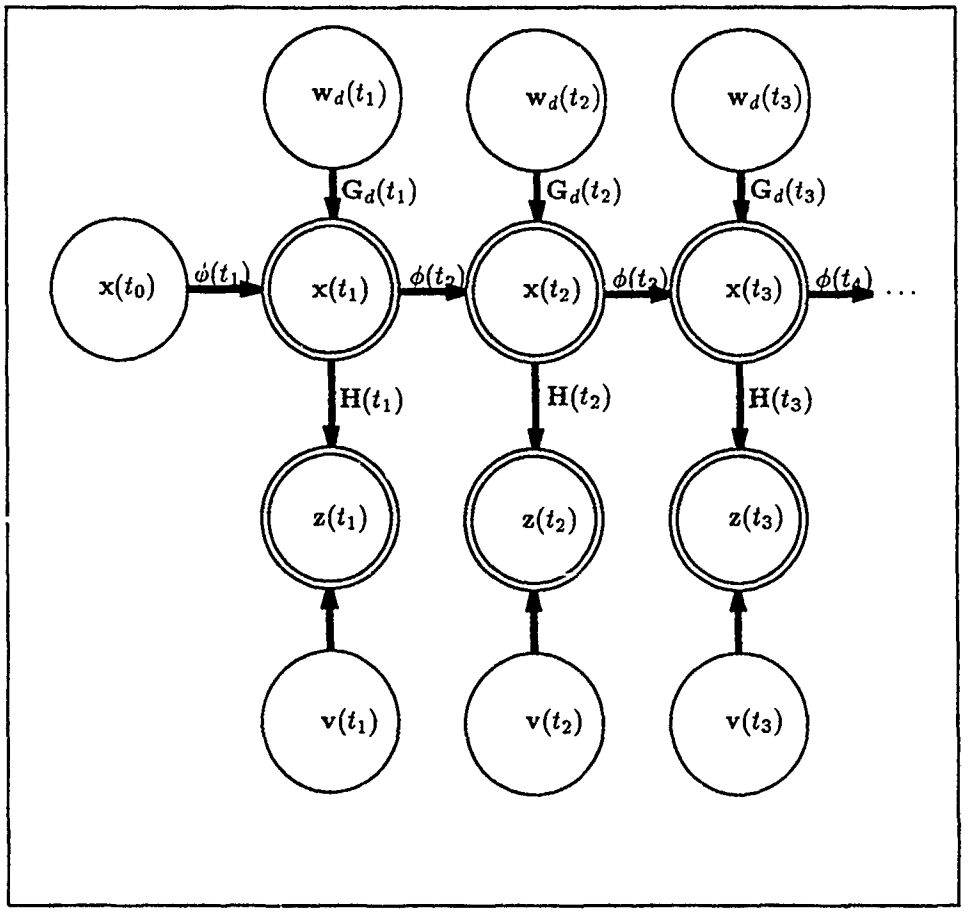


Figure 19. Vector Form of Discrete-Time System Model Over Several Time Intervals

velocity, and the velocity is the integral of the acceleration. The acceleration is modeled as a continuous-time Gaussian white noise $w(t)$ through a first order lag filter. The equations of motion in state variable form are:

$$\dot{x}_1(t) = x_2(t) \tag{52}$$

$$\dot{x}_2(t) = x_3(t) \tag{53}$$

$$\dot{x}_3(t) = -\frac{1}{T}x_3(t) + w(t) \tag{54}$$

where T is the time constant of the first order lag filter. In matrix form, these equations are:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{55}$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{T} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \tag{56}$$

The white, Gaussian noise model is zero mean with an autocorrelation kernel of:

$$E\left[w(t)w(t+\tau)\right] = Q\delta(\tau) \tag{57}$$

In this example, let $Q = 2/T$. This yields an autocorrelation kernel for $x_3(t)$ of:

$$E\left[x_3(t)x_3(t+\tau)\right] = e^{-|\tau|/T} \tag{58}$$

The state transition matrix can be computed, for a constant time interval of τ, as:

$$\Phi(t+\tau,t) = L^{-1}\{[s\mathbf{I}-\mathbf{F}]^{-1}\} = \begin{bmatrix} 1 & \tau & T^2(\tau/T - 1 + e^{-\tau/T}) \\ 0 & 1 & T(1-e^{-\tau/T}) \\ 0 & 0 & e^{-\tau/T} \end{bmatrix} \tag{59}$$

where $L^{-1}\{\}$ indicates the inverse Laplace transform. Assume $T = 2$ and $\tau = 1$ so that the sample time is less than the time constant of the first order lag filter. The simplified the state transition matrix, to four significant digits, is:

$$\Phi(t+1,t) = \begin{bmatrix} 1 & 1 & 0.4261 \\ 0 & 1 & 0.7870 \\ 0 & 0 & 0.6065 \end{bmatrix} \tag{60}$$

The discrete-time equivalent noise w_d is zero mean, and has covariance kernel

$$E\{\mathbf{w}_d(t_i)\mathbf{w}_d^T(t_j)\} = \begin{cases} \mathbf{Q}_d(t_i) = \int_0^1 \Phi(1,s)\mathbf{G}\mathbf{Q}\mathbf{G}^T\Phi^T(1,s)ds & t_i = t_j, \\ \mathbf{0} & t_i \neq t_j \end{cases} \tag{61}$$

where $\mathbf{G}$ and $\mathbf{Q}$ are time invariant. The result of the integration is a time-invariant, discrete-time equivalent covariance matrix $\mathbf{Q}_d$ calculated to four significant digits as:

$$\mathbf{Q}_d = \begin{bmatrix} 3.063 & -2.336 & -0.5677 \\ -2.336 & 1.904 & 0.4160 \\ -0.5677 & 0.4160 & 0.1080 \end{bmatrix} \tag{62}$$

The matrix $\mathbf{G}_d$ is assumed to be the identity matrix. Alternatively, the matrix product $\mathbf{G}_d\mathbf{Q}_d\mathbf{G}_d^T$ could take on the value assigned to $\mathbf{Q}_d$ and an arbitrary $\mathbf{G}_d$ could be factored from the product (for instance, let $\mathbf{G}_d$ be the U-factor and $\mathbf{Q}_d$ be the D-factor of $\mathbf{G}_d\mathbf{Q}_d\mathbf{G}_d^T$, as in the U-D filter [7:396]).

The initial position, velocity, and acceleration are known, with an initial covariance matrix as:

$$\mathbf{x}(t_0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{63}$$

$$\mathbf{P}(t_0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{64}$$

Measurement are taken at the end of the time interval τ according to the model:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + v(t_i) \tag{65}$$

$$\mathbf{z}(t_i) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}(t_i) + v(t_i) \tag{66}$$

where $v(t_i)$ is a discrete-time, zero mean, Gaussian noise with covariance kernel:

$$E\{v(t_i)v(t_j)\} = \begin{cases} 1 & t_i = t_j, \\ 0 & t_i \neq t_j \end{cases} \tag{67}$$

The Kalman filter equations for the state estimate and covariance matrix at time t_1, before the measurements are made, are calculated by Equations (35) and (36). These equations yield:

$$\hat{\mathbf{x}}(t_1^-) = \begin{bmatrix} 1 & 1 & 0.4261 \\ 0 & 1 & 0.7870 \\ 0 & 0 & 0.6065 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{68}$$

$$\hat{\mathbf{x}}(t_1^-) = \begin{bmatrix} 2.426 \\ 1.787 \\ 0.6065 \end{bmatrix} \tag{69}$$

$$\begin{aligned} \mathbf{P}(t_1^-) &= \begin{bmatrix} 1 & 1 & 0.4261 \\ 0 & 1 & 0.7870 \\ 0 & 0 & 0.6065 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0.4261 \\ 0 & 1 & 0.7870 \\ 0 & 0 & 0.6065 \end{bmatrix}^T \\ &+ \begin{bmatrix} 3.063 & -2.336 & -0.5677 \\ -2.336 & 1.904 & 0.4160 \\ -0.5677 & 0.4160 & 0.1080 \end{bmatrix} \end{aligned} \tag{70}$$

$$\mathbf{P}(t_1^-) = \begin{bmatrix} 5.244 & -1.001 & -0.3092 \\ -1.001 & 3.523 & 0.8933 \\ -0.3092 & 0.8933 & 0.4760 \end{bmatrix} \tag{71}$$

The Kalman filter equations given in Equations (44) through (46) are the state estimate and covariance matrix at time t_1, after the measurements. Assume the measurement of the position, according to Equation (39), results in an observation of 2.000. Then the Kalman filter equations result in:

$$\mathbf{HP}(t_1^-)\mathbf{H}^T + \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5.244 & -1.001 & -0.3092 \\ -1.001 & 3.523 & 0.8933 \\ -0.3092 & 0.8933 & 0.4760 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + [1] \tag{72}$$

$$= 6.244 \tag{73}$$

$$\mathbf{K}(t_1) = \begin{bmatrix} 5.244 & -1.001 & -0.3092 \\ -1.001 & 3.523 & 0.8933 \\ -0.3092 & 0.8933 & 0.4760 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} (1/6.244) \tag{74}$$

$$\mathbf{K}(t_1) = \begin{bmatrix} 0.8399 \\ -0.1603 \\ -0.04952 \end{bmatrix} \tag{75}$$

$$\hat{\mathbf{x}}(t_1^+) = \begin{bmatrix} 2.426 \\ 1.787 \\ 0.6065 \end{bmatrix} + \begin{bmatrix} 0.8399 \\ -0.1603 \\ -0.04952 \end{bmatrix} (2.000 - \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2.426 \\ 1.787 \\ 0\ 6065 \end{bmatrix}) \tag{76}$$

$$\hat{\mathbf{x}}(t_1^+) = \begin{bmatrix} 2.068 \\ 1.855 \\ 0.6276 \end{bmatrix} \tag{77}$$

$$\mathbf{P}(t_i^+) = \begin{bmatrix} 5.244 & -1.001 & -0.3092 \\ -1.001 & 3.523 & 0.8933 \\ -0.3092 & 0.8933 & 0.4760 \end{bmatrix} - \begin{bmatrix} 0.8399 \\ -0.1603 \\ -0.04952 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5.244 & -1.001 & -0.3092 \\ -1.001 & 3.523 & 0.8933 \\ -0.3092 & 0.8933 & 0.4760 \end{bmatrix} \tag{78}$$

$$\mathbf{P}(t_i^+) = \begin{bmatrix} 0.8399 & -0.1603 & -0.04952 \\ -0.1603 & 3.363 & 0.8437 \\ -0.04952 & 0.8437 & 0.4607 \end{bmatrix} \tag{79}$$

2.6.2 Numerical Example of Discrete-Time Influence Diagram Filter. The influence diagram can be used to calculate the conditional probability densities for the state estimates at time t_1, as was done by the Kalman filter equations in the previous subsection. The following series of figures are the influence diagram operations for this numerical example. Figure 20 depicts the entire model for the propagation of the states from time t_0 to time t_1 as well for the measurement model at time t_1. The labels on the nodes correspond to the nomenclature in the model equations (26) and (40).

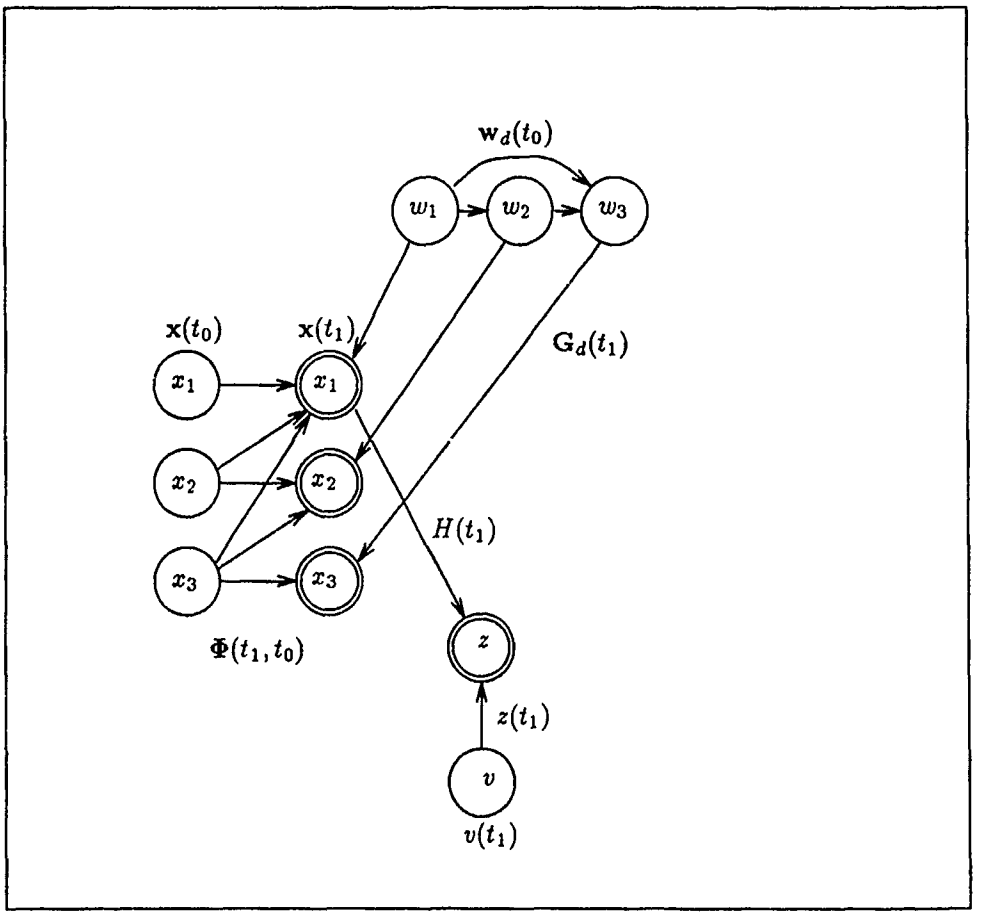


Figure 20. One Cycle of Example Discrete-Time Filter

Figure 21 is the same diagram with the nodes numbered one through eleven. This permits simpler labels on subsequent influence diagram. The second diagram also assumes that the initial state estimates (the unconditional mean of the vector $\mathbf{x}(t_0)$), have been propagated to all successor

nodes by the regression coefficients. This is equivalent to assuming the a priori information is a realization of the random vector $\mathbf{x}(t_0)$, but the nodes of the random vector remain on the diagram and are not instantiated.

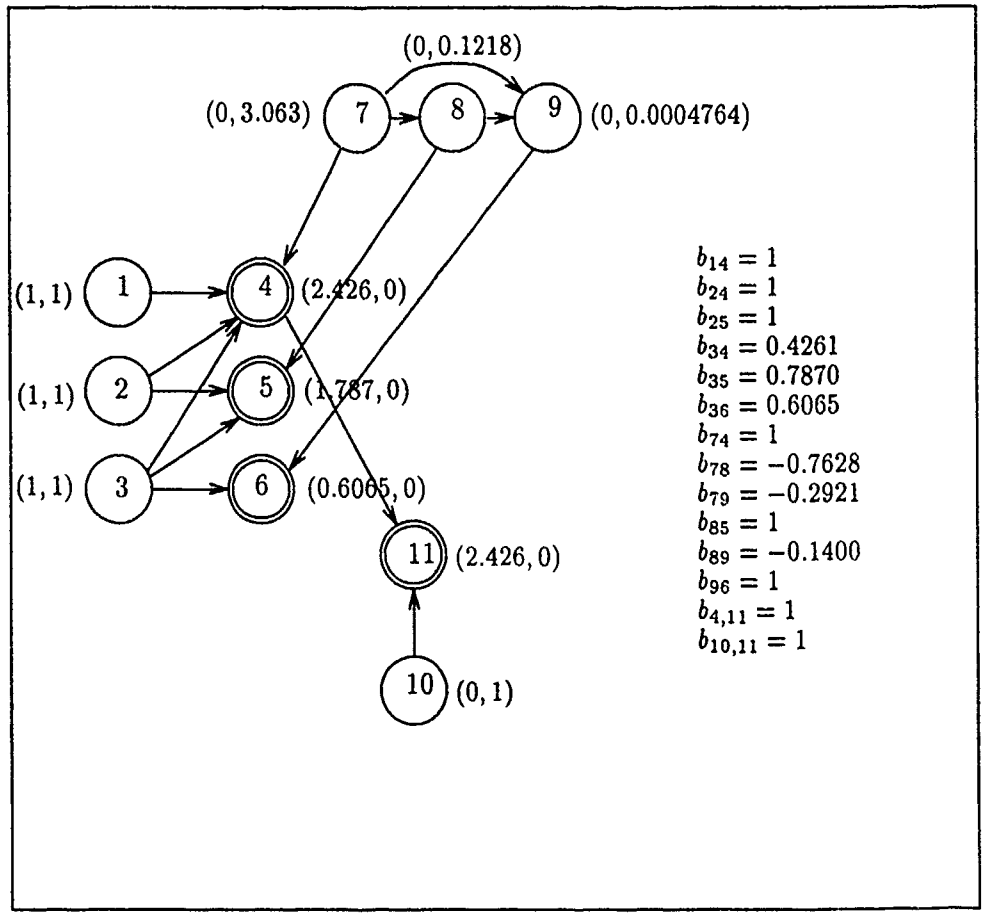


Figure 21. Example of Discrete-Time Filter with Nodes Relabeled

Each node has a mean and variance associated with it, given in the form (μ_i, v_i). The regression coefficients are written on the right side of the diagram, rather than directly on the lines, for legibility. All values on the influence diagram come from the following matrix equations.

$$\Phi(t_1, t_0) = \begin{bmatrix} 1 & 1 & 0.4261 \\ 0 & 1 & 0.7870 \\ 0 & 0 & 0.6065 \end{bmatrix} \tag{80}$$

$$\mathbf{Q}_d(t_0) = \begin{bmatrix} 3.063 & -2.336 & -0.5677 \\ -2.336 & 1.904 & 0.4160 \\ -0.5677 & 0.4160 & 0.1080 \end{bmatrix} \tag{81}$$

$$\begin{aligned} \mathbf{Q}_d(t_0) = & \begin{bmatrix} 1 & 0 & 0 \\ -0.7628 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -0.2922 & 1 & 0 \\ -0.1400 & 0 & 1 \end{bmatrix} \\ & \begin{bmatrix} 3.063 & 0 & 0 \\ 0 & 0.1218 & 0 \\ 0 & 0 & 0.0004764 \end{bmatrix} \begin{bmatrix} 1 & -0.7628 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.2922 \\ 0 & 1 & -0.1400 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{82}$$

$$\mathbf{G}_d(t_0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{83}$$

$$\mathbf{P}(t_0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{84}$$

$$\hat{\mathbf{x}}(t_0) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{85}$$

$$\mathbf{H}(t_1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{86}$$

$$v(t_1) = 1 \tag{87}$$

Each diagram starting with Figure 22 depicts a single operation of arc reversal, node removal, or measurement update. These subsequent diagrams also show the variance on the left side of the diagram for legibility. The means are not shown until they change during the update. At the top of each of these diagrams, there is a short description of the changes between the current and the previous diagram. The equations for calculating the changes in the variances and regression coefficients are repeated here in Equations (88) through (92). The actual calculations are not shown, but the predecessor node (node i), the successor node (node j), and the common predecessors of both (the set of nodes K) are identified on each diagram. Finally, the changed values are emphasized in each diagram.

$$v_j' = v_j + b_{ij}^2 v_i \tag{88}$$

$$b_{kj}' = b_{kj} + b_{ki} b_{ij} \tag{89}$$

$$v_i' = \frac{v_i v_j}{v_j'} \tag{90}$$

$$b_{ji}' = \frac{v_i b_{ij}}{v_j'} \tag{91}$$

$$b_{ki}' = b_{ki} + b_{kj}' b_{ji}' \tag{92}$$

As stated earlier, the total effect of the regression coefficients from the measurement node to the state estimates is identical to the Kalman gain matrix. Furthermore, the variance of the measurement node, just prior to the measurement in Figure 38 is identical to the value calculated for $\mathbf{HP}(t_1^-)\mathbf{H}^T + \mathbf{R}$ in the Kalman filter. The means of the nodes are identical to the vector $\hat{\mathbf{x}}(t_1^+)$. The covariance matrix represented by the variances and regression coefficients in Figure 36 is the same as the covariance matrix calculated with the Kalman filter in the previous subsection

as $\mathbf{P}(t_1^-)$. This can be verified by multiplying the matrices as:

$$\mathbf{P}(t_1^-) = \begin{bmatrix} 1 & 0 & 0 \\ -0.1908 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -0.01118 & 1 & 0 \\ 0.2504 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5.244 & 0 & 0 \\ 0 & 3.332 & 0 \\ 0 & 0 & 0.24886 \end{bmatrix} \begin{bmatrix} 1 & -0.1908 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.01118 \\ 0 & 1 & 0.2504 \\ 0 & 0 & 1 \end{bmatrix} \quad (93)$$

$$\mathbf{P}(t_1^-) = \begin{bmatrix} 5.244 & -1.001 & -0.3092 \\ -1.001 & 3.523 & 0.8933 \\ -0.3092 & 0.8933 & 0.4760 \end{bmatrix} \quad (94)$$

Similarly, the covariance matrix in Figure 39 can be verified by multiplying:

$$\mathbf{P}(t_1^+) = \begin{bmatrix} 1 & 0 & 0 \\ -0.1908 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -0.01118 & 1 & 0 \\ 0.2504 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8399 & 0 & 0 \\ 0 & 3.332 & 0 \\ 0 & 0 & 0.24886 \end{bmatrix} \begin{bmatrix} 1 & -0.1908 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.01118 \\ 0 & 1 & 0.2504 \\ 0 & 0 & 1 \end{bmatrix} \quad (95)$$

$$\mathbf{P}(t_i^+) = \begin{bmatrix} 0.8399 & -0.1603 & -0.04952 \\ -0.1603 & 3.363 & 0.8437 \\ -0.04952 & 0.8437 & 0.4607 \end{bmatrix} \quad (96)$$

2.6.3 Chapter Summary. This chapter was a tutorial approach to influence diagrams for probabilistic and deterministic variables. It was intended to summarize the work of numerous

authors. It began with the first type of influence diagram created, the discrete variable type. The basic workings of the diagram were shown and explained using an example. The next type of influence diagram used Gaussian random variables. The mathematics of this type of diagram were explained using another example. The final result was an explanation and demonstration of the Gaussian influence diagram for discrete-time filtering. There was also a demonstration of some of the similarities between the the Kalman filter and the influence diagram.

The purpose of the explanations in this chapter was to prepare the reader for more detailed discussion of the influence diagram in subsequent chapters. The implementation of the discrete-time filter was due to Kenley [6], but the next chapter will demonstrate some changes in implementation that will make the discrete-time filter algorithm more efficient.

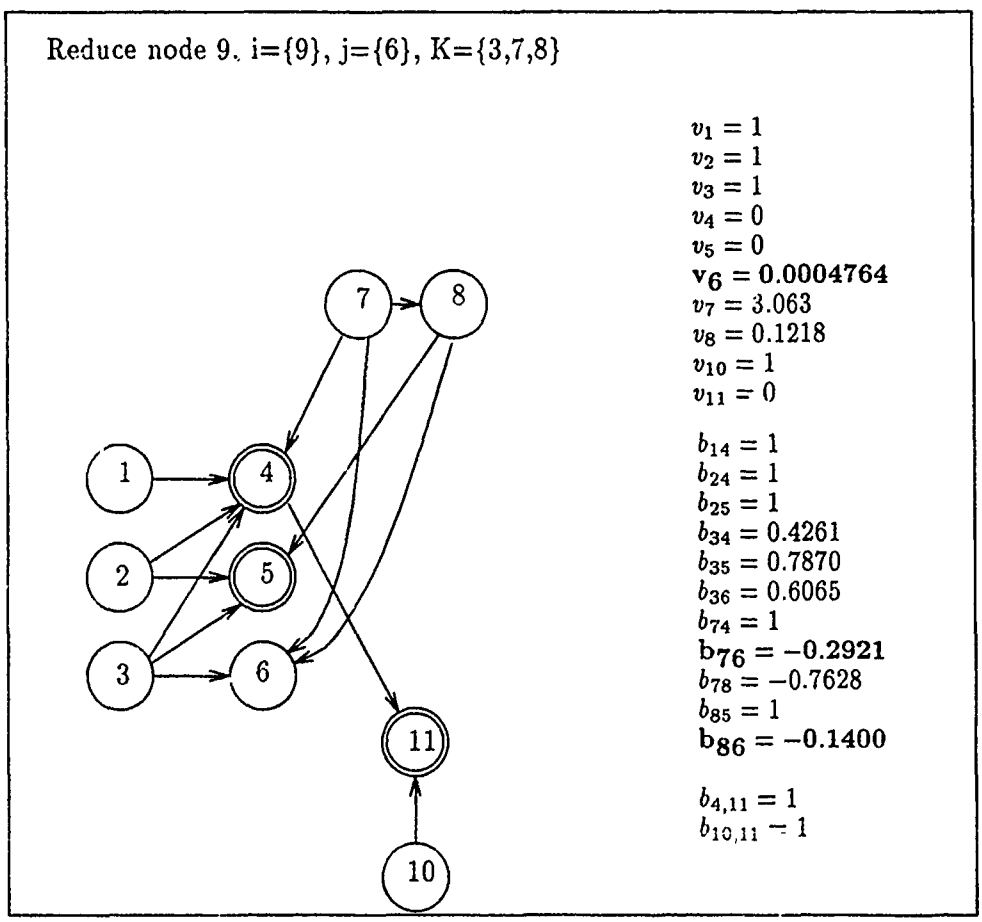


Figure 22. Example of Discrete-Time Filter with Node 9 Removed

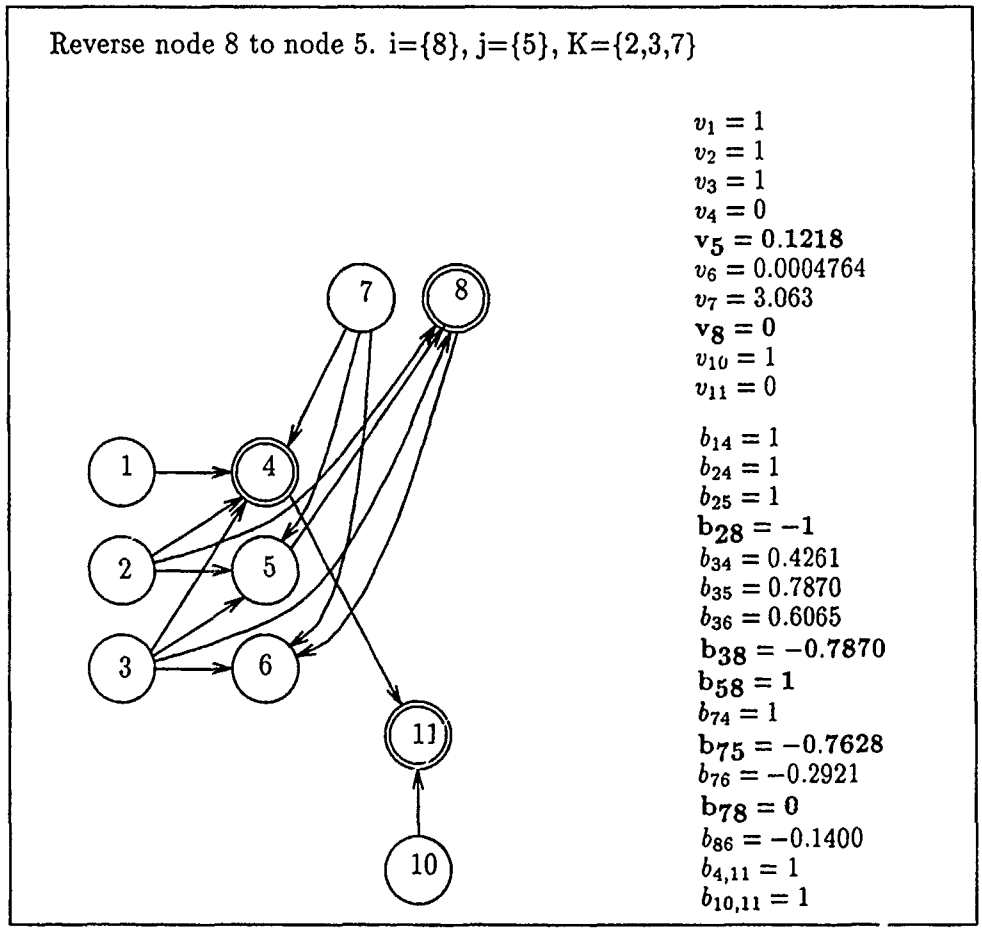


Figure 23. Example of Discrete-Time Filter with Arc Between Nodes 8 and 5 Reversed

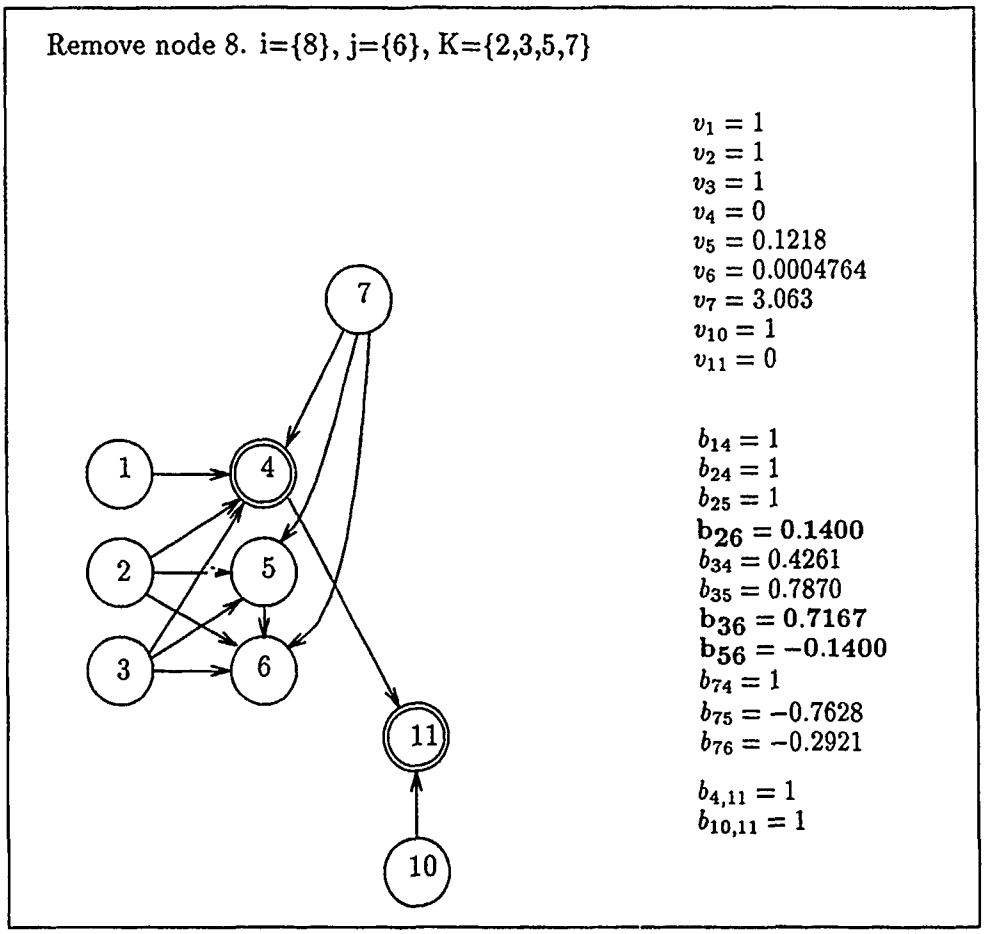


Figure 24. Example of Discrete-Time Filter with Node 8 Removed

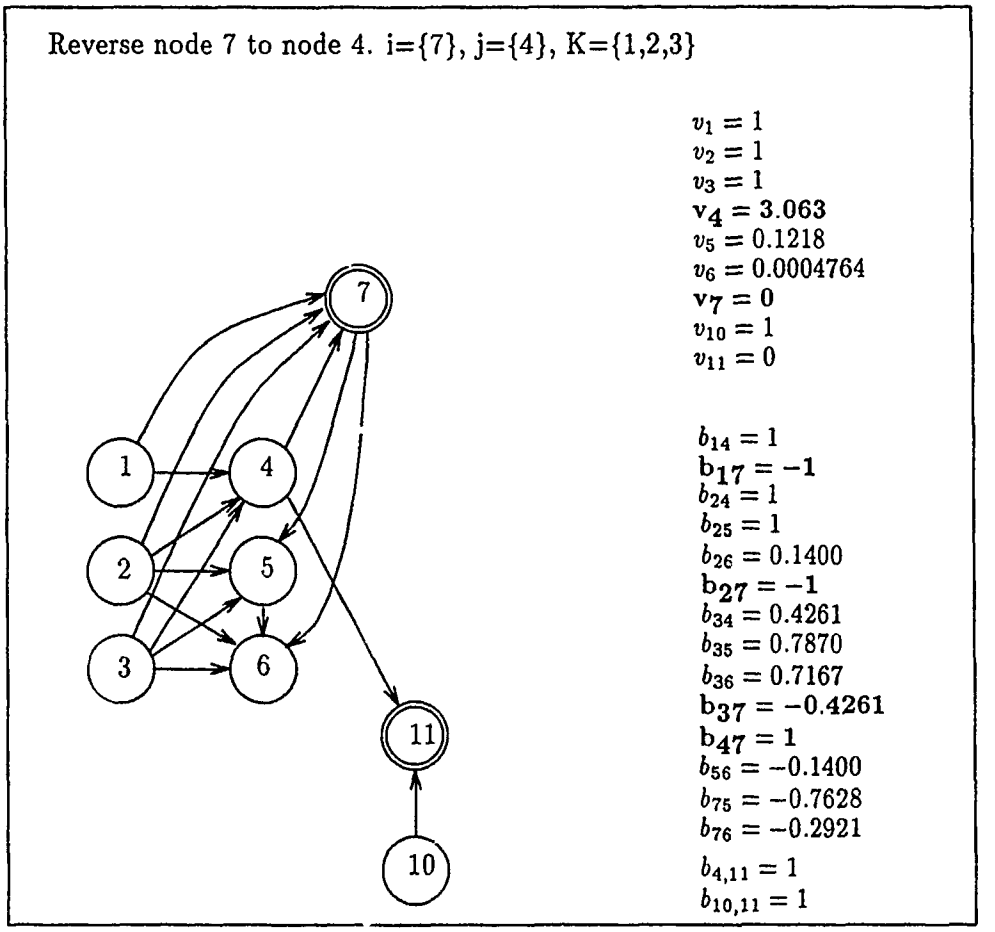


Figure 25. Example of Discrete-Time Filter with Arc Between Nodes 7 and 4 Reversed

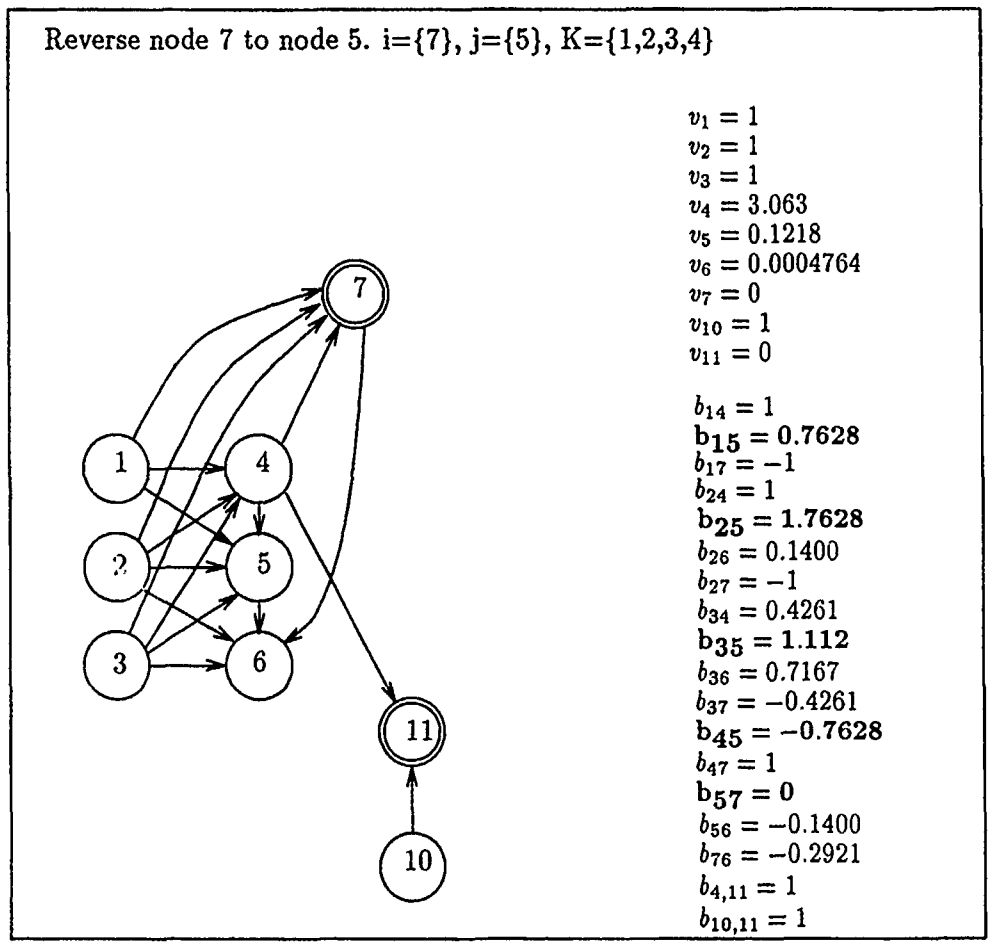


Figure 26. Example of Discrete-Time Filter with Arc Between Nodes 7 and 5 Reversed

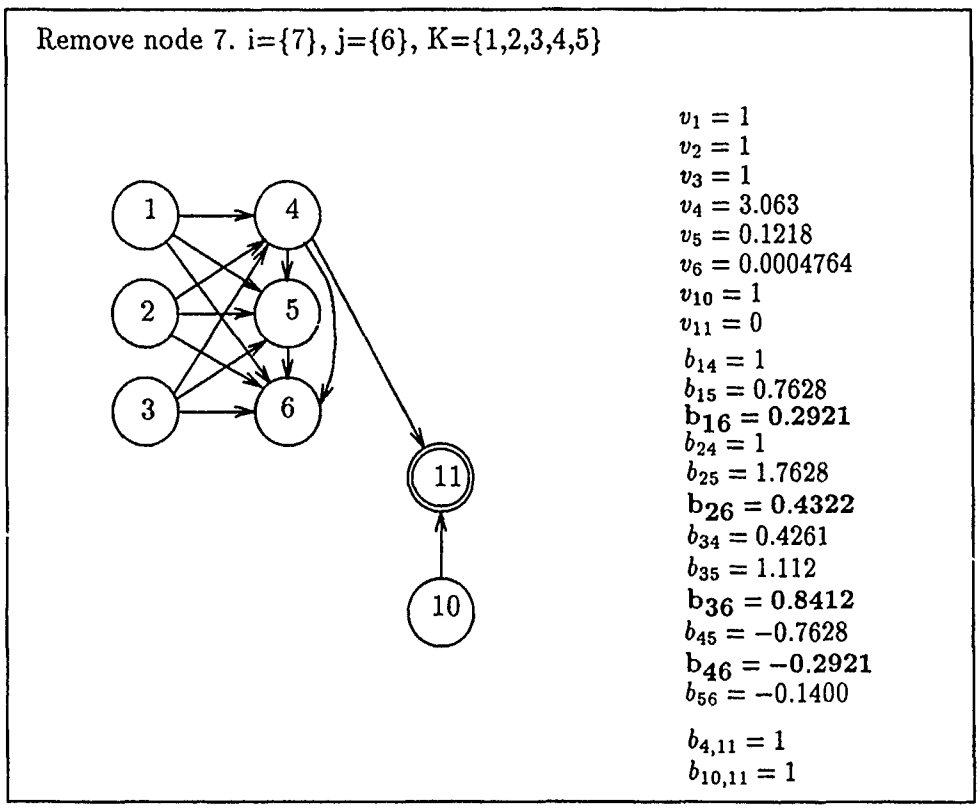


Figure 27. Example of Discrete-Time Filter with Node 7 Removed

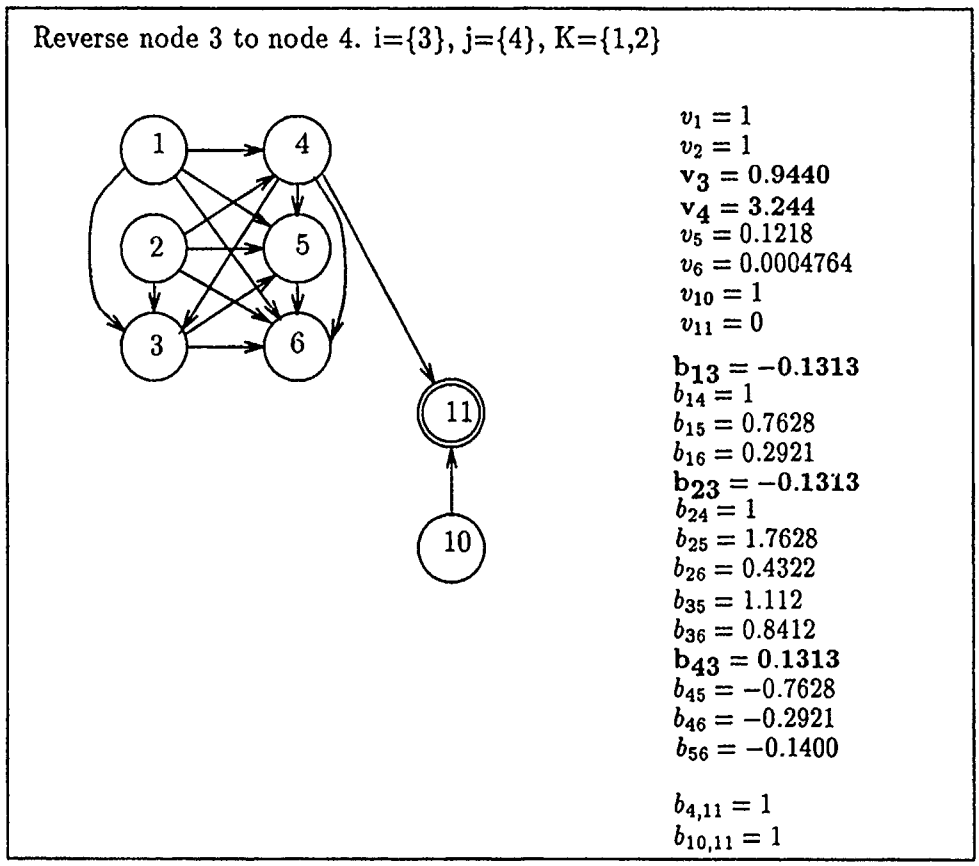


Figure 28. Example of Discrete-Time Filter with Arc Between Nodes 3 and 4 Reversed

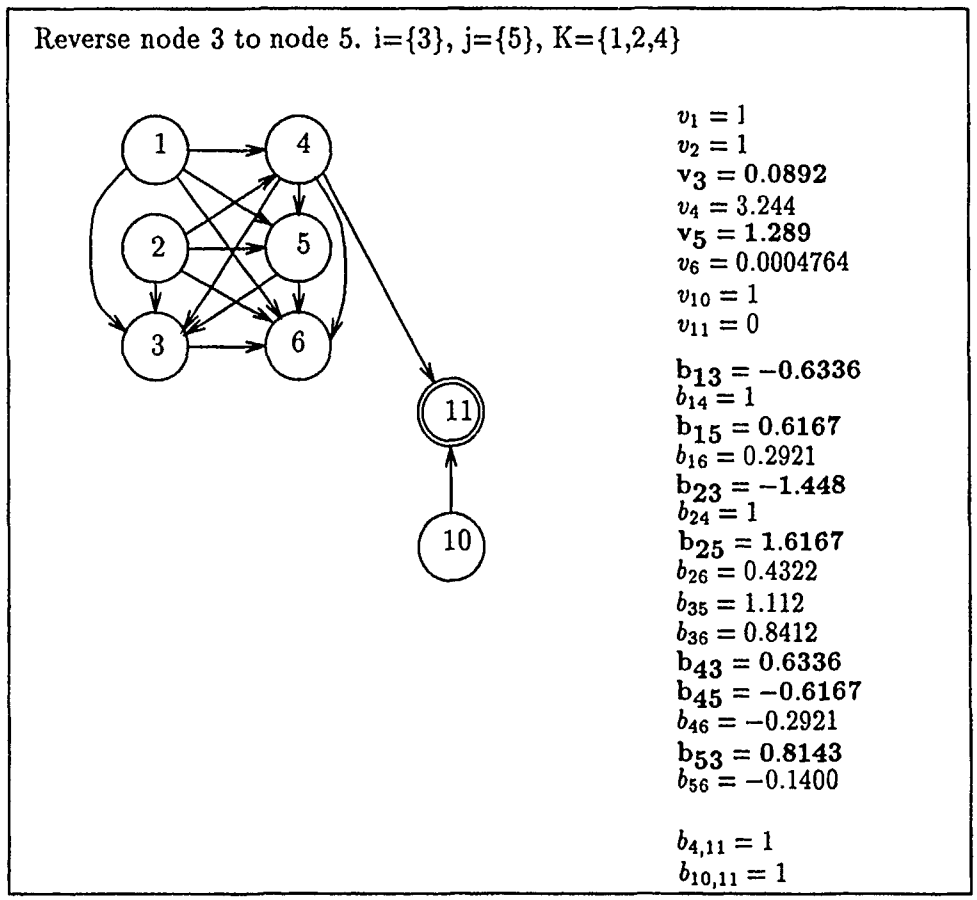


Figure 29. Example of Discrete-Time Filter with Arc Between Nodes 3 and 5 Reversed

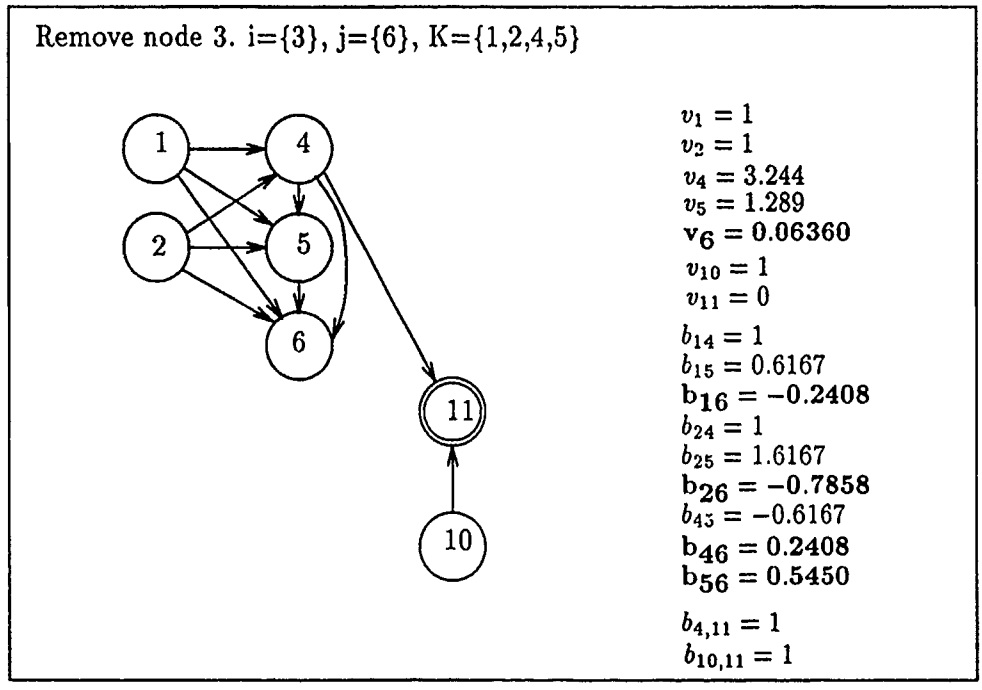


Figure 30. Example of Discrete-Time Filter with Node 3 Removed

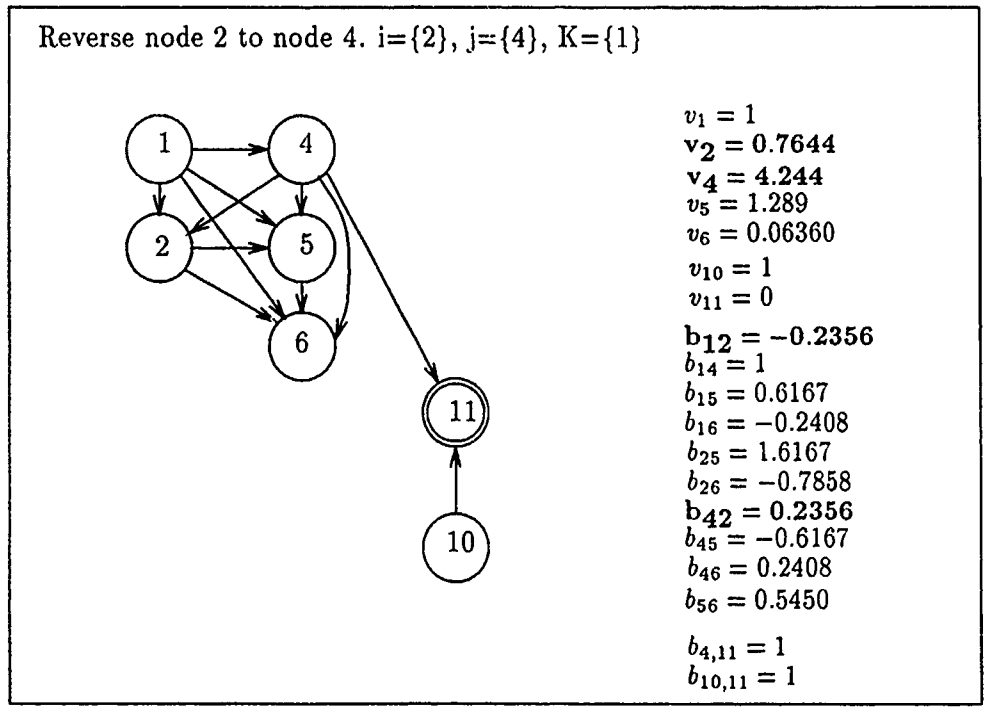


Figure 31. Example of Discrete-Time Filter with Arc Between Nodes 2 and 4 Reversed

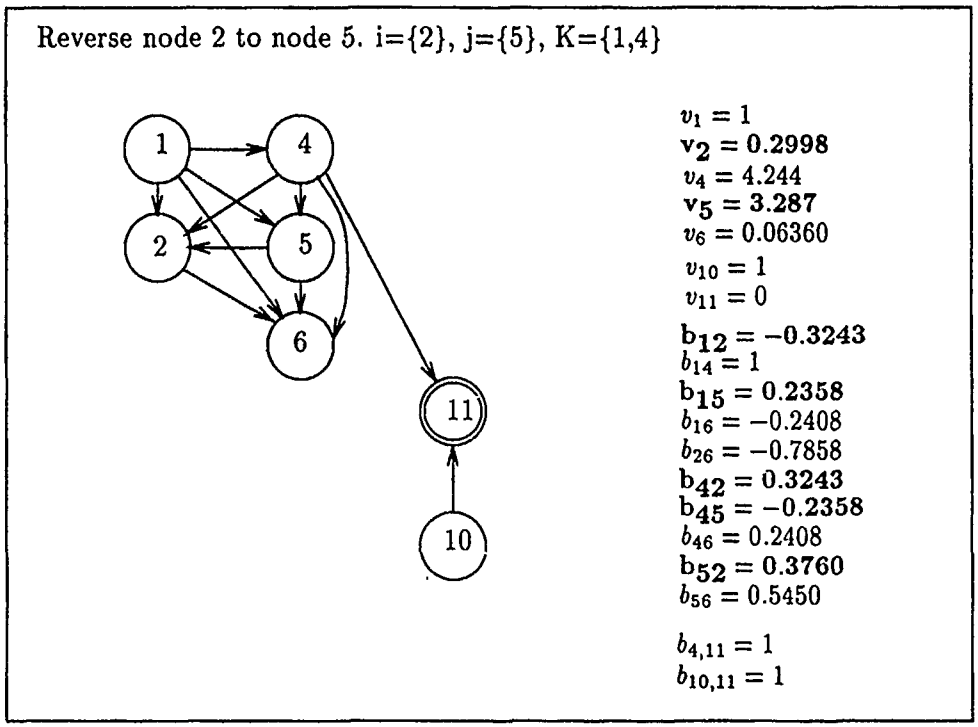


Figure 32. Example of Discrete-Time Filter with Arc Between Nodes 2 and 5 Reversed

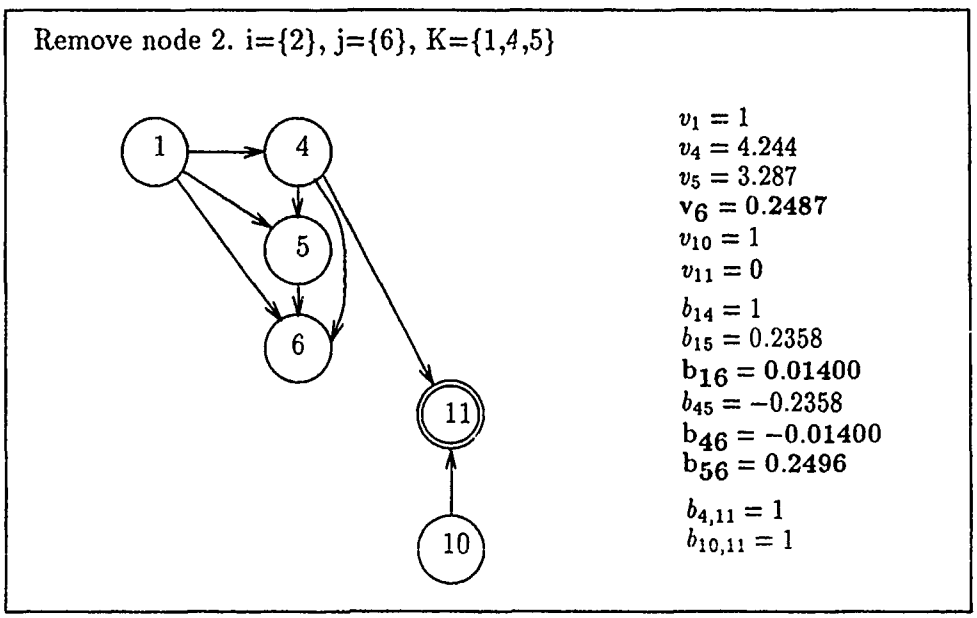


Figure 33. Example of Discrete-Time Filter with Node 2 Removed

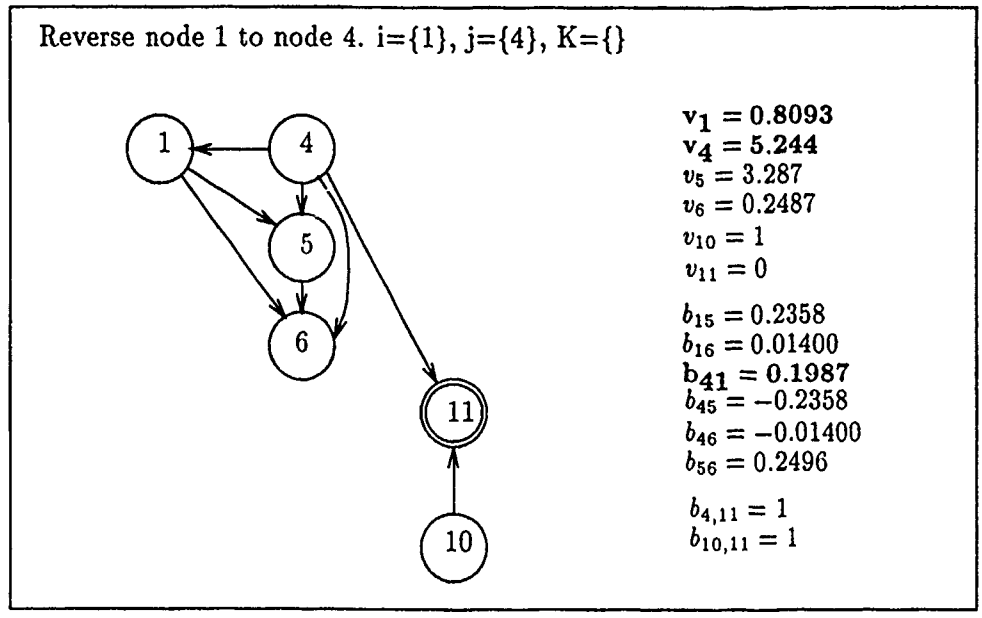


Figure 34. Example of Discrete-Time Filter with Arc Between Nodes 1 and 4 Reversed

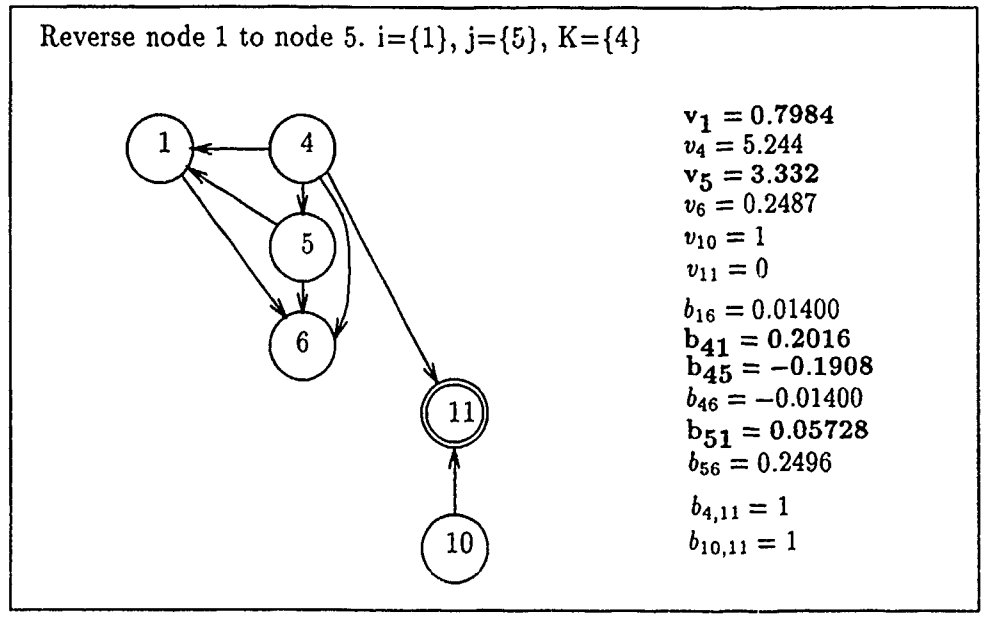


Figure 35. Example of Discrete-Time Filter with Arc Between Nodes 1 and 5 Reversed

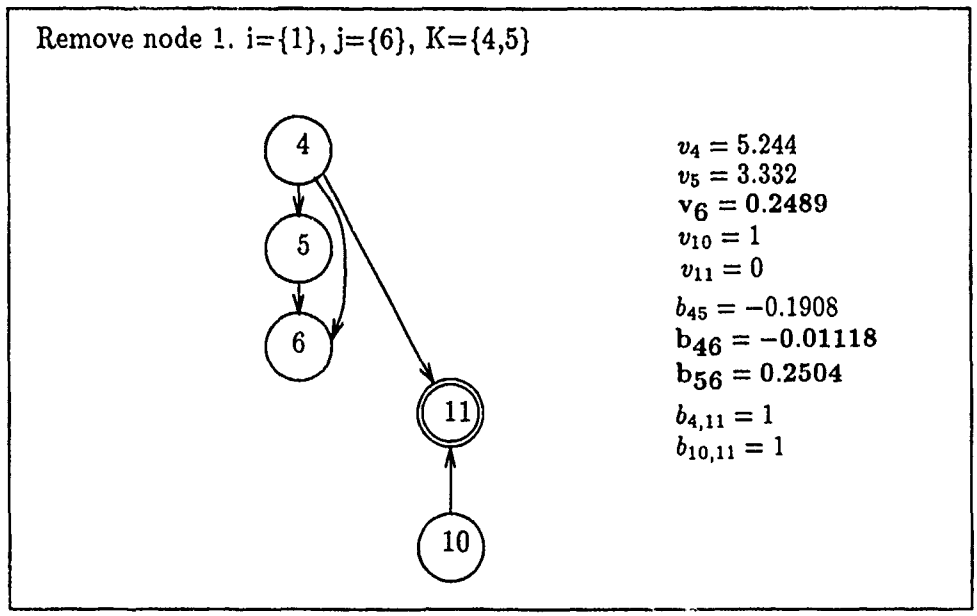


Figure 36. Example of Discrete-Time Filter with Node 1 Removed

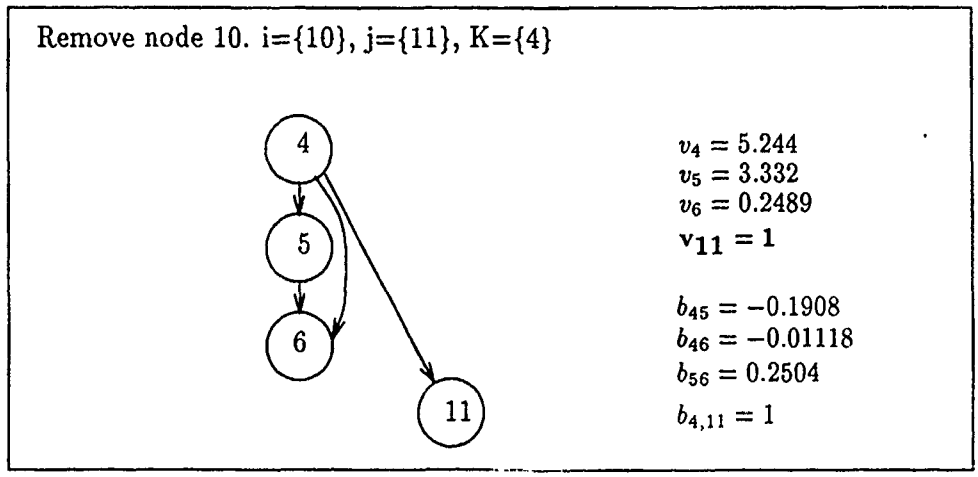


Figure 37. Example of Discrete-Time Filter with Node 10 Removed

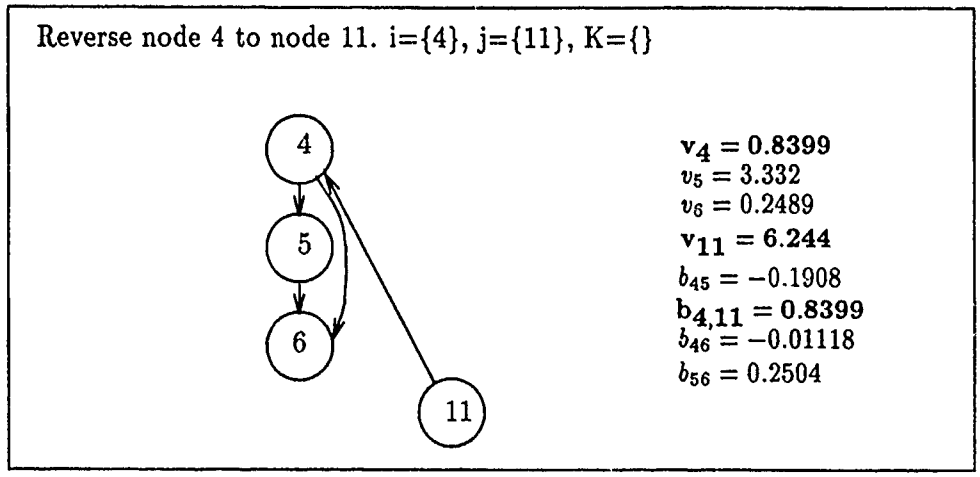


Figure 38. Example of Discrete-Time Filter with Arc Between Nodes 4 and 11 Reversed

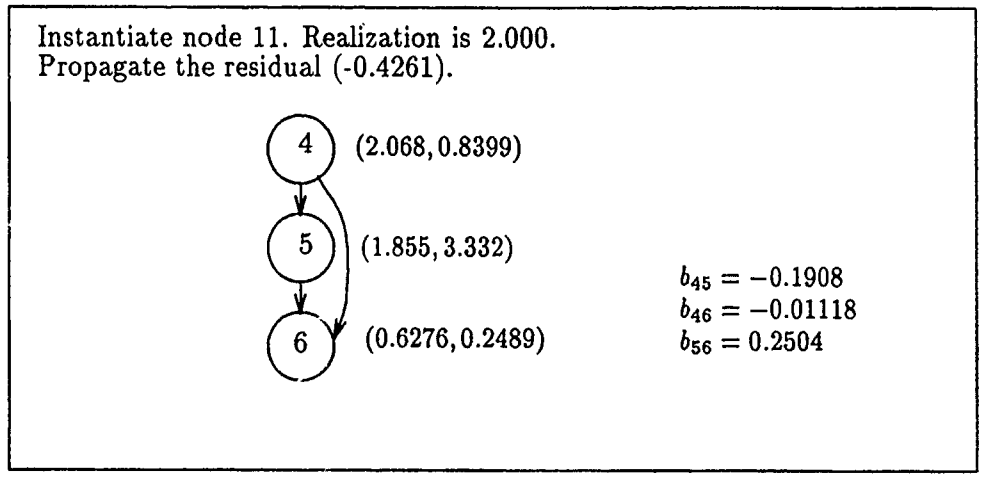


Figure 39. Example of Discrete-Time Filter with Node 11 Instantiated and Means Updated

III. Implementing the Influence Diagram

The previous chapter explained Kenley's implementation of the influence diagram. This chapter picks up on Kenley's original work and demonstrates three improvements to the original algorithm. One improvement is a solution to the infinite successor problem, mentioned in the previous chapter. This problem occurs because the influence diagram uses the terms of the **U** matrix to calculate the changes to the conditional means of successor nodes. When there are a large number of successors, **U** is very large and it becomes impractical to propagate the changes of the means.

A second improvement is a more efficient method of realizing a vector of measurements. In the previous chapter, the algorithm for updating the conditional means was a form of scalar update. When a node in unconditional form was realized, the residual was propagated to all successor nodes via the regression coefficients. After the realization, the original node was no longer needed and was instantiated (removed). After the first node was removed, the second node in the vector was in unconditional form, and it too could be realized and removed. The process continued, and the measurement nodes were removed one at a time, until all measurements were accomplished.

The third improvement is a demonstration that, under certain circumstances, the influence diagram can be more efficient than Kenley showed originally [6:pp. 89–106]. The conditions for this reduced operation count will be shown. Also, Kenley stated, but did not show, that the influence diagram can be used in a parallel processing architecture. This thesis will examine the implementation of the influence diagram in a parallel processing environment, and compare the improvement in processing time.

Before discussing these improvements, it will be useful to define two terms which will be used later in this thesis, the "path product" and the "path coefficient." These terms both represent scalar functions of the regression coefficients.

A path product will be defined as the product of the regression coefficients on a given path from a predecessor node to a successor node. A path coefficient will be defined as the sum of the path products on all possible paths from a predecessor node to a successor node. An example will demonstrate the use of these terms.

Consider an influence diagram of five nodes, numbered sequentially 1 through 5. The path products from node 1 to node 5 are:

$$b_{15} \tag{97}$$

$$b_{14}b_{45} \tag{98}$$

$$b_{13}(b_{35} + b_{34}b_{45}) \tag{99}$$

$$b_{12}[b_{25} + b_{24}b_{45} + b_{23}(b_{35} + b_{34}b_{45})] \tag{100}$$

The sum of these path products results in the path coefficient, which will be denoted by the term u_{15}.

In more general terms, let $u_{m,n}$ be the path coefficient from a node m to a successor node n, and let $b_{m,n}$ be the regression coefficient on the arrow from a node m to its direct successor node n. Another expression for $u_{1,r}$ is:

$$u_{1,r} = b_{1,r} + b_{1,r-1}(u_{r-1,r}) + b_{1,r-2}(u_{r-2,r}) \quad \ldots, b_{1,2}(u_{2,r}) \tag{101}$$

As was shown in Chapter 2, the linear change in the conditional mean of a successor node, conditioned on the realization of a predecessor, is equal to the path coefficient from the first to the second.

Let a positive definite, symmetric covariance matrix be factored into a unit lower triangular matrix, a diagonal matrix, and a unit upper triangular matrix equal to the transpose of the unit lower triangular matrix (a unit triangular matrix is defined to have ones on the main diagonal).

Such a factorization is expressed as $\mathbf{U}^T\mathbf{D}\mathbf{U}$ (or equivalently $\mathbf{L}\mathbf{D}\mathbf{L}^T$ where $\mathbf{U} = \mathbf{L}^T$) and is unique [13:pp. 133–143]. Using the first notation, the $\mathbf{U}$ matrix can be factored as a product of unit upper triangular as $\mathbf{U} = \mathbf{U}_1\mathbf{U}_2\mathbf{U}_3 \ldots \mathbf{U}_n$, where n is the dimension of U. Each matrix $\mathbf{U}_j$ is an identity matrix, plus nonzero terms only in the jth column above the diagonal. As discussed in Chapter 2, Kenley and Schacter showed that such a factorization was equivalent to an influence diagram representation of the covariance matrix [12:pp. 547–548]. The regression coefficients are the nonzero terms above the diagonal in the $\mathbf{U}_j$ matrix. Using the newly defined terminology, the path coefficients are the u_{ij} terms above the diagonal in the U matrix.

As an example, consider a 4-dimensional covariance matrix factored as described above. The matrix $\mathbf{D}$ is

$$\mathbf{D} = \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & 0 & 0 \\ 0 & 0 & v_3 & 0 \\ 0 & 0 & 0 & v_4 \end{bmatrix} \tag{102}$$

where $v_1, v_2, v_3, v_4 \neq 0$. The matrix $\mathbf{U}$ is expressed as

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & b_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & b_{13} & 0 \\ 0 & 1 & b_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & b_{14} \\ 0 & 1 & 0 & b_{24} \\ 0 & 0 & 1 & b_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{103}$$

or in terms of the path coefficients:

$$\mathbf{U} = \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{104}$$

The expression for the path coefficient, in terms of the regression coefficients as in Equation (101), can be verified by matrix multiplication. In the case of u_{14}, u_{24}, and u_{34} for example, the multiplication yields:

$$u_{34} = b_{34} \tag{105}$$

$$u_{24} = b_{24} + b_{23}b_{34} \tag{106}$$

$$u_{14} = b_{14} + b_{13}b_{34} + b_{12}(b_{24} + b_{23}b_{34}) \tag{107}$$

9.1 The Recursive Algorithm

The influence diagram algorithm has a problem when implementing the discrete-time filter over many, possibly infinitely many, time intervals. It makes the assumption that when a measurement is made, the residual will be propagated to all successors. In a recursive discrete-time filter with possibly an infinite number of successors, propagating the mean is not possible.

One implementation scheme avoids the infinite successor problem. Assume a vector of nodes labeled $\mathbf{x}(t_i)$ conditions another vector $\mathbf{z}(t_i)$ and the influence diagram contains no other nodes. Such an influence diagram was shown in Figure 17. Also assume that the vector $\mathbf{x}(t_i)$ represents the state estimate at time t_i, before incorporating the measurements at time t_i. The vector $\mathbf{z}(t_i)$ represents the measurement vector at the same time. As shown in the second diagram in Figure 17, the nodes of $\mathbf{z}(t_i)$ are moved to the beginning of the ordered sequence, and the random variables associated with those node are realized. The change in mean is propagated to all nodes of $\mathbf{x}(t_i)$, and the nodes of $\mathbf{z}(t_i)$ are removed. At this time, $\mathbf{x}(t_i)$ is conditioned on $\mathbf{z}(t_i)$, the actual measurements. If $\mathbf{x}(t_i)$ were followed by other vectors representing $\mathbf{x}(t_{i+1})$, $\mathbf{x}(t_{i+2})$, and so on, then the influence diagram requires that the residuals be propagated to these vectors as well. The operation is equivalent to calculating the conditional means of $\mathbf{x}(t_{i+1}), \mathbf{x}(t_{i+2}), \mathbf{x}(t_{i+3}), \ldots, \mathbf{x}(t_{i+n})$, conditioned on the measurements at time t_i. This prediction information is not normally desired.

Assume now that the residuals are not propagated to subsequent vectors. Then $\mathbf{x}(t_i)$ would have the correct conditional mean, but subsequent vectors would not. The state estimate at any time t_i will only be calculated when it is needed, normally at time t_i. The equation for doing this is the Kalman filter update equation:

$$\hat{\mathbf{x}}(t_i^-) = \Phi(t_i, t_{i-1})\hat{\mathbf{x}}(t_{i-1}^+) \tag{108}$$

The measurement estimates $\mathbf{z}(t_{i+1}), \mathbf{z}(t_{i+2}), \mathbf{z}(t_{i+3}), \ldots, \mathbf{z}(t_{i+n})$ are also successors of $\mathbf{x}(t_i)$ because they are successors of $\mathbf{x}(t_{i+1}), \mathbf{x}(t_{i+2}), \mathbf{x}(t_{i+3}), \ldots, \mathbf{x}(t_{i+n})$. The means could be propagated from $\mathbf{x}(t_i)$ to form the measurement estimates at these times, conditioned on the measurement at time t_i. This information is again not normally needed and such calculations are useless. The only estimate usually needed is the estimate (prediction) of the measurements at time t_i, conditioned on measurements through time t_{i-1}. This can be calculated similar to the Kalman filter equation as

$$\hat{\mathbf{z}}(t_i^-) = H(t_i)\hat{\mathbf{x}}(t_i^-) \tag{109}$$

To summarize this implementation method, the influence diagram should be used to calculate the conditional mean of the states at time t_i, based on the measurements at time t_i. However, when there are many successors, representing later time intervals, the influence diagram should not be used to calculate the conditional means for all of them. Instead, the conditional means for later states and measurements should be calculated only when needed using Equations (108) and (109) above.

Kenley's original work did not discuss deterministic inputs to the model. An equation that can represent deterministic inputs to the system is given by

$$\mathbf{x}(t_i) = \Phi(t_i, t_{i-1})\mathbf{x}(t_{i-1}) + \mathbf{G}_d(t_{i-1})\mathbf{w}_d(t_{i-1}) + \mathbf{B}_d(t_{i-1})\mathbf{u}_d(t_{i-1}) \tag{110}$$

where $\mathbf{u}_d(t_{i-1})$ is the deterministic input and $\mathbf{B}_d(t_{i-1})$ is the matrix that describes how the input affects the system states [7:220].

In the influence diagram, a deterministic input $\mathbf{u}_d(t_{i-1})$ can be modeled as a deterministic vector which is a predecessor of the $\mathbf{x}(t_i)$ vector. The arrows connecting $\mathbf{u}_d(t_{i-1})$ to $\mathbf{x}(t_i)$ are the $\mathbf{B}_d(t_{i-1})$ matrix. The mean of this deterministic vector is assumed to be the zero vector so that the actual inputs can modeled as a change of mean to the deterministic nodes, and then propagated. Using strict influence diagram rules, these changes of mean must be propagated to all subsequent nodes. Using the same logic as earlier though, propagating the conditional mean to later times is not useful. Instead, the matrix equation for the conditional mean at time t_i, including deterministic inputs at time t_{i-1} is

$$\hat{\mathbf{x}}(t_i^-) = \mathbf{\Phi}(t_i, t_{i-1})\hat{\mathbf{x}}(t_{i-1}^+) + \mathbf{B}_d(t_{i-1})\mathbf{u}_d(t_{i-1}) \tag{111}$$

3.2 Vector Measurement Update

When there is more than one random variable to be realized (measurement), then there are two methods for incorporating the measurements. In one case, the nodes are moved upwards and removed immediately as they become unconditional. In the other case, all nodes are moved upwards until the entire vector of measurements is unconditional (the nodes of the vector are conditioned only on other nodes in the vector), then all the measurements are incorporated and the measurement nodes removed.

There are advantages to each update method. If a node of the measurement vector is removed, then the arrows to subsequent nodes of the vector are not needed. As other nodes of the measurement vector are made unconditional, there are fewer predecessors and fewer calculations required. On the other hand, the change in mean at any node is the product of the change in mean of the predecessor (measurement node) time the path coefficient between the two. Thus, propagating the mean requires calculating the path coefficients from each node in the measurement vector

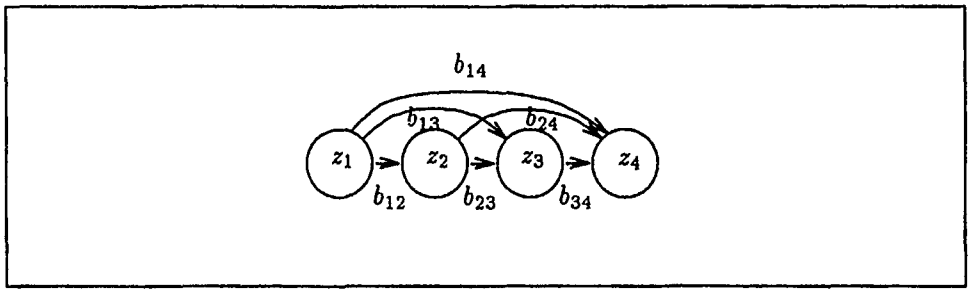


Figure 40. Measurement at Two Nodes Propagated to Two Successor Nodes

to each successor node. This process is equivalent to recalculating the matrix of path coefficients $\mathbf{U} = \mathbf{U}_1\mathbf{U}_2\mathbf{U}_3 \ldots \mathbf{U}_n$ for every measurement made. For an n-dimensional matrix, recalculating U requires $1/6(n^3 - 3n^2 + 2n)$ additions and multiplies.

There is a much more efficient method of updating the means which is of the order n^2. It is equivalent to a vector update and it requires the entire measurement vector in unconditional form. It does not explicitly calculate the path coefficients, but calculates the change of mean directly. It will be demonstrated by example.

Assume four nodes of a vector z in an influence diagram as shown in Figure 40. The first two nodes will be updated by a measurement. The second two are conditioned on the first two. The means will be propagated as the measurements are made. Also assume μ_i is the mean of the ith node before the update, and ζ_i is the measured value for i=1,2. If a single prime indicates the updated mean after only ζ_1 is propagated, then let r_1 be the calculated residual and use the conventional method:

$$\mu_2' = \mu_2 + b_{12}(\zeta_1 - \mu_1) \tag{112}$$

$$= \mu_2 + u_{12}r_1 \tag{113}$$

$$\mu_3' = \mu_3 + (b_{13} + b_{12}b_{23})(\zeta_1 - \mu_1) \tag{114}$$

$$= \mu_3 + u_{13}r_1 \tag{115}$$

$$\mu_4' = \mu_4 + [b_{12}(b_{24} + b_{23}b_{34}) + b_{13}b_{34} + b_{14}](\zeta_1 - mu_1) \quad (116)$$

$$= \mu_4 + u_{14}r_1 \quad (117)$$

Let a double prime represent the mean after the second update, that is the measurement ζ_2. Similarly, r_2' is the residual $(\zeta_2 - \mu_2')$. This residual uses μ_2', the updated mean of node two. But the updated mean is just the mean of node 2 conditioned on node 1 having a mean of ζ_1. Call the term r_2' the conditional residual. It is this conditional residual which is propagated as follows:

$$\mu_3'' = \mu_3' + b_{23}(\zeta_2 - \mu_2') \quad (118)$$

$$= \mu_3' + u_{23}r_2' \quad (119)$$

$$\mu_4'' = \mu_4' + (b_{24} + b_{23}b_{34})(\zeta_2 - \mu_2') \quad (120)$$

$$= \mu_4' + u_{24}r_2' \quad (121)$$

After simplifying the expressions for μ_4'' and μ_3'' in terms of the unprimed variables, the following equations result:

$$\mu_3'' = b_{13}(\zeta_1 - \mu_1) + b_{23}(\zeta_2 - \mu_2) + \mu_3 \quad (122)$$

$$\mu_4'' = (b_{14} + b_{13}b_{34})(\zeta_1 - \mu_1) + (b_{24} + b_{23}b_{34})(\zeta_2 - \mu_2) + \mu_4 \quad (123)$$

An alternative form for μ_4'' can be calculated be assuming node 3 had a measurement as well. Let this new measurement be ζ_3 and calculate using the mean of node 3 conditioned on both previous measurements. The result appears as:

$$\mu_4'' = b_{14}(\zeta_1 - \mu_1) + b_{24}(\zeta_2 - \mu_2) + b_{34}(\zeta_3 - \mu_3) + \mu_4 \quad (124)$$

By letting the assumed ζ_3 be the previously calculated $\mu 3''$, the previous expression for μ_4'' results.

The insight is that the new mean of node 3 (propagated from previous nodes) can be treated as a measurement at node 3, and the resulting residual propagated to the next node. The same effect can be extended to node 4. The new mean of node 4, calculated by propagating the change of mean from its predecessors, can be treated as a measurement at node 4, and the resulting residual propagated to the next node. The argument extends by induction to any number of nodes. For a change of mean that is not the true realization of the random variable, such as for nodes 3 and 4, the variance is not set to zero.

To find a useful algorithm, use the term ζ_i to refer to the new mean of node i after all updates, including possible updates to node i itself. Use the term r_i to be the residual $(\zeta_i - \mu_i)$ where μ_i is the mean before any updates. The first p nodes are the actual measurements, and they represent p realizations of random variables. For each of these nodes, the residual is calculated:

$$for\ i = 1\ to\ p \quad r_i = (\zeta_i - \mu_i) \tag{125}$$

The change of mean (the residual) and the mean itself (the assumed measurement) for node $p+1$ is calculated by:

$$r_{p+1} = b_{1,p+1}r_1 + b_{2,p+1}r_2 + b_{3,p+1}r_3 +, \ldots, b_{p,p+1}r_p \tag{126}$$

$$\zeta_{p+1} = r_{p+1} + \mu_{p+1} \tag{127}$$

The residual for node p+1 is propagated to node p+2 just as if it were the result of a measurement. The equation for this propagation and the resulting mean is:

$$r_{p+2} = (b_{1,p+2}r_1 + b_{2,p+2}r_2 + b_{3,p+2}r_3 +, \ldots, b_{p,p+2}r_p) + (b_{p+1,p+2}r_{p+1}) \tag{128}$$

$$\zeta_{p+2} = r_{p+2} + \mu_{p+2} \tag{129}$$

The process continues with each new residual treated as a measurement to be propagated to the next node. The term in parenthesis on the second line of Equation (128) is the residual from the assumed measurement created by propagating to the p+1 node. The calculations for the last, or n+p node, are:

$$\begin{aligned} r_{p+n} &= (b_{1,p+n}r_1 + b_{2,p+n}r_2 + b_{3,p+n}r_3 +, \ldots, b_{p,p+n}r_p) + \\ &\quad (b_{p+1,p+n}r_{p+1} + b_{p+2,p+n}r_{p+2}, \ldots, b_{p+n-1,p+n}rp + n - 1) \qquad (130) \\ \zeta_{p+n} &= r_{p+n} + \mu_{p+n} \qquad (131) \end{aligned}$$

These calculations can be simplified to a series of matrix operations. If there are p nodes to be measured, with a mean propagated to n successor nodes, then the residuals for the successor nodes $p+1$ to $p+n$ is given by:

$$\begin{bmatrix} r_{p+1} \\ r_{p+2} \\ \vdots \\ r_{p+n} \end{bmatrix} = \left[\begin{array}{cccc|cccc} b_{1,p+1} & b_{2,p+1} & \cdots & b_{\rho,p+1} & 0 & 0 & \cdots & 0 \\ b_{1,p+2} & b_{2,p+2} & \cdots & b_{p,p+2} & b_{p+1,p+2} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{1,p+n} & b_{2,p+n} & \cdots & b_{p,p+n} & b_{p+1,p+n} & b_{p+2,p+n} & \cdots & b_{p+n-1,p+n} \end{array}\right] \left[\begin{array}{c} r_1 \\ r_2 \\ \vdots \\ r_p \\ \hline r_{p+1} \\ r_{p+2} \\ \vdots \\ r_{p+n-1} \end{array}\right] \qquad (132)$$

The matrix operation shown above implies that the update uses a single matrix. The true operation is the sum of a series of vector inner products. The matrix is the transpose of the $p+1$ through $p+n$ columns of the **B** matrix. The first p elements of the rightmost residual column vector are partitioned as the residuals from the measurement vector. The result of each inner product operation is augmented to the residual column vector as a new residual and used in the next inner

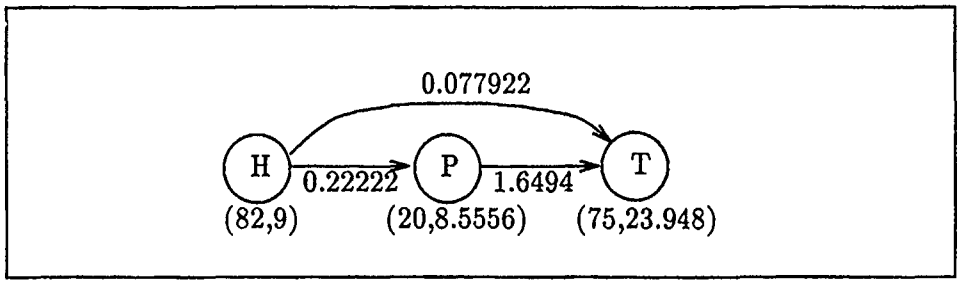


Figure 41. Example of Continuous Gaussian Influence Diagram

product calculation. The total number of operations required for these calculations is $\frac{1}{2}n^2 - \frac{1}{2}n + pn$ adds and multiplies. The effect is comparable to performing a vector update instead of sequential scalar updates. An example will demonstrate the algorithm.

In the example given in Chapter 2, the conditional mean of the playing time in a basketball game was expressed in influence diagram form, conditioned on the player's height and the number of points scored. The diagram is repeated here in Figure 41. The conditional mean of the playing time was calculated as the update of the two scalar measurements of height and points scored per game. First, the residual of the measured height was propagated to subsequent nodes and the height node was instantiated. Then the residual of the realized number of points per game was propagated to the remaining node. The conditional mean of playing time, given an 84 inch player was 75.8889%. If an 84 inch player scored only 16 points per game, then the conditional mean of the playing time, conditioned on both measurements, was 68.558%.

Both of these conditional means could be calculated by the vector update algorithm. For the first example, the average playing time will be calculated based on only the player's height. The equations are:

$$b_{12}r_1 = r_2 \quad r_2 + \mu_2 = \zeta_2 \tag{133}$$

$$b_{13}r_1 + b_{23}r_2 = r_3 \quad r_3 + \mu_3 = \zeta_3 \tag{134}$$

or in numerical terms:

$$0.2222(84-82)=0.4444 \quad 0.4444+20=20.4444 \tag{135}$$

$$0.077922(84-82)+1.6494(0.4444)=0.8889 \quad 0.8889+75=75.8889 \tag{136}$$

The conditional mean for both the number of points scored and the percentage of playing time are the same as calculated earlier. These equations use the residual r_2. This is not a true residual because there is no measurement of the points per game. This number is treated as a residual only to make the algorithm work.

As a second example, assume the measured height and the number of points scored will be used as a measurement vector. where the dimension of the vector is p=2. The update equations are now:

$$b_{13}r_1+b_{23}r_2=r_3 \quad r_3+\mu_3=\zeta_3 \tag{137}$$

or in numerical terms:

$$0.077922(84-82)+1.6494(16-20)=-6.4417 \quad -6.4417+75=68.558 \tag{138}$$

Again, the vector update equations yield the same results as the earlier example. The average points per game, conditioned on the player's height, was not calculated. There was no need to calculate it because it was part of the measurement vector and it would take on the measured value no matter what its conditional mean was.

3.3 Efficient Implementation Form

Kenley demonstrated that the influence diagram was competitive with the U-D filter in terms of the number of required mathematical operations [6:pp. 52–106]. He used a form of the influence

diagram as was shown in Figure 18. In that diagram, the nodes of $\mathbf{w}_d(t_{i-1})$ have no arrows between them, implying they are independent. This is equivalent to assuming the matrix product $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ has been factored such that $\mathbf{Q}_d(t_{i-1})$ is a diagonal matrix.

The example in Chapter 2 of this thesis used an influence diagram implementation of the discrete-time filter with different assumptions about the factorization of $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$. In Figure 21, $\mathbf{w}_d(t_{i-1})$ was expressed as an influence diagram factorization of the matrix product $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ and $\mathbf{G}_d(t_{i-1})$ was assumed to be the identity matrix. The general case of discrete-time filter, assuming $\mathbf{G}_d(t_{i-1}) = \mathbf{I}$ is shown in Figure 42. The influence diagrams in Figure 18 and Figure 42 are different representations of the same discrete-time model. When $\mathbf{w}_d(t_{i-1})$ is removed, both diagrams reduce to the same influence diagram shown in Figure 43.

Because of the identity matrix transformation, and the fact that $\mathbf{x}(t_i)$ is a deterministic function of $\mathbf{w}_d(t_{i-1})$, the removal of $\mathbf{w}_d(t_{i-1})$ results in a simple form. The conditional variances of the nodes of $\mathbf{x}(t_i)$ in Figure 43 are the same as the conditional variances of the nodes of $\mathbf{w}_d(t_{i-1})$ in Figure 42. Furthermore, the regression coefficients between the nodes of $\mathbf{x}(t_i)$ in Figure 43 are the same as the regression coefficients between the nodes of $\mathbf{w}_d(t_{i-1})$ in Figure 42. A demonstration of this can be seen by comparing the appropriate numbers for the example in Chapter 2, Figure 21 and Figure 27. In those figures, the variances and regression coefficients of $\mathbf{w}_d(t_{i-1})$ of Figure 21 appear to transfer directly to the nodes of $\mathbf{x}(t_i)$ in Figure 27.

A further comparison of these example diagrams reveals that the regression coefficients corresponding to the matrix $\Phi(t_i, t_{i-1})$ in Figure 21 are not the same as the comparable regression coefficients in Figure 27. The new regression coefficients are no longer the elements of the state transition matrix $\Phi(t_i, t_{i-1})$, but have been modified because of the influence diagram operations between Figure 21 and Figure 27. These new regression coefficients can be thought of as being the

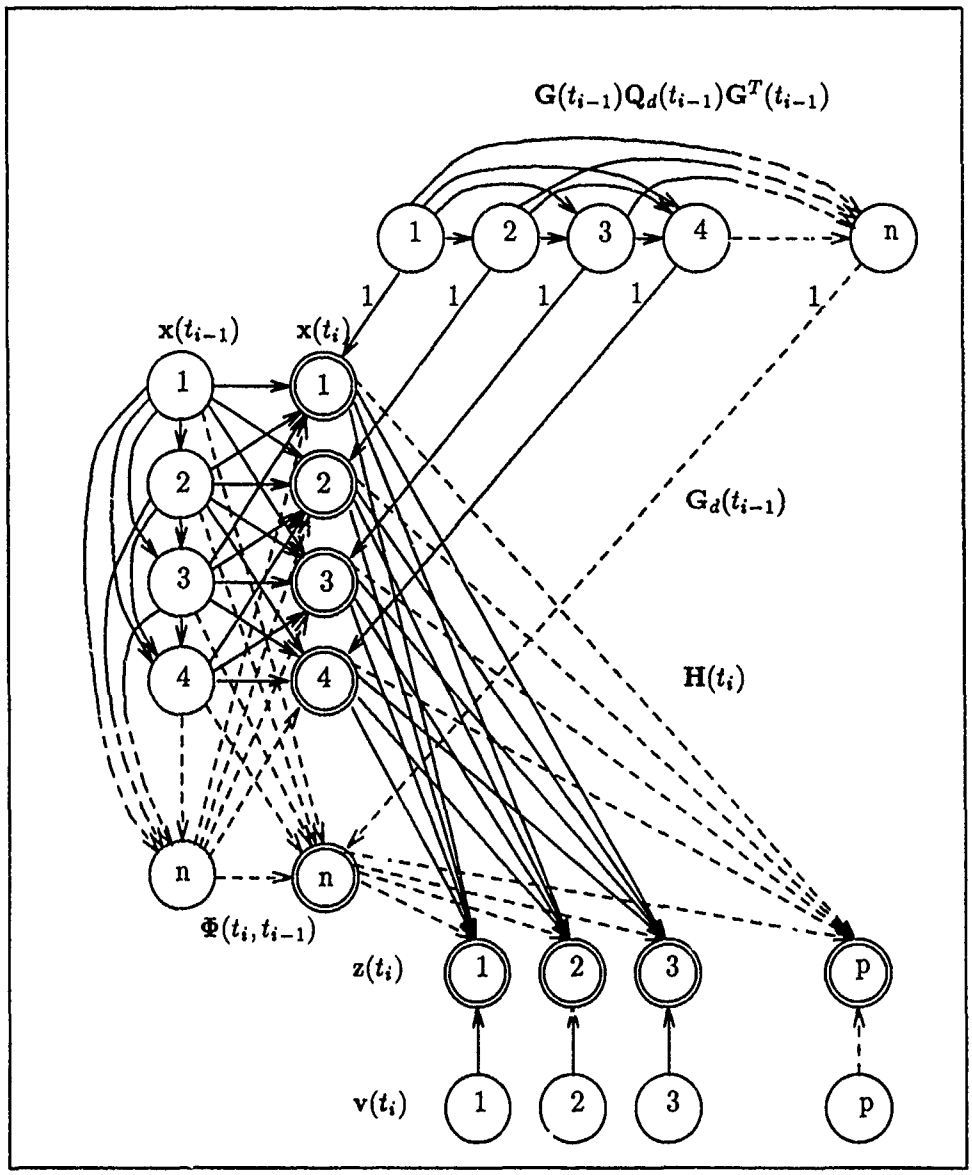


Figure 42. Alternative Representation of One Propagation/Update Cycle of the Influence Diagram Discrete Time Filter

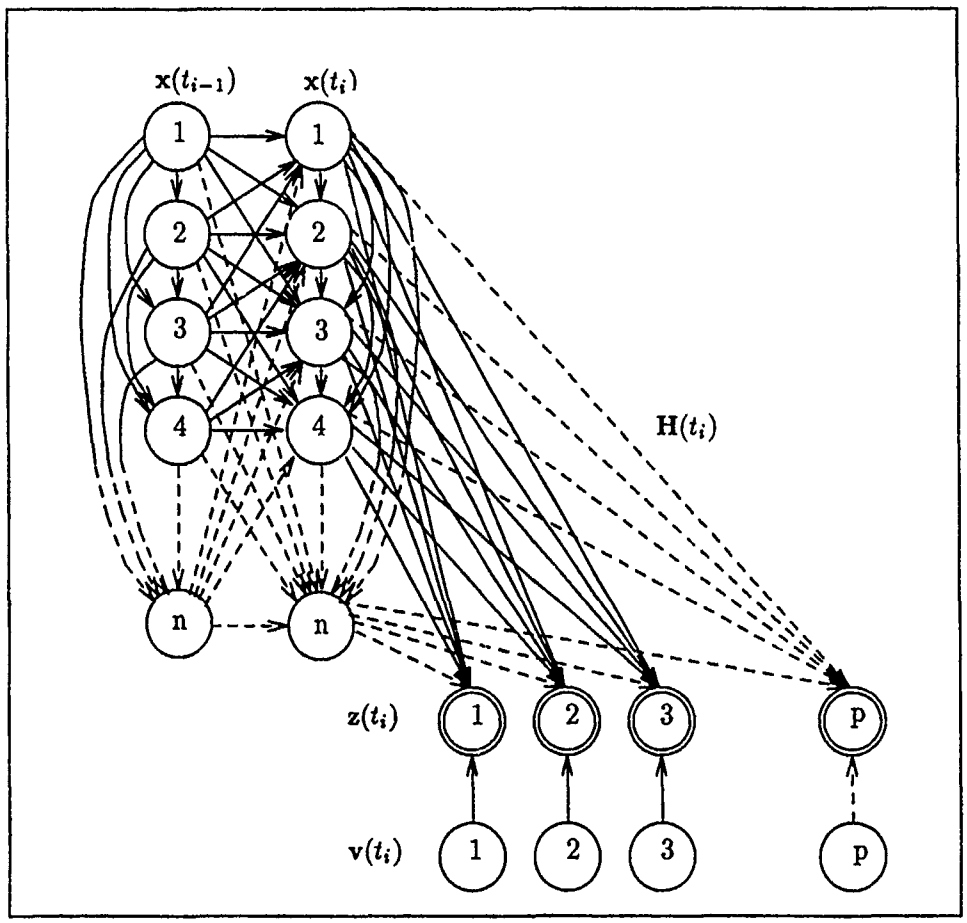


Figure 43. Influence Diagram Discrete-Time Filter with $\mathbf{w}_d(t_{i-1})$ Removed

elements of a new matrix. Call this equivalent matrix Φ_q. The elements of Φ_q can be calculated as

$$\Phi_q = (\mathbf{I} - \mathbf{B}_q)^T \Phi(t_i, t_{i-1}) \tag{139}$$

where $\mathbf{B}_q$ is the influence diagram $\mathbf{B}$ matrix corresponding to the influence diagram factorization of $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$, and $\mathbf{I}$ is the identity matrix of appropriate dimension. In the previous example, the Φ_q can be calculated as:

$$\Phi_q = \{\mathbf{I} - \begin{bmatrix} 0 & -0.7628 & -0.2921 \\ 0 & 0 & -0.1400 \\ 0 & 0 & 0 \end{bmatrix}\}^T \begin{bmatrix} 1 & 1 & 0.4261 \\ 0 & 1 & 0.7870 \\ 0 & 0 & 0.6065 \end{bmatrix} \tag{140}$$

$$\Phi_q = \begin{bmatrix} 1 & 1 & 0.4261 \\ 0.7628 & 1.7628 & 1.112 \\ 0.2921 & 0.4322 & 0.8412 \end{bmatrix} \tag{141}$$

These are seen to be the same as the regression coefficients in Figure 27.

One way to understand these changes in the regression coefficients corresponding to the $\Phi(t_i, t_{i-1})$ matrix is as follows. The total effect of all regression coefficients from a node of $\mathbf{x}(t_{i-1})$ to a node of $\mathbf{x}(t_i)$ is equivalent to the appropriate element of the $\Phi(t_i, t_{i-1})$ matrix. This total effect is the change in the conditional mean of $\mathbf{x}(t_i)$, given a realization of the predecessor nodes in $\mathbf{x}(t_{i-1})$. This total effect does not change when the nodes of $\mathbf{x}(t_i)$ are conditioned on one another. As the regression coefficients are added between the nodes of $\mathbf{x}(t_i)$, the coefficients from $\mathbf{x}(t_{i-1})$ must be changed to compensate.

The previous analysis demonstrates a significant reduction in computations when the matrix product $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is given in influence diagram form. Under these circumstances, all influence diagram operations needed to remove $\mathbf{w}_d(t_{i-1})$ are unnecessary. Instead, an influence

diagram as was shown in Figure 43 can be drawn immediately, with the regression coefficients from $\mathbf{x}(t_{i-1})$ modified as in Equation (139). In the example of Chapter 2, this would have been equivalent to skipping from Figure 21 to Figure 27 without needing to go through the intermediate equations. The author is not aware of any such operations savings with other forms of the Kalman filter when $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is available in a factored form.

3.4 Operations Count

An important characteristic of the influence diagram algorithm is the number of floating point operations required for implementation on a digital computer. Kenley calculated the number of operations necessary for updating the variances, and compared them with the operations counts for other filter implementations [6:pp. 89–106]. In that comparison, he assumed a filter of the form in Figure 18. He demonstrated that the U-D filter and the influence diagram require a similar number of operations.

Kenley's original operations count only addressed the computations required for the update of the variances, not the state estimates. If the influence diagram state estimates are updated using the vector update method from the previous section, then the influence diagram remains equivalent to the U-D filter in terms of computational efficiency.

The influence diagram has a significant advantage when $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is available in influence diagram form. However, as was shown in Chapter 2, the influence diagram form of the matrix product $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is the same as a factorization of the form $\mathbf{U}_n^T \ldots \mathbf{U}_3^T\mathbf{U}_2^T\mathbf{U}_1^T\mathbf{D}\mathbf{U}_1\mathbf{U}_2\mathbf{U}_3 \ldots \mathbf{U}_n$ where n is the dimension of the matrix product. It will be shown later, in Chapter 4, that such a factorization is very similar to the $\mathbf{U}\mathbf{D}\mathbf{U}^T$ factorization used in the U-D form of the Kalman filter. Because of this similarity, it will be assumed that a covariance matrix can be expressed in influence diagram form just as efficiently as it can be expressed in U-D factored form.

Table 4 compares the conventional Kalman filter, the U-D filter, and the influence diagram. The Kalman filter implementation assumes $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is available as a single matrix. The U-D filter form assumes that $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is in U-D factored form such that $\mathbf{G}_d$ is equal to the upper triangular $\mathbf{U}$ factor and $\mathbf{Q}_d$ is equal to the diagonal $\mathbf{D}$ factor. The influence diagram implementation assumes $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is expressed in influence diagram form as in Figure 42. The only operations needed to transform the influence diagaram from a form as in Figure 42 to a form as in Figure 43 are given in Equation (139).

The number of operations required for the influence diagram implementation is calculated in Appendix A. The number of operations for the Kalman filter and the U-D filter are reproduced from Maybeck [7:403] using the same assumptions about $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ as in the previous paragraph. For a specific example, Table 5 represents the execution time for a typical discrete-time filtering problem with a 10-dimensional state vector and a 2-dimensional measurement vector. The execution times come from Maybeck [7:404]. These tables show that the influence diagram significantly exceeds the U-D filter in speed.

Under certain conditions, the matrices $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ and $\mathbf{\Phi}(t_i, t_{i-1})$ may be constant from one time interval to the next. Under these conditions, all terms on the right side of Equation (139) are unchanged from one time interval to the next and there is no need to recompute $\mathbf{\Phi}_q$. The operations count for the influence diagram will require $n(n-1)$ fewer additions and multiplications if Equation (139) is unnecessary. Using the assumptions of Table 5, this equates to $(2.7 + 4.1)45$ or 306 fewer microseconds per cycle.

3.4.1 Pipeline Processing. Kenley mentions that the influence diagram lends itself to parallel processing because reversing the conditioning on two nodes results in changes that are isolated within the diagram. A specific example shown in Figure 44 demonstrates the potential for parallel processing.

Table 4. Operations for One Time Propagation and One Measurement Update

Filter	Adds	Multiplies	Divides
Conventional Kalman	$\frac{1}{2}(3n^3+3n^2p+$ $5np-n)$	$\frac{1}{2}(3n^3+3n^2(p+1)+$ $3np)$	m
U-D	$\frac{1}{2}(5n^3+n^2(3p+2)+$ $n(3p+1))$	$\frac{1}{2}(5n^3+n^2(3p+11)+$ $n(p-6))$	$n(p+1)-1$
Influence Diagram	$2n^3+n^2(p-0.5)+$ $n(p^2+p-0.5)$	$2n^3+n^2(p+3.5)+$ $n(p^2+5p-1.5)$	$n(n+p-1)$

Assumptions: $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is available as a single matrix,
for the Kalman filter, in U-D factored form for the U-D filter,
or in influence diagram form for the influence diagram.
State and dynamics driving noise dimension = n
Measurement dimension = p

Table 5. Operation Time for One Filter Recursion

Filter	Adds	Multiplies	Divides	Time (msec)
Conventional Kalman	1845	2040	2	13.36
U-D	2935	3330	29	21.77
Influence Diagram	2205	2675	110	17.65

Assumptions: $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ known as in previous table.
State and dynamics driving noise dimension = 10
Measurement dimension = 2
The execution time per cycle assumes each operation requires:
2.7 μsec per addition
4.1 μsec per multiplication
6.6 μsec per division

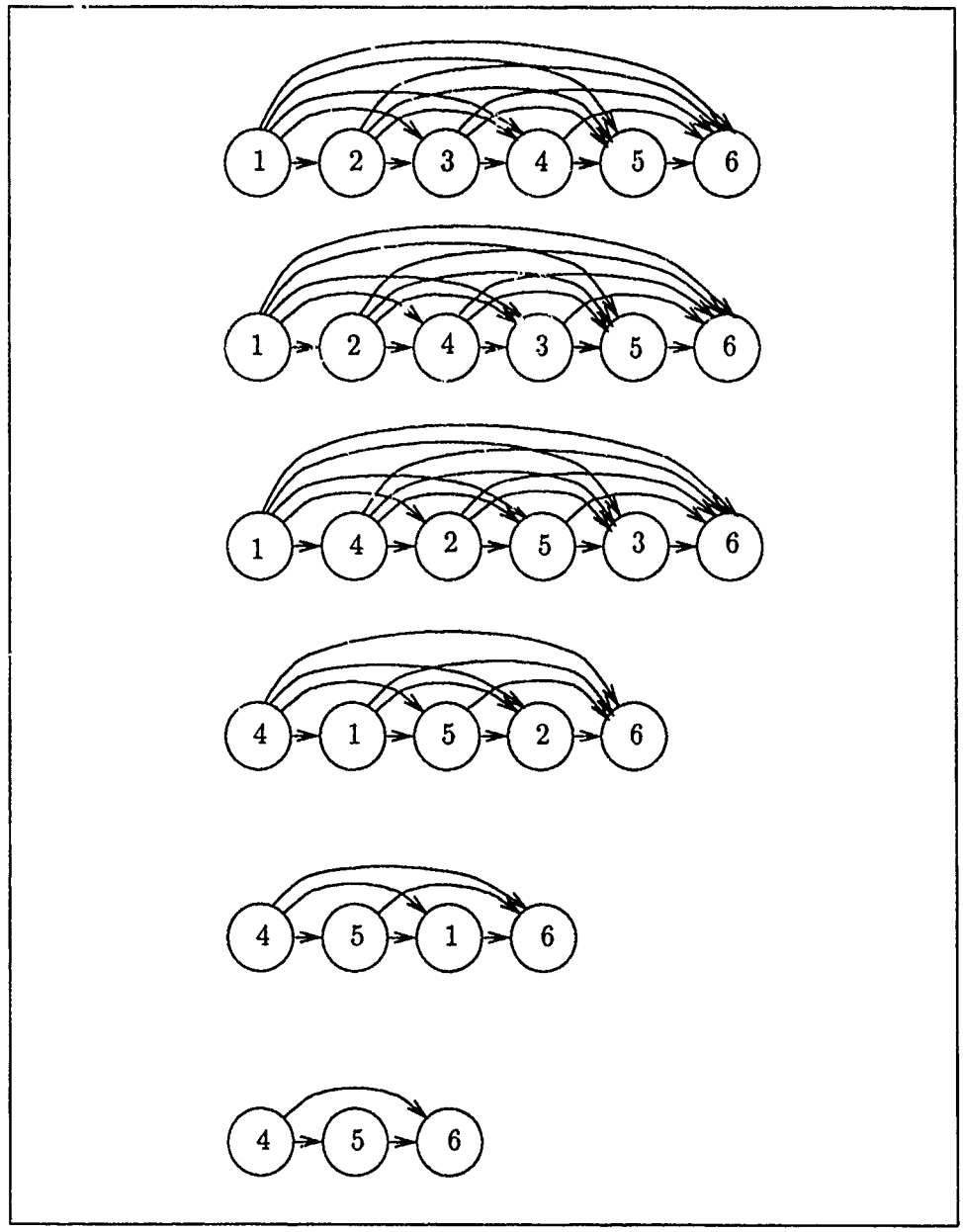


Figure 44. Pipeline Implementation of the Removal of Three Nodes

Assume the first three nodes are to be moved to the right and eliminated from the diagram. The third node is removed first because it is the most conditional. The arc between it and the succeeding node is reversed resulting in the second influence diagram in Figure 44. The next operation could be either moving node 3 further to the right, or move node 2 to the right. The predecessor values modified in one operation are distinct from those modified in the other operation. As a result, both operations could be done simultaneously in separate processors. This simultaneous reversal is shown in the third diagram in Figure 44. The next node can be moved to the right at the same time as the first two move further rightwards as in the fourth diagram. The process continues until all three nodes are removed.

Assume that there is a single processor dedicated to the task of calculating all necessary equations for each of the n nodes being moved rightwards. This is only one possible way of applying parallel processing to the influence diagram algorithm, but it does serve as a basis for comparison. In this example, there would be three processors, one assigned to each of the first three nodes. Between the first and second influence diagrams, a single processor would reverse nodes 3 and 4. Between the second and third diagrams, one processor would be used to exchange nodes 3 and 5, while another would be used to exchange 2 and 4. Moving from the third to the fourth diagram requires all three processors; one processor removes node 3, another reverses nodes 2 and 5, while the third reverses nodes 1 and 4. Moving to the fifth diagram requires only two processors, one to remove node 2 and another to reverse nodes 1 and 5. Finally, only a single processor is needed to remove node 1 and result in the last influence diagram.

Under these assumptions, the computation time for each cycle is reduced greatly. As an example, assume the same operations counts as in Table 4 and Table 5. In this case, also assume that there are n separate processors, each dedicated to the task of moving one node rightwards as in Figure 44. The computation time for such a configuration is given in Table 6. As in Table 5, the time can be reduced by another 306 microseconds if there is no need to recompute Φ_q. The

Table 6. Operations Counts and Times for Influence Diagram, Pipeline Architecture

Adds	Multiplies	Divides	Time (msec)
$9n^2 + n(6p-19) +$	$9n^2 + 6n(p-1) +$	$3n + p - 3$	-
$p^2 - 4p + 11$	$p^2 + 1$		-
837	965	29	6.41

Assumptions: $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ is known beforehand,
in either matrix or influence diagram factored form.
State and dynamics driving noise dimension = 10
Measurement dimension = 2
The execution time per cycle assumes each operation
requires the following execution times:
addition=2.7 μsec
mutiplication = 4.1 μsec
division=6.6 μsec

operations counts in Table 6 are developed in Appendix B. The reason that the time is not reduced by a factor of n in Table 6 is because there is idle time for each processor as it waits for the previous processors to accomplish their tasks sufficiently to begin operation itself. Idle time can be reduced by doing other operations (such as calculating Equation (139)).

9.5 Chapter Summary

This chapter demonstrated three different methods of improving the implementation of the influence diagram for discrete-time filtering. These improvements make the influence diagram more efficient than the U-D factored form of the Kalman filter under the conditions given. For the special case where Equation (139) does not need to be calculated for each time interval, the influence diagram is even faster. This chapter concluded with a demonstration of the influence diagram implemented in parallel processing form. In this case, the influence diagram was shown to be very fast indeed.

IV. Equivalent Operations

The influence diagram's relationships to matrix operations, as previously shown by Kenley, were described in Chapter 2 of this thesis. These matrix operations will be a key to numerical analysis later in Chapter 5, so this chapter is dedicated to a detailed analysis of the influence diagram operations. The intent is to relate the influence diagram operations to well understood arithmetic and matrix operations. The analysis will occur in three parts. The first part will be a demonstration of the equivalence of matrix products and node reversal operations. The second part will be a demonstration of a matrix equivalent to the discrete-time filter shown in Chapter 2. The third part will be a direct comparison of the influence diagram and the U-D algorithm for discrete-time filtering.

4.1 Equivalence to Matrix Operations

The most important operation in influence diagram manipulation is node reversal. It is equivalent to changing the conditioning order of a pair of random variables while maintaining the joint distribution of the two. For the case of Gaussian random variables, only the conditional and unconditional variances and means are needed to describe the continuous density functions. The Gaussian influence diagram calculates conditional variances directly, and calculates the conditional means as linear functions of the realizations of conditioning variables.

Gaussian influence diagram operations can be considered in terms of two different operations. One operation is the process of moving a node upwards, towards the beginning of an ordered sequence. It is equivalent to expressing a random variable as being conditioned on fewer random variables than it is currently. The other operation is moving a node downwards, toward the end of the ordered sequence It is equivalent to expressing a random variable as being conditioned on more random variables than it is currently.

Before discussing these two types of operations, there is one notable characteristic of the influence diagram worth observing. If node i is a conditional predecessor of node j, and both are conditioned on a set of nodes K, then Kenley demonstrates the conditional covariance between them, conditioned on K, is equal to $u_{ij}v_i$, where v_i is the conditional variance of node i [12:534]. For a given set of conditioning variables, in this case K, the conditional covariance matrix is invariant. This means that v_i and $u_{ij}v_i$, the conditional variance and covariance, are unchanged as well. Recall that u_{ij} is the path coefficient as defined in Chapter 3 and represented in Equation (101). The author presents the following theorem and corollary, without further proof, that will be useful in later calculations.

Theorem 1 *If the conditional predecessors of a node are unchanged, then the path coefficient from that node to a given successor node is constant for all possible orderings of successor nodes.*

Corollary 1 *For a given set of conditioning variables, the path coefficient from a given node to a successor node is equivalent to the regression coefficient from the first to the second when the two nodes are placed consecutively in an ordered sequence.*

4.1.1 Exchanging a Node with a Predecessor. Consider now a single node in an influence diagram, exchanged with its predecessor to make it less conditional. This continues until the node is at the beginning of the ordered sequence, in unconditional form. If the node was originally the rth node in the sequence, then $r-1$ reversals are generally needed to move the node to the beginning. By necessity, the variance of the node, expressed in unconditional form, must equal the rth term on the diagonal of the covariance matrix that corresponds to the original influence diagram. In matrix form, the rth diagonal term in the covariance matrix can be calculated by $\mathbf{U}^T\mathbf{D}\mathbf{U}$ as

$$\Sigma_{rr} = v_r + (u_{r-1,r})^2 v_{r-1} + (u_{r-2,r})^2 v_{r-2} + , \ldots , (u_{2,r})^2 v_2 + (u_{1,r})^2 v_1 \qquad (142)$$

Since all terms in Equation (142) are positive, the result must be positive. Furthermore, the unconditional variance must always be greater than the value of the conditional variance v_r.

The influence diagram calculations to express a node in unconditional form must be equivalent to the expression in Equation (142). To verify that the influence diagram operations yield the same value, consider the general case of four nodes in an influence diagram, numbered sequentially, corresponding to the **U** and **D** matrices as given in Equations (102) and (104). For the matrix operations, Σ_{44} is calculated by:

$$\Sigma_{44} = v_4 + v_3 u_{34}^2 + v_2 u_{24}^2 + v_1 u_{14}^2 \tag{143}$$

$$= v_4 + v_3 b_{34}^2 + v_2 (b_{24} + b_{23} b_{34})^2$$

$$+ v_1 [b_{14} + b_{13} b_{34} + b_{12} (b_{24} + b_{23} b_{34})]^2 \tag{144}$$

Now assume that node 4 is being moved upwards to the unconditional position in the influence diagram. The equations for this influence diagram operation were given in Chapter 2. These equations result in new values for the variance and predecessor coefficients after one exchange as:

$$v_4' = v_4 + v_3 b_{34}^2 \tag{145}$$

$$b_{24}' = b_{24} + b_{23} b_{34} \tag{146}$$

$$b_{14}' = b_{14} + b_{13} b_{34} \tag{147}$$

It is worth noticing that the equations for the regression coefficient b_{24}' is equal to the value of u_{24} calculated earlier. This is a demonstration of Corollary 1 that the path coefficient u_{24} is equal to the regression coefficient b_{24}' when node 4 immediately follows node 2 in the ordered sequence. Also, the coefficient b_{14}' is equal to the first two terms of the previously calculated u_{14} term and b_{24}' is equal to the terms in parenthesis in Equation (107).

Another exchange upward results in

$$v''_4 = v'_4 + v_2(b'_{24})^2 \tag{148}$$

$$b''_{14} = b'_{14} + b_{12}b'_{24} \tag{149}$$

so that the new b''_{14} term is the same as the calculated u_{14}. The last exchange moves the 4th node to the top and the resulting variance is

$$v'''_4 = v''_4 + v_1(b''_{14})^2 \tag{150}$$

$$= v_4 + v_3b_{34}^2 + v_2(b_{24} + b_{23}b_{34})^2 + v_1[b_{14} + b_{13}b_{34} + b_{12}(b_{24} + b_{23}b_{34})]^2 \tag{151}$$

When v_4 is eventually moved to the upper left of the matrix and put into unconditional form, its variance will equal the fourth diagonal term in the covariance matrix, that is the Σ_{44} term. The expression for this variance, as calculated by the influence diagram algorithm, is identical to the expression given in Equation (144), computed by straightforward matrix multiplication of the original $\mathbf{U}^T\mathbf{D}\mathbf{U}$. The conclusion to be drawn is that the influence diagram algorithm for moving a node upwards is identical to calculating the diagonal terms of the $\mathbf{P}$ matrix by multiplying $\mathbf{U}^T\mathbf{D}\mathbf{U}$.

The factored form of the matrix also demonstrates the concept of "nuisance" variables. In the above example, if the values for nodes 1, 2, and 3, are not needed after they are conditioned on node 4, then there is no need to save them. In the influence diagram, this means that the nuisance nodes can be put at the end of an ordered sequence, and removed.

In the matrix operation, the new (primed) values would replace the old ones after every exchange. The rows and columns for the "nuisance" variables would not be needed. As an example,

if the **D** and **U** matrices are as before:

$$\mathbf{D} = \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & 0 & 0 \\ 0 & 0 & v_3 & 0 \\ 0 & 0 & 0 & v_4 \end{bmatrix} \tag{152}$$

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & b_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & b_{13} & 0 \\ 0 & 1 & b_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & b_{14} \\ 0 & 1 & 0 & b_{24} \\ 0 & 0 & 1 & b_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{153}$$

or in terms of the path coefficients:

$$\mathbf{U} = \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{154}$$

then, after the first exchange, only the first three rows and columns need be stored. In this case, $\mathbf{U}'$ and $\mathbf{D}'$ become:

$$\mathbf{D}' = \begin{bmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_4' \end{bmatrix} \tag{155}$$

$$\mathbf{U}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & b_{12} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & b_{14}' \\ 0 & 1 & b_{24}' \\ 0 & 0 & 1 \end{bmatrix} \tag{156}$$

The fourth row and column of the matrices do not affect the unconditional variance of the third term in the matrix. These extra rows and columns can be removed. After another exchange, the matrices are two dimensional. After the last exchange, $\mathbf{D}''' = v_4'''$, and $\mathbf{U}''' = 1$. This demonstrates that the matrix equivalent of removing a "nuisance" variable is the removal of the last row and column of the **U** and **D** matrices, or equivalently, the covariance matrix **P**. The "nuisance" variable corresponds to the last row and column, just as a "nuisance" variable in the influence diagram must be moved to the end of the ordered sequence before it is removed.

4.1.2 Exchanging a Node with a Successor. For every node that moves up, another moves down. The operation of moving a node downwards makes it "more" conditional in the sense that it is conditioned on more variables. As it becomes more conditional, the conditional variance becomes smaller. In the last example, these variables were discarded as "nuisance" variables. Sometimes though the conditional variance is the desired value. Such was the case for the discrete-time filter in which the conditional variance of the vector $\mathbf{x}(t_i)$, conditioned on the measurement vector $\mathbf{z}(t_i)$, was the value being calculated.

There are two useful ways of demonstrating the effect of making a variable more conditional. One is to move a node from an unconditional position to a conditional position where it has several predecessors. Another way is to move several nodes down one step as would happen when another node is moved upwards past them to become unconditional. Both processes will be demonstrated.

If four nodes represent an influence diagram, and the nodes are numbered sequentially in the ordered sequence, then node 1 can be conditioned on all other nodes by exchanging it one at a time with its successors. At each exchange, the new variance is recalculated in terms of the original conditional variances. The process is shown in Figure 45.

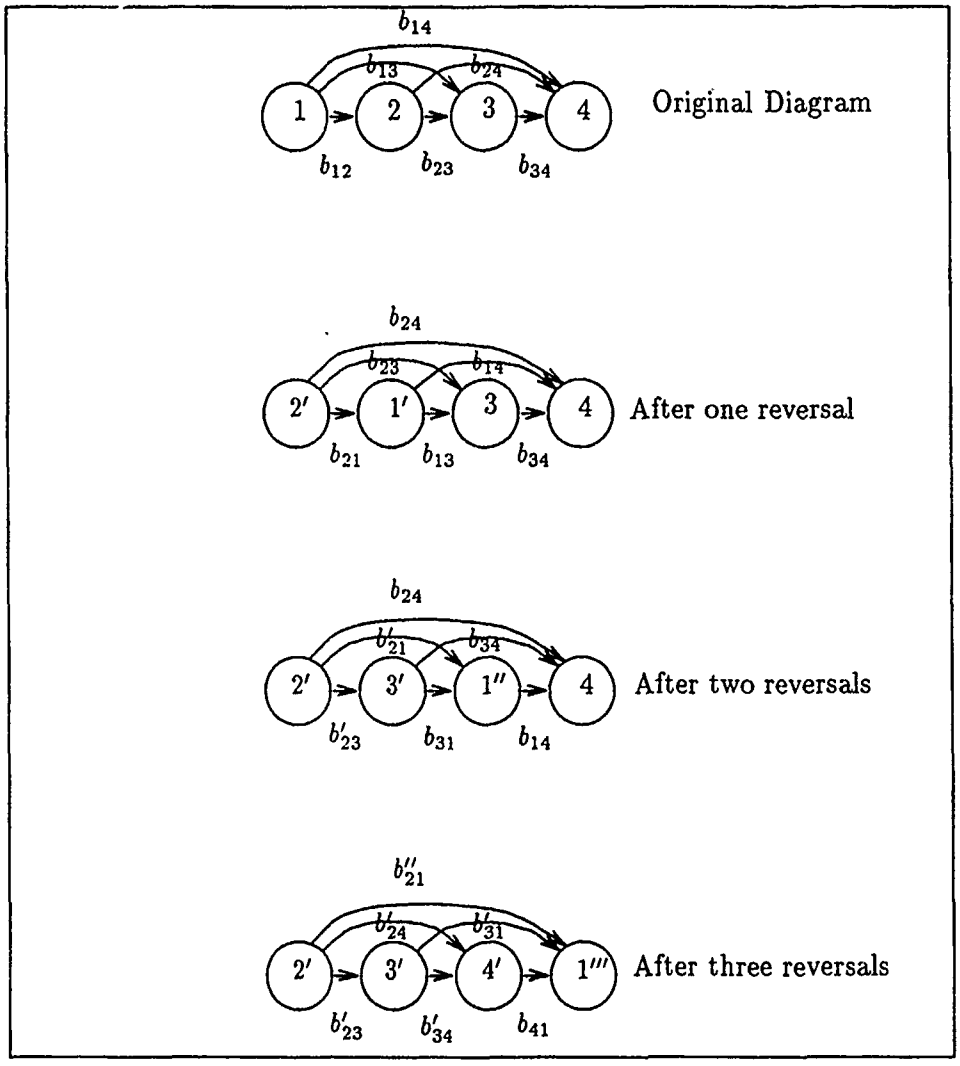


Figure 45. Conditioning a Node on All Other Nodes

The calculations for the variance of v_1 as it moves downward are:

$$v_2' = v_2 + b_{12}^2 v_1 \tag{157}$$

$$v_1' = \frac{v_1 v_2}{v_2'} \tag{158}$$

$$b_{21} = \frac{b_{12} v_1}{v_2'} \tag{159}$$

Moving node 1 down further requires an exchange of nodes 1 and 3. The equations for this next exchange are:

$$v_3' = v_3 + b_{13}^2 v_1' \tag{160}$$

$$b_{23}' = b_{23} + b_{21} b_{13} \tag{161}$$

$$v_1'' = \frac{v_1' v_3}{v_3'} \tag{162}$$

$$b_{31} = \frac{b_{13} v_1'}{v_3'} \tag{163}$$

$$b_{21}' = b_{21} - b_{23}' b_{31} \tag{164}$$

Finally, node 1 is exchanged with node 4 so that all nodes are conditioned on node 1. The resulting equations are:

$$v_4' = v_4 + b_{14}^2 v_1'' \tag{165}$$

$$b_{24}' = b_{24} + b_{21}' b_{14} \tag{166}$$

$$b_{34}' = b_{34} + b_{31} b_{14} \tag{167}$$

$$v_1''' = \frac{v_1'' v_4}{v_4'} \tag{168}$$

$$b_{41} = \frac{b_{14} v_1''}{v_4'} \tag{169}$$

$$b_{21}'' = b_{21}' - b_{24}' b_{41} \tag{170}$$

$$b_{31}' = b_{31} - b_{34}' b_{41} \tag{171}$$

If the resulting equation for v_1''' is solved in terms of the original variables, the result is:

$$v_1''' = \frac{v_1 v_2 v_3 v_4}{b_{12}^2 v_1 v_3 v_4 + b_{13}^2 v_1 v_2 v_4 + b_{14}^2 v_1 v_2 v_3 + v_2 v_3 v_4} \tag{172}$$

or, if $v_i \neq 0$ for i=1,2,3,4, then v_1''' can be expressed as:

$$\frac{1}{v_1'''} = \frac{1}{v_1} + \frac{b_{12}^2}{v_2} + \frac{b_{13}^2}{v_3} + \frac{b_{14}^2}{v_4} \tag{173}$$

The matrix for the inverse covariance in factored form is $(\mathbf{U}^T\mathbf{D}\mathbf{U})^{-1}$ or $\mathbf{U}^{-1}\mathbf{D}^{-1}\mathbf{U}^{-T}$ where $-T$ implies the inverse of the transpose. For the example given, the matrix operations are:

$$\begin{bmatrix} 1 & -b_{12} & -b_{13} & -b_{14} \\ 0 & 1 & -b_{23} & -b_{24} \\ 0 & 0 & 1 & -b_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/v_1 & 0 & 0 & 0 \\ 0 & 1/v_2 & 0 & 0 \\ 0 & 0 & 1/v_3 & 0 \\ 0 & 0 & 0 & 1/v_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -b_{12} & 1 & 0 & 0 \\ -b_{13} & -b_{23} & 1 & 0 \\ -b_{14} & -b_{24} & -b_{34} & 1 \end{bmatrix} \tag{174}$$

By multiplying these matrices, it can be shown that the first diagonal term of the resulting matrix is equal to the inverse of the the previously calculated value for v_1'''. This process can be generalized to show the conditional variance of a previously unconditional variable. If node 1 is to be conditioned on all other nodes, and if no nodes are deterministic (conditional variance equals zero), then the final conditional variance of v_1 has the form

$$v_1' = \frac{1}{\frac{1}{v_1} + \frac{b_{12}^2}{v_2} + \frac{b_{13}^2}{v_3} + \frac{b_{14}^2}{v_4} + \cdots \frac{b_{1n}^2}{v_n}}$$

Instead of v_1, assume the jth node is conditioned on all other nodes. Also assume node j is already conditioned on nodes 1 through $j-1$, so it must be further conditioned on nodes $j+1$ through n. If none of the nodes j through n are deterministic, then the conditional variance of v_j

is given by either

$$v_j' = \frac{1}{\frac{1}{v_j} + \frac{b_{1,j+1}^2}{v_{j+1}} + \frac{b_{1,j+2}^2}{v_{j+2}} + \frac{b_{1,j+3}^2}{v_{j+3}} + \cdots \frac{b_{1n}^2}{v_n}} \tag{175}$$

or

$$\frac{1}{v_j'} = \frac{1}{v_j} + \frac{b_{1,j+1}^2}{v_{j+1}} + \frac{b_{1,j+2}^2}{v_{j+2}} + \frac{b_{1,j+3}^2}{v_{j+3}} + \cdots \frac{b_{1n}^2}{v_n} \tag{176}$$

The form given in Equation (176) would be the expression for the conditional variance in the discrete-time filter if, for example, the jth node of the j dimensional vector $\mathbf{x}(t_i)$ were to be conditioned on the (n-j) dimensional vector $\mathbf{z}(t_i)$.

The form given in Equation (176) is important because it is the jth diagonal term of the inverse covariance matrix. This demonstrates that the conditional variance of a random variable, when conditioned on all other random variables in the vector, is equal to the inverse of the diagonal term of the inverse covariance matrix. It also demonstrates the relationship between the conditioning of a random variable as calculated by the influence diagram, and the factored form of the inverse covariance matrix.

The second of the two processes is a different look at the same kind of operation. Assume now that node 4 is to be put in unconditional form The remaining nodes will be conditioned on node 4. The variances of these other nodes can be expressed in terms of the original variables.

$$v_3' = v_3 \frac{v_4}{v_4 + b_{34}^2 v_3} \tag{177}$$

$$= v_3 \frac{v_4}{v_4 + u_{34}^2 v_3} \tag{178}$$

$$v_2' = v_2 \frac{v_4 + b_{34}^2 v_3}{v_4 + b_{34}^2 v_3 + (b_{24} + b_{23} b_{34})^2 v_2} \tag{179}$$

$$= v_2 \frac{v_4 + u_{34}^2 v_3}{v_4 + u_{34}^2 v_3 + u_{24}^2 v_2} \tag{180}$$

$$v_1' = v_1 \frac{v_4 + b_{34}^2 v_3 + (b_{24} + b_{23}b_{34})^2 v_2}{v_4 + b_{34}^2 v_3 + (b_{24} + b_{23}b_{34})^2 v_2 + [b_{14} + b_{13}b_{34} + b_{12}(b_{24} + b_{23}b_{34})]^2 v_1} \quad (181)$$

$$= v_1 \frac{v_4 + u_{34}^2 v_3 + u_{24}^2 v_2}{v_4 + u_{34}^2 v_3 + u_{24}^2 v_2 + u_{12}^2 v_1} \quad (182)$$

The conditional variances represented by v_1', v_2', and v_3' could occur in a discrete-time filter when the three dimensional vector $\mathbf{x}(t_i)$ is conditioned on the scalar measurement $z(t_i)$ represented by node 4. These conditional variances are calculated by division, not subtraction. They are expressed in terms of a fraction of the original variances. The numeric properties of this form will be discussed in Chapter 5.

4.1.3 Matrix Representation of Node Reversal. The operation of exchanging nodes in a covariance matrix $\mathbf{P}$ can itself be expressed in matrix form. The author offers this demonstration of a matrix equivalent of the influence diagram operation of node reversal. Assume the inverse factored matrix is $\mathbf{P}^{-1} = \mathbf{U}^{-1}\mathbf{D}^{-1}\mathbf{U}^{-T}$ and let $\mathbf{U}^{-1} = \mathbf{V} = (\mathbf{I} - \mathbf{B})$. The factored form of the covariance matrix is $\mathbf{V}^{-T}\mathbf{D}\mathbf{V}^{-1}$. The exchange of rows and columns can be accomplished by multiplying the covariance matrix, both before and after, by a transposition matrix of the form:

$$\mathbf{R} = \left[\begin{array}{c|cc} \mathbf{I} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline & 0 & 1 \\ \mathbf{0} & & \\ & 1 & 0 \end{array}\right] \quad (183)$$

where $\mathbf{R} = \mathbf{R}^{-1}$.

The new covariance matrix is $\mathbf{RPR}$. In this case, $\mathbf{R}$ exchanges the ith and jth row and column, where j is the last row of the matrix, and i is the second to the last row. In general, it is possible to exchange any ith and jth row and column, as long as i and j differ by one. The new matrix, in factored form, is $\mathbf{R}\mathbf{V}^{-T}\mathbf{D}\mathbf{V}^{-1}\mathbf{R}$ or $(\mathbf{RV})^{-T}\mathbf{D}(\mathbf{RV})^{-1}$. Unfortunately, the parenthetical expression $\mathbf{RV}$ is no longer a triangular matrix. It can be made triangular again by post-multiplying with a

matrix $\mathbf{X}$ such that $(\mathbf{RVX})^{-T}(\mathbf{X}^T\mathbf{DX})(\mathbf{RVX})^{-1}$. There is an $\mathbf{X}$ matrix that retriangularizes the $\mathbf{RVX}$ term, and maintains a diagonal middle term. One such value of $\mathbf{X}$ and $\mathbf{X}^{-1}$ that satisfies these conditions is:

$$\mathbf{X} = \left[\begin{array}{c|cc} \mathbf{I} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline & \multicolumn{2}{c}{} \\ \mathbf{0} & b_{ij} & \frac{v_j}{(v_j+b_{ij}^2 v_i)} \\ & 1 & \frac{-b_{ij}v_i}{(v_j+b_{ij}^2 v_i)} \end{array}\right] \tag{184}$$

$$\mathbf{X}^{-1} = \left[\begin{array}{c|cc} \mathbf{I} & \multicolumn{2}{c}{\mathbf{0}} \\ \hline & \multicolumn{2}{c}{} \\ \mathbf{0} & \frac{b_{ij}v_i}{(v_j+b_{ij}^2 v_i)} & \frac{v_j}{(v_j+b_{ij}^2 v_i)} \\ & 1 & -b_{ij} \end{array}\right] \tag{185}$$

After the matrix operations are carried out, the result is $(\mathbf{RVX})^{-T}(\mathbf{X}^T\mathbf{DX})(\mathbf{RVX})^{-1} = \mathbf{V}'^{-T}\mathbf{D}'\mathbf{V}'^{-1}$ where $\mathbf{D}'$ is diagonal and $\mathbf{V}'$ is upper triangular. This constitutes a reordering of the variables while still in factored form. By necessity, it must be equivalent to the factorization of the permuted matrix $\mathbf{RPR}$. The expressions for $\mathbf{V}'$ and $\mathbf{D}'$ are:

$$\mathbf{V}' = \begin{bmatrix} 1 & -b_{12} & -b_{13} & \cdots & -(b_{1j}+b_{1i}b_{ij}) & -\left\{b_{1i}-(b_{1j}+b_{1i}b_{ij})(\frac{b_{ij}v_i}{v_j+b_{ij}^2v_i})\right\} \\ & 1 & -b_{23} & \cdots & -(b_{2j}+b_{2i}b_{ij}) & -\left\{b_{2i}-(b_{2j}+b_{2i}b_{ij})(\frac{b_{ij}v_i}{v_j+b_{ij}^2v_i})\right\} \\ & & 1 & \cdots & -(b_{3j}+b_{3i}b_{ij}) & -\left\{b_{3i}-(b_{3j}+b_{3i}b_{ij})(\frac{b_{ij}v_i}{v_j+b_{ij}^2v_i})\right\} \\ & \mathbf{0} & & \ddots & \vdots & \vdots \\ & & & & 1 & -(\frac{b_{ij}v_i}{v_j+b_{ij}^2v_i}) \\ & & & & 0 & 1 \end{bmatrix} \tag{186}$$

$$\mathbf{D}' = \begin{bmatrix} v_1 & & & & \\ & v_2 & & \mathbf{0} & \\ & & v_3 & & \\ & & & \ddots & \\ & \mathbf{0} & & v_j + b_{ij}^2 v_i & \\ & & & & \frac{v_j v_i}{v_j + b_{ij}^2 v_i} \end{bmatrix} \tag{187}$$

The important details of these matrices are the operations in the ith and jth columns. These are seen to be identical to the influence diagram operations necessary for reversing the ith and jth node.

4.1.4 Comparison with Kalman Filtering. The influence diagram is equivalent to a factored form of the covariance matrix. One cycle of the influence diagram for the discrete-time filter can be represented in two useful forms, as was shown in Figures (18) and (42). Each of these diagrams represents a singular covariance matrix $\mathbf{P}$, or its factored form $\mathbf{U}^T\mathbf{D}\mathbf{U}$. When the matrices are correctly defined, the matrix operations in the previous section can be used to perform the influence diagram operations. This section will present the matrices necessary for the influence diagram representation of the discrete-time filter.

Assume that the discrete time filter is represented by the influence diagram in Figure 18. Let the vector $\mathbf{x}(t_{i-1})$ be represented in influence diagram form as a diagonal matrix $\mathbf{D}_x$ and a strictly upper triangular matrix $\mathbf{B}_x$ such that the covariances matrix for $\mathbf{x}(t_{i-1})$ is given by $(\mathbf{I}-\mathbf{B}_x)^{-T}\mathbf{D}_x(\mathbf{I}-\mathbf{B}_x)^{-1}$. The vector $\mathbf{w}_d(t_{i-1})$ has a covariance represented by a diagonal matrix $\mathbf{Q}_d(t_{i-1})$ and a linear input matrix $\mathbf{G}_d(t_{i-1})$. The vector $\mathbf{v}(t_i)$ has a covariance of matrix $\mathbf{R}(t_i)$ which is assumed to be diagonal. The case of $\mathbf{R}(t_i)$ not being diagonal will be discussed later. With

these variables defined, the diagonal matrix for the entire discrete-time filter influence diagram is

$$\mathbf{D}_{diagram} = \begin{bmatrix} \mathbf{D}_x & 0 & 0 & 0 & 0 \\ 0 & \mathbf{Q}_d(t_{i-1}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{R}(t_i) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{188}$$

The dimension of the square matrix represented by the 0 term in the middle of the diagonal is the same as the dimension of $\mathbf{D}_x$. It corresponds to the deterministic function represented by $\mathbf{x}(t_i)$. The dimension of the square matrix represented by the last 0 is the number of terms in the measurement vector. It corresponds to the deterministic function represented by $\mathbf{z}(t_i)$.

The **B** matrix for the influence diagram can be constructed similarly. Using the linear relations from the influence diagram, the matrix is

$$\mathbf{B}_{diagram} = \begin{bmatrix} \mathbf{B}_\mathbf{X} & 0 & \mathbf{\Phi}^T(t_i, t_{i-1}) & 0 & 0 \\ 0 & 0 & \mathbf{G}_d^T(t_{i-1}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{H}^T(t_i) \\ 0 & 0 & 0 & 0 & \mathbf{I} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{189}$$

The dimension of each row and column is the same as the corresponding rows and columns of Equation (188).

These matrices, combined with the matrix operations from the previous section, can be used to calculate the influence diagram operations using matrix software tools. These matrix operations will accomplish the following steps. First, eliminate the second group of rows and columns from both matrices by moving them to the end. This is equivalent to eliminating the vector $\mathbf{w}_d(t_{i-1})$.

Then remove the first group of rows and columns in the same way, equivalent to removing the vector $\mathbf{x}(t_{i-1})$. Remove the rows and columns corresponding to $\mathbf{R}(t_i)$, leaving only the rows and columns for $\mathbf{x}(t_i)$ and $\mathbf{z}(t_i)$. Now reverse the two so that the rows and columns for $\mathbf{z}(t_i)$ precede those of $\mathbf{x}(t_i)$. Finally, use the resulting $\mathbf{B}$ matrix to update the states with the measurements, and discard the first columns that correspond to $\mathbf{z}(t_i)$. The diagonal and strictly upper triangular matrix that remain are the influence diagram equivalent for the covariance matrix associated with $\mathbf{x}(t_i)$, after incorporating the measurements at time t_i. These remaining matrices can be inserted back into the matrices of Equations (188) and (189) as $\mathbf{D}_x$ and $\mathbf{B}_x$, and the process starts over for the next time interval.

For the case of $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ being represented in influence diagram form as in Figure 42, there are some minor changes to the matrices of Equations (188) and (189). Instead of $\mathbf{Q}_d(t_{i-1})$, the appropriate diagonal term of $\mathbf{D}_{diagram}$ is the diagonal matrix from the factored form of $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$. The 0 in the second diagonal place of $\mathbf{B}_{diagram}$ is replaced with the $\mathbf{B}$ matrix from the factored form of $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$. The $\mathbf{G}_d^T(t_{i-1})$ term of $\mathbf{B}_{diagram}$ becomes the identity matrix. The description of the matrix operations remains unchanged.

If $\mathbf{R}(t_i)$ is not diagonal, then it can be factored in influence diagram form. The diagonal terms of the factorized matrix, called $\mathbf{D}_v$, would replace the diagonal terms of $\mathbf{R}(t_i)$. The zero matrix in the second to the last diagonal position of $\mathbf{B}_{diagram}$ would be replaced by $\mathbf{B}_v$, the upper triangular matrix terms from the factorization of $\mathbf{R}(t_i)$.

The descriptions in the preceding paragraphs for removing columns and rows are identical to the operations described in Chapter 2 for implementing the influence diagram. The matrices such as in Equations (188) and (189) demonstrate the correct setup for each set of operations. Equations (185) and (187) are the matrix operations needed to accomplish the correct reversals and reductions.

4.1.5 Comparison with U-D Filtering. The U-D factored form of the Kalman filter is a stable, efficient method of calculating the means and covariances. The name refers to the factored form of the covariance matrix, which is in the form of a unit upper triangular, diagonal, and a transposed unit upper triangular matrix, or $\mathbf{UDU}^T$. If the covariance matrix is nonsingular, then the factorization is unique. If $\mathbf{D}$ is itself factored into two identical matrices, then these matrices take the form of a diagonal square root matrix $\mathbf{S}^{1/2}$. Because it is diagonal, the transpose of $\mathbf{S}^{1/2}$ is equal to itself. The factorization $(\mathbf{US}^{1/2})(\mathbf{S}^{1/2}\mathbf{U}^T)$ is also unique and is equivalent to an upper triangular Cholesky decomposition. The conventional Cholesky decomposition takes the form of $\mathbf{P} = \mathbf{LL}^T$, where the calculated matrix is lower triangular [13:pp. 133–143].

The influence diagram also uses a factored form of the covariance matrix, but uses $\mathbf{U}_I^T\mathbf{D}_I\mathbf{U}_I$ where the subscripts refer to a different $\mathbf{U}$ and $\mathbf{D}$ matrix for the influence diagram. It is apparent that this can also be factored into lower triangular and upper triangular matrices by using $\mathbf{S}_I^{1/2}$ as the square root of $\mathbf{D}_I$ and $\mathbf{S}_I^{1/2}$ is its own transpose. This factorization, $(\mathbf{S}_I^{1/2}\mathbf{U}_I)^T(\mathbf{S}_I^{1/2}\mathbf{U}_I)$ is the Cholesky decomposition of the covariance matrix [12:pp. 547–548].

Both the lower triangular Cholesky and the upper triangular Cholesky decompositions are unique, as are the $\mathbf{UDU}^T$ and $\mathbf{U}_I^T\mathbf{D}_I\mathbf{U}_I$ factorizations. The author offers the following theorem as a comparison between the influence diagram and the U-D factored forms of the covariance matrix.

Theorem 2 *If* $\mathbf{P}$ *refers to a positive definite symmetric matrix with U-D factors of* $\mathbf{UDU}^T$, *and* $\mathbf{P}^*$ *refers the matrix* $\mathbf{P}$ *with all rows and columns in reverse order, then* $\mathbf{P}^* = \mathbf{U}^*\mathbf{D}^*\mathbf{U}^{*T}$ *where* $\mathbf{D}^* = \mathbf{D}_I$ *and* $\mathbf{U}^* = \mathbf{U}_I^T$.

Proof. Let $\mathbf{R}$ be a square matrix of same order as $\mathbf{U}$ and $\mathbf{D}$ of the form:

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Then $\mathbf{P}^* = \mathbf{RPR}$ and $\mathbf{R}$ is its own inverse because $\mathbf{RR} = \mathbf{I}$.

$$\mathbf{P}^* = \mathbf{R}(\mathbf{U}\mathbf{D}\mathbf{U}^T)\mathbf{R} \tag{190}$$

$$\mathbf{P}^* = \mathbf{R}[\mathbf{U}(\mathbf{RR})\mathbf{D}(\mathbf{RR})\mathbf{U}^T]\mathbf{R} \tag{191}$$

$$\mathbf{P}^* = (\mathbf{RUR})(\mathbf{RDR})(\mathbf{R}\mathbf{U}^T\mathbf{R}) \tag{192}$$

The factorization is unique, therefore

$$\mathbf{RUR} = \mathbf{U}^* = \mathbf{U}_I^T \tag{193}$$

$$\mathbf{RDR} = \mathbf{D}^* = \mathbf{D}_I \tag{194}$$

In other words, $\mathbf{P}$ is the covariance matrix for a state vector, and $\mathbf{P}^*$ is the covariance matrix for the state vector in reverse order. The U-D factored form of $\mathbf{P}$ will have the same $\mathbf{U}$ and $\mathbf{D}$ matrices as the influence diagram form of $\mathbf{P}^*$, but in reverse order.

After the covariance matrix is factored, the U-D algorithm does not normally allow the exchange of variable order. On the other hand, the influence diagram algorithm depends on the exchange of two consecutive variables while the matrix is in factored form. The new influence diagram corresponds to a factorization of a new covariance matrix that has a row and column pair

reversed. For a given covariance matrix, this reversal of a pair of rows and columns is equivalent to exchanging the corresponding random variables in the original vector.

Another insight comes from this analysis. For a given positive definite symmetric matrix, there are only two triangular-diagonal factorizations. One factorization is $\mathbf{U}\mathbf{D}_1\mathbf{U}^T$ and the other is $\mathbf{L}\mathbf{D}_2\mathbf{L}^T$ where $\mathbf{U}$ is unit upper triangular, $\mathbf{L}$ is unit lower triangular, and both $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal. The U-D filter uses the first factorization and the influence diagram uses the second. The distinction between the two implementations is the influence diagram factorization starts at the top left of the covariance matrix and proceeds downwards, while the U-D factorization starts at the lower right and works upwards.

The influence diagram and the U-D filter have similar forms, and they have the same values if one of them represents the covariance of a reversed order vector. But, the influence diagram is further factored into $\mathbf{U}_I = \mathbf{U}_1\mathbf{U}_2\mathbf{U}_3 \ldots \mathbf{U}_n$ The further factorization does not require more storage but does add insight into the meaning of the variables. This form stores the regression coefficients b_{ij} and uses them to calculate the path coefficients u_{ij} instead of storing the path coefficients directly.

4.2 Chapter Summary

This chapter demonstrated several similarities between influence diagram calculations and matrix operations. The calculations associated with making a node less conditional are identical to the calculation of the diagonal terms of the matrix product $\mathbf{U}^T\mathbf{D}\mathbf{U}$. The calculations associated with making a node more conditional are identical to the calculation of the inverse of the diagonal terms of the matrix product $\mathbf{U}^{-1}\mathbf{D}^{-1}\mathbf{U}^{-T}$. When two nodes in an influence diagram are exchanged, there is a simple way of expressing, in matrix form, the combined operations of making one node less conditional and one node more conditional. It was shown that the influence diagram operations are equivalent to retriangularizing a matrix using another matrix $\mathbf{X}$ as given in Equation (184).

This chapter also demonstrated a method for representing one complete cycle of a Kalman filter in matrix form for influence diagram operations. This would be useful for implementing the influence diagram algorithm on a computer without dedicated influence diagram software. Finally, this chapter demonstrated the similarities between the U-D filter and the influence diagram. It was proven that the influence diagram and U-D factorizations are equivalent in the same sense that the lower triangular Cholesky and the upper triangular Cholesky decompositions are equivalent. In this sense, they can be considered "mirror image" factorizations of the covariance matrix.

V. Numerical Properties

5.1 Matrix Operations

Because the influence diagram is based on matrix operations, the stability and numerical properties of the algorithm can be shown using matrix operations. The basis for such analysis are the matrix operations $\mathbf{U} = (\mathbf{I} - \mathbf{B})^{-1}$ and the factorization of the covariance matrix into the $\mathbf{U}$ and $\mathbf{D}$ matrices such that $\mathbf{P}(t_i) = \mathbf{U}^T\mathbf{D}\mathbf{U}$. The terms of the $\mathbf{U}$ matrix can be computed by the matrix inversion $(\mathbf{I} - \mathbf{B})^{-1}$ and were given in Equation (101) as:

$$u_{1,r} = b_{1,r} + b_{1,r-1}(u_{r-1,r}) + b_{1,r-2}(u_{r-2,r}) + \ldots, b_{1,2}(u_{2,r}) \tag{195}$$

The terms of the $\mathbf{U}$ matrix are the main source of error in the influence diagram. If $|u_{ij}| \ll |b_{ij}|$, then the u_{ij} term is calculated by subtracting terms of nearly equal size. This means there will be a loss of significant digits.

One set of circumstances that may cause a cancellation of significant digits is when a successor node is nearly independent of a predecessor. If there are interceding nodes between the two, then the regression coefficient from the predecessor to the successor may be much larger in magnitude than the path coefficient. If the successor is made less conditional, and moved ahead of the interceding nodes then, by Corollary 1, the regression coefficient will become the small arc coefficient. The calculation will the require subtraction of two numbers of nearly equal size and may have large relative error. Numerically, consider a regression coefficient b_{13} to be approximately equal to the negative of the path product $b_{12}b_{23}$. The path coefficient is calculated as $u_{13} = b_{13} + b_{12}b_{23}$. The path coefficient u_{13} will be much smaller in magnitude than the original value for b_{13}, and may have large relative error. Some general rules for decreasing the likelihood of this type of error will be discussed later in this chapter. However, the author knows of no consistent method for solving this problem.

If $\mathbf{P}$ is an unconditional covariance matrix, then the diagonal terms of the matrix product $\mathbf{U}^T\mathbf{D}\mathbf{U}$ are the unconditional variances of the random variables. They can be computed using either the matrix product or the influence diagram algorithm. The equation for calculating the unconditional variances was given in Equation (142) and is repeated here:

$$\Sigma_{rr} = v_r + (u_{r-1,r})^2 v_{r-1} + (u_{r-2,r})^2 v_{r-2} +, \cdots, (u_{2,r})^2 v_2 + (u_{1,r})^2 v_1 \tag{196}$$

This equation shows the unconditional variance of the rth term in the matrix can be calculated using the variances and path coefficients for less conditional terms, i.e. terms with subscripts less than r.

In this equation, the calculated variance is the sum of positive products. There will be a high numerical accuracy in computing this sum because there can be no cancellation of significant digits, as occurs with the small difference of large numbers. Instead, the only source of numerical difficulties occurs in the calculation of the u_{ij} terms. However, cancellation of significant digits in one of the u_{ij} terms may not affect the overall relative error in calculating the unconditional variance unless the other terms being added together are relatively small as well.

The variance of a variable, conditioned on all other variables, was shown to be the inverse of the diagonal terms of the inverse covariance matrix $\mathbf{P}^{-1}$. If any of the conditioning variables is deterministic, i.e. has a variance of zero, then the variance of the conditioned variable is zero also. If none of the conditioning variables is deterministic, then the variance of a variable, conditioned on all other variables in the vector was given in Equation (176) as:

$$\frac{1}{v_j'} = \frac{1}{v_j} + \frac{b_{1,j+1}^2}{v_{j+1}} + \frac{b_{1,j+2}^2}{v_{j+2}} + \frac{b_{1,j+3}^2}{v_{j+3}} + \cdots \frac{b_{1n}^2}{v_n} \tag{197}$$

The implication of this form is that calculating the conditional variance is done with high relative accuracy. There are no errors due to cancellation of significant digits as existed in the

calculation of unconditional variances. One possible source of problem is when one of the successor variances is very small. The inverse will be very large and will dominate the other addends in the sum. If the corresponding numerator b_{ij} value is also small, and was calculated with high relative error because of a cancellation of significant digits, then this relative error will be passed on to the calculation of the variance.

5.1.1 Stability of Calculations. Both increasing and decreasing the conditioning of a node are numerically stable operations. In this sense, numeric stability is defined as the property that the computed result of the algorithm is the same as the the exactly computed solution of a problem that has its values slightly perturbed from the true values [15].

In order to show stability, assume the the symbol $fl()$ represents the floating point equivalent of the operation in the parenthesis. Wilkinson [15] shows that $fl(\mathbf{AB})=\mathbf{AB}+\mathbf{E}$ where $\mathbf{A}$ and $\mathbf{B}$ are matrices compatible for multiplication and $\mathbf{E}$ is an error matrix. Similarly $fl(\mathbf{VD}^{-1}\mathbf{V}^T) = \mathbf{VD}^{-1}\mathbf{V}^T + \mathbf{E}$. If we define the matrix $\mathbf{E}$ to be

$$\begin{aligned}\mathbf{E} &= (\mathbf{VD}^{-1}\mathbf{E}_v^T + \mathbf{VE}_{d1}^{-1}\mathbf{V}^T + \mathbf{E}_v\mathbf{D}^{-1}\mathbf{V}^T) \\ &\quad +(\mathbf{VE}_{d1}^{-1}\mathbf{E}_v^T + \mathbf{E}_v\mathbf{E}_{d1}^{-1}\mathbf{V}^T + \mathbf{E}_v\mathbf{D}^{-1}\mathbf{E}_v^T) + (\mathbf{E}_v\mathbf{E}_{d1}^{-1}\mathbf{E}_v^T) \qquad (198)\end{aligned}$$

then

$$fl(\mathbf{VD}^{-1}\mathbf{V}^T) = \mathbf{VD}^{-1}\mathbf{V}^T + \mathbf{E} \qquad (199)$$

$$= (\mathbf{V}+\mathbf{E}_v)(\mathbf{D}^{-1}+\mathbf{E}_{d1}^{-1})(\mathbf{V}+\mathbf{E}_v^T) \qquad (200)$$

where $\mathbf{E}_v$ and $\mathbf{E}_{d1}$ are error matrices associated with the rounding errors in $\mathbf{V}$ and $\mathbf{D}^{-1}$, respectively. The "1" in the subscript of $\mathbf{E}_{d1}$ refers to the first of two error matrices associated with the diagonal matrix $\mathbf{D}$. The error matrix $\mathbf{E}_{d1}$ is a diagonal matrix with elements of the same magnitude as potential rounding errors in $\mathbf{D}$. This is because the error in computing the inverse of the diagonal

terms is on the same order as the original rounding errors in the terms themselves [15]. The elements of the error matrix $\mathbf{E}_v$ are of the same relative magnitude as the potential rounding errors in the elements of $\mathbf{V}$ [16:232].

If $\mathbf{V}$ is equal to $\mathbf{I} - \mathbf{B}$, then the matrix product $\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^T$ is the inverse of the covariance matrix as shown in Chapter 3. By Equation (200), it can be seen that all error matrices have elements on the same order of magnitude as the rounding errors in the initial values of $\mathbf{B}$ and $\mathbf{D}$. Recall that the diagonal terms of the inverse covariance matrix are the inverse of the variances, conditioned on all other variables, as calculated by the influence diagram. It can then be concluded that the errors in the influence diagram algorithm for calculating the conditional covariance terms are of the same relative magnitude as the rounding errors in the original variances and regression coefficients.

Now assume that $\mathbf{P} = \mathbf{U}^T\mathbf{D}\mathbf{U}$ is the covariance matrix of an unconditional vector. The algorithm for calculating the unconditional variances, represented by the diagonal terms of $\mathbf{P}$, is equivalent to the influence diagram operation of making a node unconditional. Using logic similar to before, and using $\mathbf{U}^T\mathbf{D}\mathbf{U} = \mathbf{V}^{-T}\mathbf{D}\mathbf{V}^{-1}$:

$$fl(\mathbf{U}^T\mathbf{D}\mathbf{U}) = \mathbf{U}^T\mathbf{D}\mathbf{U} + \mathbf{E} \tag{201}$$

where

$$\mathbf{E} = (\mathbf{U}^T\mathbf{D}\mathbf{E}_u + \mathbf{U}^T\mathbf{E}_{d2}\mathbf{U} + \mathbf{E}_u^T\mathbf{D}\mathbf{U}) + (\mathbf{U}^T\mathbf{E}_{d2}\mathbf{E}_u + \mathbf{E}_u^T\mathbf{E}_{d2}\mathbf{U} + \mathbf{E}_u^T\mathbf{D}\mathbf{E}_u) + (\mathbf{E}_u^T\mathbf{E}_{d2}\mathbf{E}_u) \tag{202}$$

and therefore

$$fl(\mathbf{U}^T\mathbf{D}\mathbf{U}) = \mathbf{U}^T\mathbf{D}\mathbf{U} + \mathbf{E} \tag{203}$$

$$= (\mathbf{U} + \mathbf{E}_u)^T(\mathbf{D} + \mathbf{E}_{d2})(\mathbf{U} + \mathbf{E}_u) \tag{204}$$

The error matrix $\mathbf{E}_{d2}$ is associated with rounding errors in $\mathbf{D}$. It is diagonal and has elements of the same order of magnitude as the rounding error in the elements of $\mathbf{D}$. Similarly, $\mathbf{E}_u$ is the error associated with the matrix $\mathbf{U}$. However, the elements of $\mathbf{E}_u$ are not necessarily of the same magnitude as the rounding errors in the original values of $\mathbf{D}$ and $\mathbf{B}$. Instead, $\mathbf{V} = (\mathbf{I} - \mathbf{B})$ used the original regression coefficients and $\mathbf{U}$ was calculated as the inverse of $\mathbf{V}$. It can be shown however that if $\mathbf{V}^{-1} = \mathbf{U}$, then

$$fl(\mathbf{V}^{-1}) = (\mathbf{V} + \mathbf{E}_v)^{-1} \tag{205}$$

$$= (\mathbf{U} + \mathbf{E}_u) \tag{206}$$

so the errors can still be related to rounding errors in the original values [15]. There is an upper bound on the error, $\mathbf{E}_u$, associated with computing the inverse of a matrix $\mathbf{V}$. However, as will be shown in the next section, this upper bound is the product of $\mathbf{E}_v$, the error matrix associated with the original matrix $\mathbf{V}$, and a positive scalar greater than unity [15:105]. The result is that the errors in calculating the inverse of a matrix are almost invariably larger than the rounding errors in the original matrix.

Using these demonstrations, the algorithms for calculating the conditional and the unconditional variances are seen to be stable. Both are equivalent to a simple matrix multiply algorithm. Such an algorithm is numerically stable, and will be relatively accurate as long as no severe cancellation of significant digits occurs [15]. Because these matrix multiplications are in quadratic form, only positive values are added and no cancellations can occur.

The relative accuracy of the algorithm for calculating conditional variances will be better than the algorithm for computing unconditional variances. This is because the algorithm for calculating the conditional variances has errors of the same magnitude as the rounding errors in the original values of variances and regression coefficients. The algorithm for calculating unconditional variances has potentially larger errors, equivalent to the errors associated with matrix inversion.

5.2 Error Analysis

The errors in calculating the variance can be compared with other operations that have known error bounds. In the first case above, the variance decreased as the node became more conditional. The process was equivalent to forming the inverse covariance matrix.

It has already been shown that, if $\mathbf{D}$ is factored into the diagonal square roots $\mathbf{D} = \mathbf{S}^T\mathbf{S}$ and $\mathbf{P} = (\mathbf{U}^T\mathbf{S}^T)(\mathbf{SU})$, then the $\mathbf{U}^T\mathbf{DU}$ factorization is equivalent to the lower triangular Cholesky decomposition of the covariance matrix $\mathbf{P}$. Similarly, the factorization $\mathbf{VD}^{-1}\mathbf{V}^T = \mathbf{P}^{-1}$ is the upper triangular Cholesky factorization of the inverse covariance matrix. The factorization of the inverse is more important now because it yields the values of the regression coefficients directly.

Wilkinson shows the Cholesky factorization is the most accurate known factorization for a symmetric, positive definite matrix [16:244]. If the matrices $\mathbf{L}$ and $\mathbf{L}^T$ represent the Cholesky decomposition matrices of the matrix $\mathbf{P}^{-1}$, then the error bounds can be given as

$$\mathbf{LL}^T = \mathbf{P}^{-1} + \mathbf{F}$$

where $\mathbf{F}$ is an error matrix. Under the assumption that inner products are accumulated in double precision and then rounded to single precision, Wilkinson gives error bounds for the elements of the $\mathbf{F}$ matrix as

$$|f_{rs}| \leq \left\{ \begin{array}{ll} |l_{rs}l_{ss}|2^{-t} & (r > s) \\ |l_{rs}l_{rr}|2^{-t} & (r < s) \\ l_{rr}^2 2^{-t} & (r = s) \end{array} \right\} \tag{207}$$

where t is the number of significant binary digits used by the computer [16:232].

When the influence diagram is used to calculate the equivalent of $\mathbf{VD}^{-1}\mathbf{V}^T = \mathbf{P}^{-1}$, there will be additional (small) errors due to the middle term. Also, there are no inner products to store in double precision as can be done with the matrix product. Instead, the variances in the influence

diagram are calculated for each node exchange and stored in single precision. Because of these differences, the possible errors are larger than given by Wilkinson. One unrestrictive error bound replaces t by $t/2$. It is the equivalent of assuming that t digits represent double precision and $t/2$ digits represent single precision. These assumptions are very pessimistic and will overestimate the error bound. Nevertheless, by simply substituting t for $t/2$, and using the influence diagram notation, the error bounds become

$$\mathbf{V}\mathbf{D}^{-1}\mathbf{V}^T = \mathbf{P}^{-1} + \mathbf{F} \tag{208}$$

$$|f_{rs}| \leq \left\{ \begin{array}{cc} |b_{ij}/v_{ii}|\, 2^{-t/2} & (i > j) \\ |b_{ij}/v_{jj}|\, 2^{-t/2} & (i < j) \\ 1/v_{ii}^2 2^{-t/2} & (i = j) \end{array} \right\} \tag{209}$$

The error bounds for the factored form $\mathbf{U}^T\mathbf{D}\mathbf{U}$ are not as small. The matrix $\mathbf{U}$ is formed by the equivalent of taking the inverse $\mathbf{U} = \mathbf{V}^{-1} = (\mathbf{I} - \mathbf{B})^{-1}$. The errors in the $\mathbf{V}$ matrix are of the same magnitude as the rounding errors due to floating point arithmetic. Using the notation and values as shown by Wilkinson, the errors in the matrix $\mathbf{U} = \mathbf{V}^{-1}$ can be bounded as follows. Let $\|\,\|$ represent any consistent matrix norm except the 2-norm. Also, if t is the number of significant binary digits in the computer, then let $t_1 = t - 0.08406$, where t_1 accounts for higher order error terms. Then, for an n-dimensioned triangular matrix V[15:105]:

$$\frac{\|\mathbf{V}^{-1} - \mathbf{U}\|}{\|\mathbf{V}^{-1}\|} \leq \frac{n2^{-t_1}\|\mathbf{V}\|\,\|\mathbf{V}^{-1}\|}{1 - n2^{-t_1}\|\mathbf{V}\|\,\|\mathbf{V}^{-1}\|} \tag{210}$$

This error bound is usually very pessimistic also. The product of the terms $\|\mathbf{V}\|\,\|\mathbf{V}^{-1}\|$ is also called the condition number of the matrix with respect to inversion. Wilkinson shows that it is possible for triangular matrices to have very large condition numbers, yet have very small errors

in calculating the inverse. In general, the inverse of a triangular matrix has an error such that:

$$\frac{\|\mathbf{V}^{-1} - \mathbf{U}\|}{\|\mathbf{V}^{-1}\|} \leq f(n)2^{-t}\|\mathbf{V}^{-1}\| \tag{211}$$

where $f(n)$ is a simple function of the dimension n [15:106].

Strictly speaking, the maximum possible error in computing the inverse is given in Equation (210). However, it is more likely that the errors will be much smaller, as given in Equation (211). In spite of these error bounds, it is still possible for some elements of the inverse of a diagonal matrix to have a high relative error. This is the case in which severe cancellation takes place during the inversion. Cancellation of significant digits in the inverse of the V matrix is the same as the cancellation of significant digits which occurs when a path coefficient u_{ij} is relatively small in magnitude compared to the regression coefficients used to calculate it.

5.3 Numeric Examples

The following examples show advantages of the influence diagram and some of the problems with numeric deterioration in both the U-D factored form of the Kalman filter, and in the influence diagram. The first example is an application of the influence diagram to a scalar form of the Kalman filter update. The second example is the influence diagram solution of a problem proposed by Bierman [3:97]. It demonstrates some of the conditions for error in the influence diagram. This example also demonstrates one of the features of the influence diagram; it shows how a change in the order of operations in the influence diagram may avoid the numeric error. The third example shows how numeric errors can occur in the U-D filter.

5.3.1 Scalar Update Example. The first example is a demonstration that the influence diagram is equivalent to the Kalman filter for the case of scalar variables. To make the demonstration simpler, let x and z be two zero-mean, jointly Gaussian random variables. The terms $\hat{x}^-$ and $\hat{z}^-$

are defined as in the Kalman filter example of Chapter 2, and $\hat{x}^- = \hat{z}^- = 0$. These vectors have variances P^- and R respectively. The minus superscript indicates that these variables are the estimates prior to the measurement update. The Kalman filter calculates the conditional mean and variance of x given ζ as the realization of z. The update equations for these scalar variables yields

$$K = \frac{P^- H}{P^- H^2 + R} \tag{212}$$

$$P^+ = P^- - KHP^- \tag{213}$$

$$\begin{aligned} \hat{x}^+ &= \hat{x}^- + K\left[\zeta - H\hat{x}^-\right] \\ &= K\zeta \end{aligned} \tag{214}$$

The influence diagram for the same conditions, and the operations for the update are shown in Figure 46. It is a pair of nodes labeled x and z with variances P^+ and R respectively. The arrow goes from x to z and has a regression coefficient of H. The variance R is seen to be the conditional variance of z given x. The reversal of this arrow by Bayes' rule yields the unconditional variance of z, the regression coefficient of x on z called K, and the conditional variance of x given z called P^+.

$$A = P^- H^2 + R \tag{215}$$

$$P^+ = \frac{P^- R}{A} = \frac{P^- R}{P^- H^2 + R} \tag{216}$$

$$K = \frac{P^- H}{A} = \frac{P^- H}{P^- H^2 + R} \tag{217}$$

The scalar update of the state estimate is

$$\begin{aligned} \hat{x}^+ &= \hat{x}^- + K\left[\zeta - \hat{z}\right] \\ &= K\zeta \end{aligned} \tag{218}$$

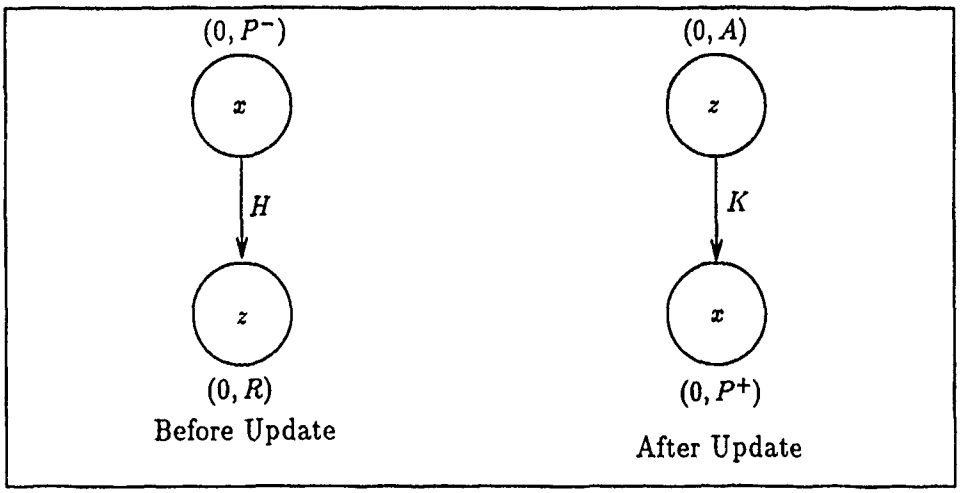


Figure 46. Scalar Update of x by z

It is apparent that these algorithms both yield the same Kalman gain K, conditional mean $\hat{x}^+$, and conditional variance P^+. The Kalman gain is the coefficient on the arrow z to x. It can be shown that the value for P^+ is the same by using the Kalman filter form and converting it as:

$$P^+ = P^- - KHP^- \tag{219}$$

$$P^+ = P^- - (\frac{P^-H}{P^-H^2 + R})HP^- \tag{220}$$

$$P^+ = \frac{(P^-H)^2 + P^-R - (P^-H)^2}{P^-H^2 + R} \tag{221}$$

$$P^+ = \frac{P^-R}{P^-H^2 + R} \tag{222}$$

In the matrix form of the Kalman filter, the simplifications given above are not permitted. In the scalar form however, the conditional variance is calculated using division, not subtraction. This form is numerically superior as R goes to zero. The influence diagram calculates the conditional variance of a random variable using the numerically superior scalar operation. It calculates the conditional variance of a random vector as the conditional variances of a series of scalar random variables, each using the numerically superior scalar equation.

5.3.2 Bierman's Example. Bierman proposed a problem that was known to cause numeric deterioration in the Kalman filter. Assume the initial estimates of x_1 and x_2 are zero and that $\mathbf{P} = \sigma^2\mathbf{I}$ where $\sigma = 1/\epsilon$ is large. Assume also that ϵ is small enough that, due to rounding, $1+\epsilon \neq 1$ but $1+\epsilon^2 = 1$. There are two observations for updating the mean and variance. The model for the observations is:

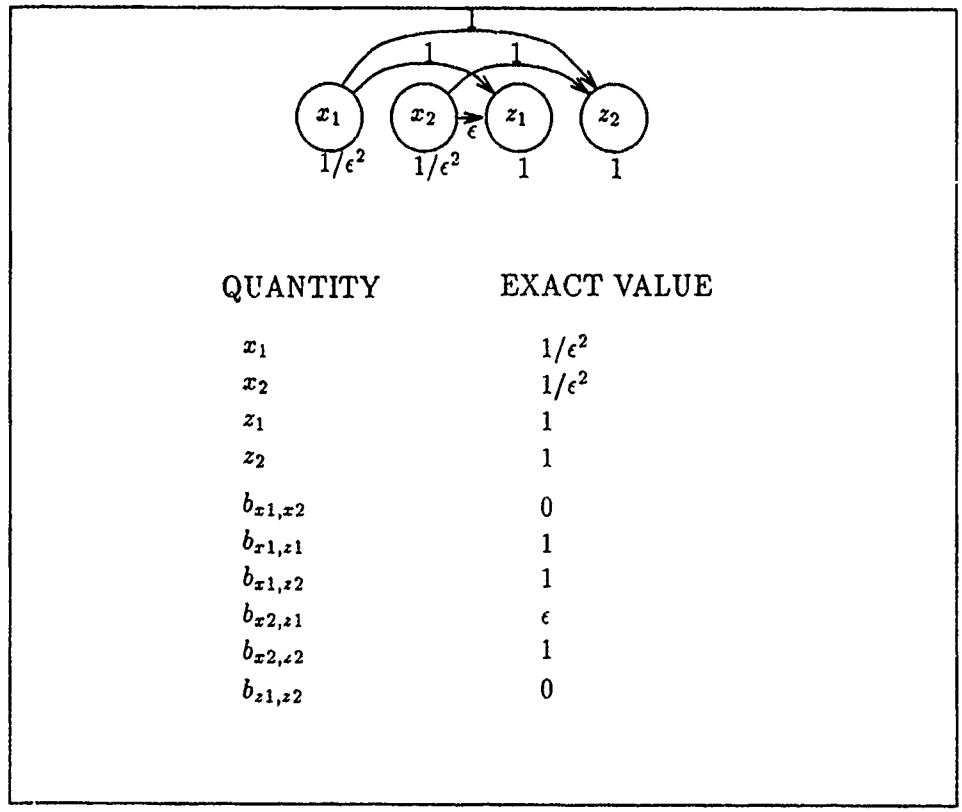
$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & \epsilon \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{223}$$

where $\mathbf{R} = \mathbf{I}$. Bierman presented the results for the conventional Kalman filter using scalar updates, the U-D factorization method, the Potter covariance square root method, and the stabilized Kalman filter. The exact and rounded values of the U-D factorization method will be included here for comparison.

Figure 47 shows the influence diagram implementation of this example. The labels on the diagram are the rounded values, with the exact values shown in a table on the diagram itself. In Figure 47, the exact values and the rounded values are the same.

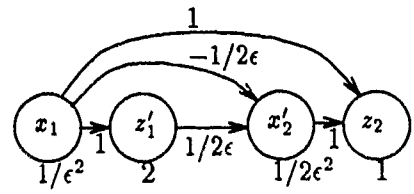
The objective of the influence diagram operations will be to move the z_1, z_2 vector to the beginning of the ordered sequence. One way is to move z_1 to the beginning, then move z_2 next to it. Using this process, the first operation will be to reverse the arrow from x_2 to z_1. Figure 48 shows the results of this reversal, where again, the rounded values are the same as the exact. There is a potential problem in this diagram because the path coefficient from x_1 to x_2 is still zero, but it is computed with regression coefficients that have large relative differences in magnitude. There are no errors in rounding yet, but they are imminent.

The second operation will be to reverse the arrow from x_1 to z_1 so that z_1 will be unconditional as shown in Figure 49. The rounding errors have caused cancellation of all significant digits in the path coefficient from z_1 to x_2. It is now incorrectly calculated as $1/2\epsilon + (1)(-1/2\epsilon) = 0$. The true value should be $1/2\epsilon + (\frac{1}{(2\epsilon^2+1)})(-1/2\epsilon)$, a small number calculated by the difference of two large



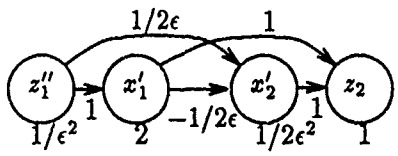
QUANTITY	EXACT VALUE
x_1	$1/\epsilon^2$
x_2	$1/\epsilon^2$
z_1	1
z_2	1
$b_{x1,x2}$	0
$b_{x1,z1}$	1
$b_{x1,z2}$	1
$b_{x2,z1}$	ϵ
$b_{x2,z2}$	1
$b_{z1,z2}$	0

Figure 47. Influence Diagram Formulation of Bierman's Example



QUANTITY	EXACT VALUE
x_1	$1/\epsilon^2$
x_2'	$1/2\epsilon^2$
z_1'	2
z_2	1
$b'_{x1,x2}$	$-1/2\epsilon$
$b'_{x1,z1}$	1
$b_{x1,z2}$	1
$b_{z1,x2}$	$1/2\epsilon$
$b_{x2,z2}$	1
$b_{z1,z2}$	0

Figure 48. Influence Diagram After Exchanging Nodes x_2 and z_1



QUANTITY	EXACT VALUE
x_1'	$2/(2\epsilon^2+1)$
x_2'	$1/2\epsilon^2$
z_1''	$2+1/\epsilon^2$
z_2	1
$b'_{x1,x2}$	$-1/2\epsilon$
$b_{z1,x1}$	$1/(2\epsilon^2+1)$
$b_{x1,z2}$	1
$b_{z1,x2}$	$1/2\epsilon$
$b_{x2,z2}$	1
$b_{z1,z2}$	0

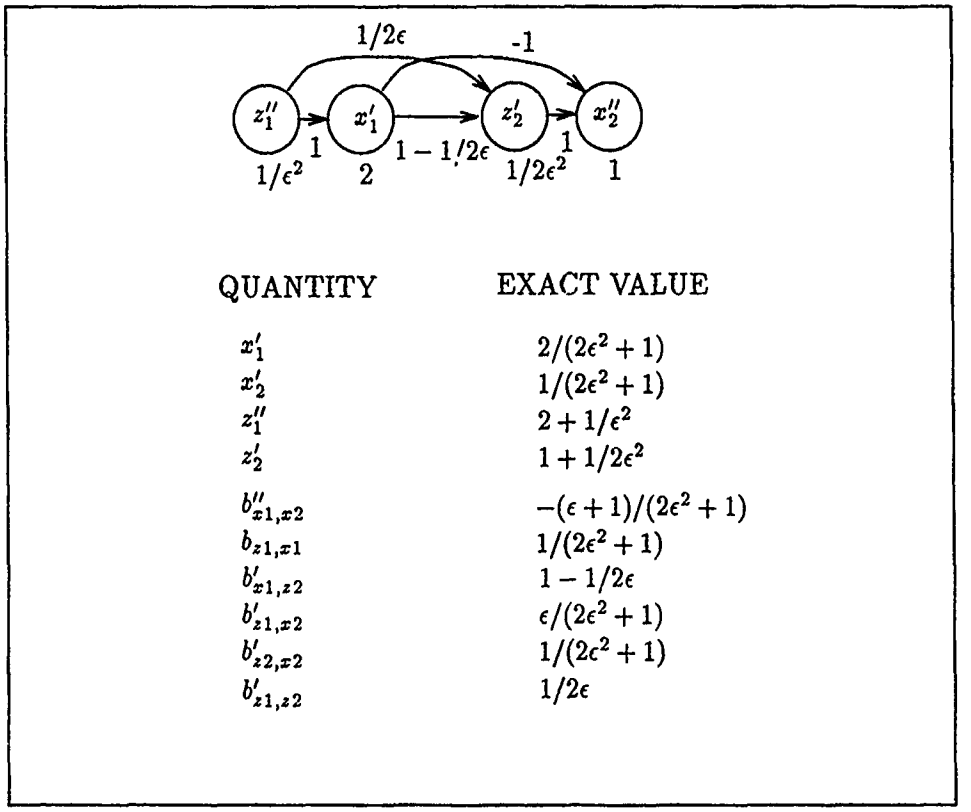
Figure 49. Influence Diagram After Exchanging Nodes x_1 and z_1

numbers. This error will be carried forward to all further operations and will eventually cause the Kalman gain to be calculated in error.

The third operation will be to reverse the arrow from x_2 to z_2, while the final operation will be to reverse the arrow from x_1 to z_2. These operations are shown in Figure 50 and Figure 51 respectively.

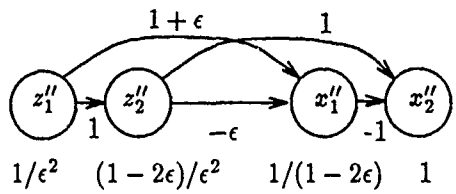
The path coefficients from the vector $\mathbf{z}$ to the vector $\mathbf{x}$ are equivalent to a Kalman gain matrix. The exact values of these path coefficients are essentially the same as the Kalman gains computed by Bierman, with the exception of $u_{z1,x2}$. The influence diagram incorrectly calculates this number as zero. These values are shown in Table 7.

This example shows the types of rounding errors that can occur in the influence diagram. Specifically, the error occurred because of cancellation of all significant digits in the computation



QUANTITY	EXACT VALUE
x_1'	$2/(2\epsilon^2+1)$
x_2'	$1/(2\epsilon^2+1)$
z_1''	$2+1/\epsilon^2$
z_2'	$1+1/2\epsilon^2$
$b''_{x1,x2}$	$-(\epsilon+1)/(2\epsilon^2+1)$
$b_{z1,x1}$	$1/(2\epsilon^2+1)$
$b'_{x1,z2}$	$1-1/2\epsilon$
$b'_{z1,x2}$	$\epsilon/(2\epsilon^2+1)$
$b'_{z2,x2}$	$1/(2\epsilon^2+1)$
$b'_{x1,z2}$	$1/2\epsilon$

Figure 50. Influence Diagram After Exchanging Nodes x_2 and z_2



QUANTITY	EXACT VALUE
x_1''	$(2\epsilon^2+1)/(1-2\epsilon+4\epsilon^2+2\epsilon^4)$
x_2''	$1/(2\epsilon^2+1)$
z_1''	$2+1/\epsilon^2$
z_2''	$(1-2\epsilon+4\epsilon^2+2\epsilon^4)/(\epsilon^2+2\epsilon 4)$
$b''_{x1,x2}$	$-(\epsilon+1)/(2\epsilon^2+1)$
$b'_{z1,x1}$	$(\epsilon^2-\epsilon+1)/(1-2\epsilon+4\epsilon^2+2\epsilon^4)$
$b_{z2,x1}$	$(2\epsilon^2-\epsilon)/(1-2\epsilon+4\epsilon^2+2\epsilon^4)$
$b_{z1,x2}$	$\epsilon/(2\epsilon^2+1)$
$b_{z2,x2}$	$1/(2\epsilon^2+1)$
$b_{z1,z2}$	$(\epsilon+1)/(2\epsilon^2+1)$

Figure 51. Influence Diagram After Exchanging Nodes x_1 and z_2

Table 7. Comparison of Computed Path Coefficients (Kalman Gains)

Quantity	Exact Value	U-D	Influence Diagram
$u_{z1,x1}$	$1/(1+2\epsilon^2)$	1	1
$u_{z1,x2}$	$\epsilon/(1+2\epsilon^2)$	ϵ	0
$u_{z2,x1}$	$(2\epsilon^2-\epsilon)/(1-2\epsilon+4\epsilon^2+2\epsilon^4)$	$-\epsilon$	$-\epsilon$
$u_{z2,x2}$	$(1-\epsilon)/(1-2\epsilon+4\epsilon^2+2\epsilon^4)$	$(1-\epsilon)/(1-2\epsilon) \approx 1+\epsilon$	$1+\epsilon$

of one of the regression coefficients. It also shows that an update using z_1 may be inaccurate, but an update at z_2 may be large enough to cause the relative error in x_2 to be small.

There are actually two equations in the influence diagram algorithm that can cause cancellation of significant digits. One is the equation $b'_{kj} = b_{kj} + b_{ki}b_{ij}$, used for calculating the regression coefficient from predecessor nodes to the node being made less conditional. The other equation is $b'_{ki} = b_{ki} - b'_{kj}b_{ij}$, used for calculating the regression coefficient from predecessor nodes to the node being made more conditional. Errors caused by the first equation were shown in the previous example. The second equation can result in large relative error as well, but its overall effect will be less. This is because the new regression coefficient, no matter how inaccurate, is only one of at least two (probably more) regression coefficients leading to that node as it becomes conditioned on more of the other nodes. The effect of these other predecessor coefficients tend to reduce the relative effect of the coefficient in error.

Although this example demonstrates rounding errors in the influence diagram, it also shows the capability of the influence diagram to avoid such problems. If the order of node reversal is changed, then the problem of canceling significant digits during arrow reversal is minimized. However, to the author's knowledge, there are no guidelines as to the "best" order for an influence diagram for avoiding numeric difficulties.

The original order of the nodes x_1, x_2, and z_1 is shown in Figure 52 along with an alternative ordering. If the second ordering is used, the problem encountered in the previous example does not occur. Instead of proceeding through the entire example again, Figure 53 shows the results after exchanging to the point that node z_1'' is unconditional. Again, the diagram uses rounded

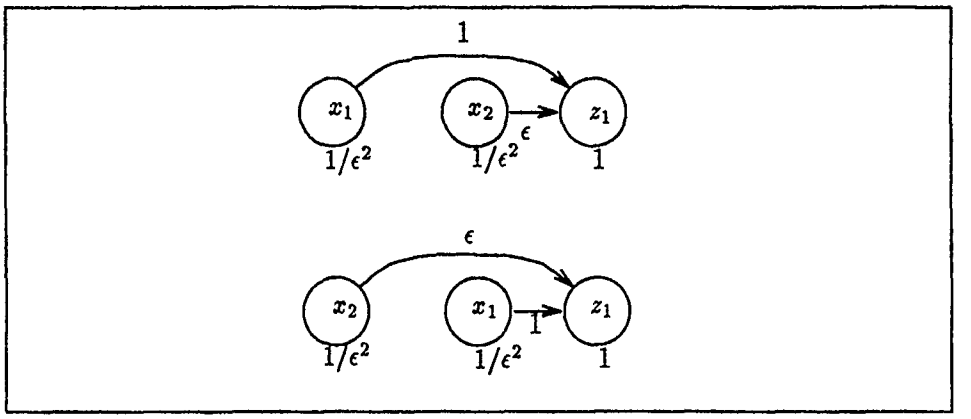
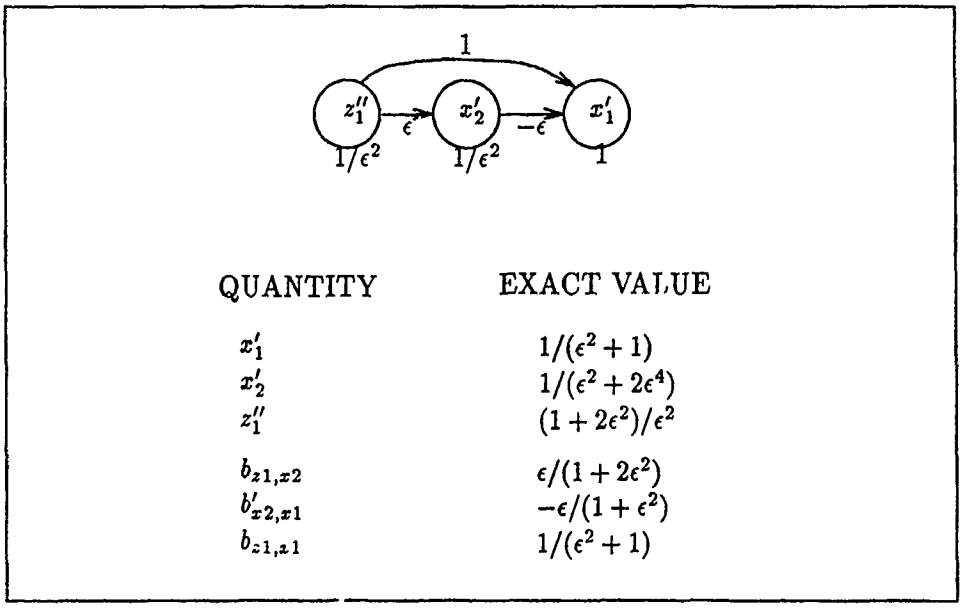


Figure 52. Two Different Orderings of the First Three Nodes in Bierman's Example

values but the exact values are included for comparison. The rounded value of the path coefficient from z_1'' to x_2' is exactly the same as the U-D filter. The rest of the calculations for making z_2 unconditional are not shown here because the error in the earlier calculation was the cause of the problems at the end. There are no more significant rounding errors in the rest of the problem. The important conclusion is that, if significant cancellation of significant digits occurs, it may be possible to reorder the sequence of node reversals and improve the numeric results.



QUANTITY	EXACT VALUE
x_1'	$1/(\epsilon^2+1)$
x_2'	$1/(\epsilon^2+2\epsilon^4)$
z_1''	$(1+2\epsilon^2)/\epsilon^2$
$b_{z1,x2}$	$\epsilon/(1+2\epsilon^2)$
$b'_{x2,x1}$	$-\epsilon/(1+\epsilon^2)$
$b_{z1,x1}$	$1/(\epsilon^2+1)$

Figure 53. Improved Numerical Characteristics with Different Node Order in Bierman's Example

It may also be possible to choose an order for the vector $\mathbf{x}$ during problem setup that is optimum for numeric results. Although possible, this is not yet a satisfactory approach either. The numeric properties of the influence diagram are not well understood. Except for this research paper, there is little insight into what circumstances yield poor numeric results. This is one of the key reasons that further research is needed on the numeric properties of the influence diagram. The need for more research is reflected in the recommendations at the end of this pape.

5.3.3 Potential Errors in the U-D Filter. The U-D factored form of the filter can also have cancellation of significant digits to the point where all remaining significant digits are in error. One set of circumstances that causes such cancellation of digits is when the off-diagonal terms of the $\mathbf{U}$ matrix are reduced by a measurement update. The algorithm for computing the off-diagonal terms of the $\mathbf{U}$ matrix requires subtraction just as the influence diagram algorithm for recomputing regression coefficients from predecessor nodes.

For example, modify Bierman's example in the previous subsection so that the measurement model is

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{224}$$

This measurement matrix creates the circumstances described earlier for cancellation of significant digits in the U-D filter. The scalar update with the first row of $\mathbf{H}$ results in the (1,2) term of the $\mathbf{U}$ matrix to be $-1/2\epsilon$, a very large term. It is normal for the off-diagonal terms of the $\mathbf{U}$ matrix to be nonzero, but in this example, the situation was created purposefully.

When the update is made using the second row of $\mathbf{H}$, the off-diagonal term of $\mathbf{U}$ decreases. In this case, the subtraction results in the cancellation of all significant digits, and the new $\mathbf{U}_{12}$ term is incorrectly calculated as zero. The calculations just described for the U-D filter are as follows.

Let the initial conditions be:

$$\mathbf{D} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{225}$$

For the first scalar update, use $\mathbf{H} = [\epsilon \ \ 1]$.

$f = \mathbf{U}^T\mathbf{H}^T$, $f_1 = \epsilon$, $f_2 = 1$
$v_1 = 1/\epsilon$, $v_2 = 1/\epsilon^2$
$a_0 = R = 1$

for k=1
$a_1 = a_0 + f_1 v_1 = 2$
$D_{11}^+ = D_{11}^- a_0/a_1 = 1/2\epsilon^2$
$b_1 = 1/\epsilon$

for k=2, j=1
$a_2 = a_1 + f_2 v_2 = 2 + 1/\epsilon^2 \approx 1/\epsilon^2$
$D_{22}^+ = D_{22}^- a_1/a_2 = 2$
$b_2 = v_2 = 1/\epsilon^2$
$p_2 = -f_2/a_1 = -1/2$
$U_{12}^+ = U_{12}^- + b_1 p_2 = -1/2\epsilon$
$b_1 = b_1 + U_{12}^- v_2 = 1/\epsilon$

Repeat for the next scalar update, use $\mathbf{H} = [1 \ \ 0]$.

$f = \mathbf{U}^T\mathbf{H}^T$, $f_1 = 1$, $f_2 = -1/2\epsilon$
$v_1 = 1/2\epsilon^2$, $v_2 = -1/\epsilon$
$a_0 = R = 1$

for k=1
$a_1 = a_0 + f_1 v_1 = 1 + 1/2\epsilon^2 \approx 1/2\epsilon^2$
$D_{11}^+ = D_{11}^- a_0/a_1 = 1$
$b_1 = 1/2\epsilon^2$

for k=2, j=1
$a_2 = a_1 + f_2 v_2 = 1/2\epsilon^2 + 1/2\epsilon^2 = 1/\epsilon^2$
$D_{22}^+ = D_{22}^- a_1/a_2 = 1$
$b_2 = v_2 = -1/\epsilon$
$p_2 = -f_2/a_1 = \epsilon$
$U_{12}^+ = U_{12}^- + b_1 p_2 = -1/2\epsilon + 1/2\epsilon = 0$
$b_1 = b_1 + U_{12}^- v_2 = 1/2\epsilon^2 + 1/2\epsilon^2 = 1/\epsilon^2$

In this example, the U_{12}^{+} term is incorrectly calculated as zero. The cancellation of significant digits is exactly the same kind of error demonstrated in the previous example using the influence diagram. Since both the influence diagram and the U-D filter implementations are factored forms of the covariance matrix, it is reasonable to expect the errors to be similar.

5.3.4 Numeric Analysis Generalizations. Some generalizations can be made about some of the situations that cause numeric problems in both the U-D filter and the influence diagram. These generalizations are based on simple examples such as demonstrated earlier, and on a very limited number of experiments, using commercially available computer software. For both algorithms, these generalizations do not cover all of the possible conditions for numeric errors.

In an influence diagram, if only one state variable is affected by a measurement ($\mathbf{H}$ has only one nonzero element in a row), then the best numeric results occur when that state variable is the first one. Changing the unconditional variance of the first node in the ordered sequence is sufficient to account for the decreased variance due to the update. No other calculations are necessary, and there is less likelihood of numerical errors.

In matrix terms, let $\mathbf{H}$ be a row vector with the first element nonzero, and all other elements zero. Let this $\mathbf{H}$ matrix represent the measurement model for a Kalman filter update. The $\mathbf{U}^T\mathbf{D}\mathbf{U}$ factorization of the covariance matrix and the updated version of that covariance matrix will differ in only the first term of the diagonal matrix. There will be no change in the $\mathbf{U}$ matrix and there will be low relative error in computing the updated covariance. Therefore, a good way to avoid cancellation of significant digits during update is to order the variables in the state estimate such that the updated variables are first.

By analogy, the U-D filter has the opposite problem. When the first variable is updated, as was done in the modified version of Bierman's example, the $\mathbf{U}$ matrix terms were significantly in error. If the ordering of the updated variables were reversed such that $\mathbf{H} = [0 \;\; 1]$, then no change would have occurred to the $\mathbf{U}$ matrix. To minimize cancellation of significant digits during update

in the U-D factored form of the filter, the variables in the state estimate should be ordered such that the updated variables are last.

5.4 Chapter Conclusions

This chapter used the arithmetic properties of the influence diagram and their relationship with matrix operations to analyze the numeric properties of the influence diagram implementation of the discrete-time filter. It was shown that the influence diagram computes the conditional covariance of a random vector as a series of scalar operations. This resulted in better numeric properties than the Kalman filter conditional covariance matrix equation. It was also shown that the influence diagram uses a stable algorithm to reverse the node order and calculate conditional variances.

It is difficult to put a strict bound on the errors that might be caused by the influence diagram algorithm. Even though the factored forms of the matrices may have bounded errors, there is no guarantee that all elements of the matrix will have a bounded relative error. However, because of the triangular form of the matrices, the errors will usually be small.

Finally, examples were purposefully constructed that showed the worst case errors that might occur in both the U-D filter and the influence diagram discrete-time filter. It was shown that the conditions for error in both filter implementations are similar. Even though experimental evidence is limited, the initial indications are that the U-D filter and the influence diagram have almost identical error properties.

VI. Summary

6.1 Conclusions

The purpose of this research was to evaluate the influence diagram as an alternative method for the discrete-time filter. Kenley's doctoral dissertation [6] laid the groundwork by proving the influence diagram could be used for jointly Gaussian random variables. He also proved the basic matrix relationships and demonstrated the influence diagram for discrete-time filtering. This research built on Kenley's original work. Some of the more important results from this thesis are summarized here.

The influence diagram implements a factored form of the Kalman filter just as does the U-D filter. These two implementations are essentially mirror images. One is related to the lower triangular version of the Cholesky decomposition, and the other is related to the upper triangular version. Because of this similarity, it could be expected that the efficiency and numerical properties of the two algorithms are very similar.

The influence diagram implementation was shown to have some advantages over the U-D filter. Specifically, it can be more efficient in terms of computational loading. It also lends itself to parallel processing architectures with a resulting reduction in processing time.

Based on theory and limited experimentation, the influence diagram probably has numeric properties equivalent to those of the U-D filter. It appears that there are conditions under which either one may be better than the other, but it is unlikely that either is inherently better or worse than the other. The one advantage of the influence diagram is that almost any numeric error can be traced to one equation.

Perhaps the most important advantage is intangible. The influence diagram is a graphic tool that gives the user tremendous insight into the meaning of the numerical operations. The calculations are much easier to understand than the comparable calculations for the U-D filter.

This is because the numeric values have physical meaning as conditional variances and regression coefficients.

6.2 Other Applications

This research only described the characteristics of the influence diagram when applied to discrete-time filtering. There are potential uses in fault detection and hypothesis testing, in areas where the inverse covariance (information) matrix is better suited, in optimal smoothing, optimal control, and parameter estimation. Furthermore, the discrete probability version of the influence diagram can be used for probabilistic analysis of Markov processes.

6.2.1 Fault Detection/Hypothesis Testing. Assume that measurements are in the form of scalar updates as discussed earlier in this thesis. As each node of $\mathbf{z}(t_i)$ is moved to the beginning of the ordered sequence, the remaining nodes of $\mathbf{z}(t_i)$ are conditioned on it. As seen previously, the variance of a conditioned node is smaller than the unconditional variance of the same node. It is this smaller variance which is useful for failure detection. This method is the similar to the Kalman filter approach which uses $\mathbf{H}(t_i)\mathbf{P}(t_i)\mathbf{H}^T(t_i) + \mathbf{R}(t_i)$ as the predicted variance matrix of the variables of $\mathbf{z}(t_i)$ [7:230]. The influence diagram incorporates each scalar update to make the variance smaller on succeeding measurements.

This property has potential use in hypothesis testing and fault detection. It would be best to incorporate the first measurements from sensors that are reliable, or that are measuring parameters which are not part of the hypothesis. The later measurements would come from sensors that are more likely to fail, or that are measuring parameters that are part of a hypothesis. In both cases, the tighter bounds of the conditional variance gives later measurements more discriminating power.

6.2.2 Inverse Covariance. The influence diagram is based on determining the conditional mean and variance of Gaussian random variables. It was shown earlier that this conditional variance

has a direct correspondence to the inverse covariance matrix. Instead of using the conditional variance to describe each variable, the "information level", taken to be the inverse of the variance, could be used [7:pp. 238–241]. The influence diagram algorithm would be modified to use the inverse of the variance. The most conditional node in the influence diagram (the last one) would be considered as having the most information from predecessors. The information level of this last node would be identical to the corresponding diagonal term of the inverse covariance matrix.

One application of such an approach is optimal smoothing. The influence diagram could be used to determine the optimal estimate based on all previous measurements. The inverse covariance form could be used to incorporate information from later measurements. For the influence diagram, this becomes nothing more than reversing all arrows from later measurements to point to the desired estimate. The desired state estimate then becomes conditioned on the later measurements. The combination of the two becomes the optimal estimate of the states based on all measurements, both previous and later.

6.2.3 Optimal Control. The influence diagram came from the field of decision analysis. The Gaussian influence diagram also allowed decisions based on linear, quadratic cost functions of appropriate variables. This aspect of the influence diagram was described in detail in Kenley's original work, but has not been addressed in this thesis. It is obvious that decisions based on linear system models and quadratic costs on jointly Gaussian random variables is equivalent to LQG control. It remains to be seen whether the influence diagram offers efficient implementation of such control inputs.

Another approach to optimal control relies on the dual nature of the optimal full-state feedback controller and the Kalman filter. Because the influence diagram implements a factored form of the Kalman filter, it could also be used to implement a factored form of an LQG optimal controller.

6.2.4 Parameter Estimation. Throughout this thesis, the coefficients on the arrows between nodes have been called regression coefficients. In the Kalman filter, it is assumed that these coefficients are known elements of either the state transition matrix or the measurement matrix. As such, these regression coefficients are not random variables.

In practice, it may be true that one or more of these coefficients is unknown or uncertain. Under linear system model assumptions, using known inputs and measured outputs, the value of such unknown or uncertain parameters may be estimated. If such analysis uses conventional least squares techniques, then determining the parameters is identical to determining regression coefficients. Such insights would be useful in system identification.

6.2.5 Discrete Probability Applications. In the Gaussian random variable influence diagram, it was the Markov nature of the random variable that permitted a complete description of the density function at a given time, based on the density function at a previous time. The same is true of a Markov process with discrete probability distributions. If the nodes of the discrete-time filter described in this thesis are replaced with discrete-time, discrete-probability-distribution random variables, then they become the description of a Markov process. Observations of such a process can be probabilistic as well. Although the mathematics are not as easy as for the simple Gaussian case, such an approach would be a Bayesian method of estimating the states of a Markov process, based on uncertain observations.

6.3 Recommendations

There are several areas left unexplored in this research. Probably the most useful research would be a definitive study of the conditions for best and worst numeric performance. Such a study would probably require both implementations running on one computer. The computer should also be capable of varying the number of significant bits used internally for storage and calculations.

Another useful study would be an attempt to decrease the numeric errors of the influence diagram. The type of errors that occur in the influence diagram are similar to the type of errors that occur in the U-D filter. However, as shown earlier, the order of the operations may affect the accuracy of the influence diagram operations. An example of a way to minimize errors is to identify them as they occur. In a digital computer, this could be done by monitoring the two equations responsible for cancellation of significant digits. If either of them results in a significant decrease in value (e.g. $|b'_{kj}| \leq |b_{kj}/1000|$), then the current calculations would be halted, and the algorithm would proceed using a different order for further reversals. Such a computer program would also need to identify those situations where the regression coefficient is unavoidably small, or where it may be intentionally zero.

It may be useful to analyze the matrix operations for the influence diagram in more detail. It may be that rows and columns of the factored covariance matrix may be reordered more efficiently than the "two at a time" method of the influence diagram. If that were that case, then whole blocks of rows could be reordered. Such an operation would be much more efficient than the current method.

As mentioned earlier, there is a relationship between decision and control theory. The influence diagram can be used to make decisions, based on jointly Gaussian random variables and linear system models. These conditions are also the the assumptions needed for LQG control. The two methods should be equivalent. Even though this seems reasonable, this equivalence has not yet been proven. It would be very useful to compare the influence diagram and LQG control. Such a comparison would need to prove the relationship of the two methods, and to evaluate their efficiency and numerical properties.

Bibliography

1. Anderson, T. W. *An Introduction to Multivariate Statistical Analysis*. New York: John Wiley and Sons, Inc., 1984.
2. Bierman, Gerald J. "A Comparison of Discrete Linear Filtering Algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, *AES-9*(1):28–37 (1973).
3. Bierman, Gerald J. *Factorization Methods of Discrete Sequential Estimation*. New York: Academic Press, 1977.
4. Bierman, Gerald J. and C. L. Thornton. "Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study," *Automatica*, *13*:23–35 (1977).
5. Kaminski, P. G., et al. "Discrete Square Root Filtering: A Survey of Current Techniques," *IEEE Transactions on Automatic Control*, *AC-16*(6):727–735 (1971).
6. Kenley, C. Robert. *Influence Diagram Models with Continuous Variables*. PhD dissertation, Stanford University, 1986.
7. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Vol 1*. San Diego, California: Academic Press, 1979.
8. Mood, Alexander M. and Franklin A. Graybill. *Introduction to the Theory of Statistics*. New York: McGraw-Hill Book Company, Inc., 1963.
9. Raiffa, H. *Decision Analysis*. Reading, Mass.: Addison-Wesley, 1968.
10. Schacter, Ross D. "Evaluating Influence Diagrams," *Operations Research*, *34*(6):871–882 (1986).
11. Schacter, Ross D. "Probabilistic Inference and Influence Diagrams," *Operations Research*, *36*(4):589-604 (1988).
12. Schacter, Ross D. and C. Robert Kenley. "Gaussian Influence Diagrams," *Management Science*, *35*(5):527–550 (1989).
13. Stewart, G. W. *Introduction to Matrix Computations*. Orlando, Florida: Academic Press, 1973.
14. Tatman, Joseph A. *Decision Processes in Influence Diagrams: Formulation and Analysis*. PhD dissertation, Stanford University, 1985.
15. Wilkinson, J. H. *Rounding Errors in Algebraic Processes*. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1963.
16. Wilkinson, J. H. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford. Oxford University Press, 1965.
17. Yule, G. U. "On the Theory of Correlation for any Number of Variables, Treated by a New System of Notation," *Proceedings of the Royal Society of London*, pages 182–193 (1907). Series A.79.

Appendix A. *Appendix A: Operations Count*

This appendix is an explanation of the method used to calculate the number of operations required to implement the influence diagram algorithm for discrete time filtering. Assume an influence diagram of the form of Figure 42 in Chapter 3. The new regression coefficients between $\mathbf{x}(t_{i-1})$ and $\mathbf{x}(t_i)$ must be modified using

$$\Phi_q = (\mathbf{I} - \mathbf{B}_q)^T \Phi(t_i, t_{i-1}) \tag{226}$$

where $\mathbf{B}_q$ is the influence diagram $\mathbf{B}$ matrix corresponding to the influence diagram factorization of $\mathbf{G}_d(t_{i-1})\mathbf{Q}_d(t_{i-1})\mathbf{G}_d^T(t_{i-1})$ and $\mathbf{I}$ is the identity matrix of appropriate dimension. The term $(\mathbf{I} - \mathbf{B}_q)$ requires no operations because the diagonal terms of $\mathbf{B}_q$ are zero.

Because the matrix $(\mathbf{I} - \mathbf{B}_q)^T$ is a lower triangular unit matrix, only the nonzero additions and the non-unity multiplications need to be counted. For n-dimensional matrices, the number of multiplications and additions is:

$$\sum_{r=0}^{n-1} n = \frac{n(n-1)}{2} \tag{227}$$

After computing the new regression coefficients of Φ_q, the influence diagram of Figure 43 can be drawn.

Now determine the number of operations needed to remove a node. Assume a node in the middle of an ordered sequence is to be moved to the end of the ordered sequence and removed. If i is the predecessor node and j is the successor node, then the equation for calculating the variance of the new predecessor after reversal is:

$$v_j' = v_j + b_{ij}^2 v_i \tag{228}$$

This equation requires 3 multiplications and 1 addition.

Assume that both nodes have the set of predecessors K, where the elements of K are designated k. The regression coefficients from each node k to the new predecessor are calculated using:

$$b'_{kj} = b_{kj} + b_{ki}b_{ij} \tag{229}$$

This equation requires 1 multiplication and 1 addition.

For the new successor, the equation for calculating the new variance is modified slightly to increase the efficiency. The new successor's variance is calculated using:

$$v_{ratio} = \frac{v_i}{v'_j} \tag{230}$$

$$v'_i = v_j v_{ratio} \tag{231}$$

The new regression coefficient between the two nodes is calculated by:

$$b'_{ji} = b_{ij} v_{ratio} \tag{232}$$

These equations require 1 division and 2 multiplications.

For the same set of predecessors K, the equation for modifying the regression coefficients to the new successor is:

$$b'_{ki} = b_{ki} - b'_{kj}b'_{ji} \tag{233}$$

This equation requires 1 multiplication and 1 addition.

When a node is moved to the end of an ordered sequence and removed, then it must be reversed with each successor node in the sequence. After the last reversal, the subject node becomes

a nuisance node and is removed from the diagram. There is no need to calculate its new conditional variance or the values of the regression coefficients for it. This implies that, during the last reversal, only Equation (228) and Equation (229) must be calculated.

Assume now that the node to be removed has n successors and m predecessors. Removal of the node requires n-1 reversals using Equations (228) through (233), and one removal using only Equation (228) and Equation (229).

In order to make the calculations simpler, calculate the number of operations due to Equation (229) and Equation(233) later. The number of operations to to remove one node by making n-1 reversals and one removal, using the remaining equations, is:

$$\begin{aligned} \text{multiplications:} \quad & 5n-2 \\ \text{additions:} \quad & n \\ \text{divisions:} \quad & n-1 \end{aligned}$$

Now calculate the number of operations due to Equation (233). The node being removed and the node with which it is exchanged both have the same number of predecessors. At first, there are m predecessors, then $m+1$, $m+2$, and so on until the last exchange has $m+n-1$ predecessors. Therefore, the number of times Equation (233) will be used is

$$\sum_{r=0}^{n-1}(m+r) = mn + \frac{n(n-1)}{2} \tag{234}$$

There is one multiplication and one addition due to Equation (233) each time it is used. Equation (229) is used $m+n-1$ times less than Equation (233) because the regression coefficients are not calculated after the last reversal. Together, the two equations require $2(mn+n(n-1)/2)-(m+n-1)$ multiplications and additions to remove one node.

Now assume that there are n nodes to be removed, as in Figure 43. Instead of m predecessors, the first node removed will begin with $n-1$ predecessors. The second node will begin with $n-2$ predecessors. The last node will have no predecessors. The number of operations to remove n nodes then becomes:

$$\begin{aligned}
\text{multiplications:} \quad & (5n-2)n + \textstyle\sum_{m=n-1}^{0} 2(mn + n(n-1)/2) - (m+n-1) \\
\text{additions:} \quad & n^2 + \textstyle\sum_{m=n-1}^{0} 2(mn + n(n-1)/2) - (m+n-1) \\
\text{divisions:} \quad & (n-1)n
\end{aligned}$$

The result of these operations is the second influence diagram in Figure 43.

The next operation is to condition the nodes of $\mathbf{x}(t_i)$ on the nodes of $\mathbf{z}(t_i)$. This was depicted in Figures 16 and 17. No operations are required to move directly from Figure 16 to the first diagram of Figure 17 because $\mathbf{R}(t_i)$ is assumed to be diagonal. The variances of the nodes of $\mathbf{z}(t_i)$ in Figure 17 are identical to the variances of the nodes of $\mathbf{v}(t_i)$ in Figure 16, which were also the diagonal terms of the $\mathbf{R}(t_i)$ matrix.

If there are p nodes in the measurement, then they must be moved so that they are at the beginning of the ordered sequence. These operations are depicted in Figure 17. Again, calculate the number of operations due to Equation (228), Equation (231), and Equation (232) first, and calculate the operations due to predecessor nodes later. Each node of $\mathbf{z}(t_i)$ must be moved past the n nodes of $\mathbf{x}(t_i)$. This time, no nuisance nodes will be removed. Moving one node requires:

$$\begin{aligned}
\text{multiplications:} \quad & 5n \\
\text{additions:} \quad & n \\
\text{divisions:} \quad & n
\end{aligned}$$

Now calculate the operations due to Equation (229) and Equation (233). When the first node of $\mathbf{z}(t_i)$ is moved to unconditional position, then number of applications of each of these equations

is:

$$\sum_{m=n-1}^{0} m \tag{235}$$

The second node of $\mathbf{z}(t_i)$ starts off with one more predecessor than the first node. The number of operations required to move it up n positions is:

$$\sum_{m=n-1}^{0} m+1 \tag{236}$$

Each node of $\mathbf{z}(t_i)$ is moved up n positions similarly. Because there are p of them, then the total number of applications of Equations (229) and (233) are:

$$\sum_{q=0}^{p-1}\Big(\sum_{m=n-1}^{0} m+q\Big) \tag{237}$$

This double sum adds to $np(n+p-2)/2$ applications of both Equation (229) and Equation (233). The total number of operations to move a vector of p nodes to the beginning of the ordered sequence is:

$$\begin{aligned} \text{multiplications:} \quad & np(n+p-2)+5np \\ \text{additions:} \quad & np(n+p-2)+np \\ \text{divisions:} \quad & np \end{aligned}$$

The means are propagated as described in Chapter 3. The vector update algorithm uses p additions to calculate the initial p residuals. The number of additions or multiplications to calculate the sum of the series of inner products and the new conditional means is:

$$\sum_{j=1}^{n} p+j-1 \tag{238}$$

This sum reduces to $n(n+2p-1)/2$ as given in Chapter 3.

The Kalman filter update equation given in Equation (108) and the measurement prediction of Equation (109) require a total of $n(n+p)$ multiplications and $(n-1)(n+p)$ additions. The total number of operations needed to update the conditional means is the sum of all these operations counts. They are tabulated as:

$$\begin{aligned} \text{multiplications:} \quad & \tfrac{n(n+2p-1)}{2} + n(n+p) \\ \text{additions:} \quad & \tfrac{n(n+2p-1)}{2} + (n-1)(n+p) + p \end{aligned}$$

The total number of operations is the sum of all appropriate equations. These sums are:

$$\begin{aligned} \text{multiplications:} \quad & 2n^3 + n^2(p-0.5) + n(p^2+p-0.5) \\ \text{additions:} \quad & 2n^3 + n^2(p+3.5) + n(p^2+5p-1.5) \\ \text{divisions:} \quad & n(n+p-1) \end{aligned}$$

Appendix B. *Appendix B: Operations Count, Parallel Processing*

This appendix is an explanation of the method used to calculate the number of operations required to implement the influence diagram algorithm for discrete time filtering, assuming parallel processing. Assume that n nodes will be moved past another group of n nodes and removed. Such was the case shown in Figure 44 in which $n = 3$. The equations are the same as those given in Appendix A.

Begin with the assumption that there is a single processor dedicated to the task of calculating all necessary equations for each of the n nodes being moved rightwards. For example, in Figure 44, there would be three processors, one assigned to each of the first three nodes. Between the first and second influence diagrams, a single processor would reverse nodes 3 and 4. Between the second and third diagrams, one processor would be used to exchange nodes 3 and 5, while another would be used to exchange 2 and 4. Moving from the third to the fourth diagram requires all three processors; one processor removes node 3, another reverses nodes 2 and 5, while the third reverses nodes 1 and 4. Moving to the fifth diagram requires only two processors, one to remove node 2 and another to reverse nodes 1 and 5. Finally, only a single processor is needed to remove node 1 and result in the last influence diagram.

During each time interval, one processor will take longer than the other processors to complete all of its calculations. The processor needing the most time will be the one with the most operations to complete. In the example of Figure 44, the processor assigned to node 3 will have more calculations during the first two time intervals because it has more predecessor regression coefficients. During the next time interval, node 2's processor has the most calculations because node 3 is being removed and its processor has less equations. Similarly, node 1's processor has the most calculations during the next time interval. During the last time interval, only one processor is operating.

The number of operations required to update the conditional means and to calculate $\mathbf{\Phi}_q = (\mathbf{I} - \mathbf{B}_q)^T \mathbf{\Phi}(t_i, t_{i-1})$ were given in Appendix A. Because of this, in the following computations, only the operations needed to manipulate the influence diagram will be counted. Also, the only operations counted at each time interval will be those of the processor taking the longest time.

As before, in Appendix A, calculate the number of operation due to Equations (228), (231), and (232) first. Calculate the number of operations due to Equations (229) and (233) later. At each time interval except the last, there is at least one processor reversing a pair of nodes. Equations (228), (231), and (232) require 5 multiplications, 1 additions, and 1 division as a minimum in each time interval. There are $(n-1)+(n-1)$ such reversals as a vector of n nodes is removed. Additionally, the last time interval only removes one node and requires require 3 multiplications and 1 addition.

The next step is to calculate the number of predecessors for the node with the most operations. In every case, the node with the most operations is the node being reversed (not removed) that is furthest to the right. During the first $n-1$ successive time intervals, this node has $n-1, n, n+1, \ldots, 2n-3$ predecessors. During the next $n-1$ successive time intervals, this node has $2n-4, 2n-5, \ldots, n-1, n-2$ predecessors. The number of predecessors is given by:

$$(\sum_{q=1}^{n-1} n-2+q)+(\sum_{q=1}^{n-1} n-3+q) = 3n^2-8n+5 \tag{239}$$

There are 2 multiplications and 2 additions associated with each predecessor. When the last node is removed, it only requires 1 multiplication and addition for each of $n-1$ predecessors.

For the removal of n nodes, the number of operations required is equivalent to the sum of operations required by the longest time interval. This is the same as adding all operations discussed

above. These numbers are:

$$\begin{aligned} \text{multiplications:} \quad & 6n^2 - 5n + 2 \\ \text{additions:} \quad & 6n^2 - 13n + 8 \\ \text{divisions:} \quad & 2n - 2 \end{aligned}$$

When two vectors are being reversed, and neither is being removed, the calculations are similar. This time, reverse an n-dimensioned vector on the left of the diagram with a p-dimensioned vector on the right. This is the operation occurring when $\mathbf{x}(t_i)$ is conditioned on $\mathbf{z}(t_i)$. The node requiring the most operations is the node being reversed that is furthest to the right. It is not in the same position as in the previous calculations because there are no nodes being removed. During the first $p-1$ time intervals there are $n-1, n, n+1, \ldots, p+n-3$ predecessors for this node. During the next n time intervals, there are $p+n-2, p+n-3, \ldots, p, p-1$ predecessors. There are 2 multiplications and 2 additions for each predecessor operation. The number of multiplications or additions due to predecessors is:

$$(\sum_{q=1}^{p-1} n - 2 + q) + (\sum_{q=1}^{n} p - 2 + q) = n^2 + n(4p-5) + p^2 - 5p + 4 \tag{240}$$

The total number of operations must include operations needed to update the conditional means, and to calculate $\mathbf{\Phi}_q = (\mathbf{I} - \mathbf{B}_q)^T \mathbf{\Phi}(t_i, t_{i-1})$. These were given before in Appendix A. The total of all operations becomes:

$$\begin{aligned} \text{multiplications:} \quad & 9n^2 + 6n(p-1) + p^2 + 1 \\ \text{additions:} \quad & 9n^2 - n(6p-19) + p^2 - 4p + 11 \\ \text{divisions:} \quad & 3n + p - 3 \end{aligned}$$

Vita

Major Frederick H. Zeitz III was born on March 28, 1956 in San Antonio, Texas to parents Frederick and Mary Lou Zeitz. He graduated from Bellevue High School, Bellevue, Nebraska in 1974. He attended the Air Force Academy from 1974 to 1978, and graduated in May 1978 with a Bachelor of Science in Electrical Engineering. In the process, he earned the award for being the outstanding cadet in Electrical Engineering. Following graduation at the Academy he went on to Undergraduate Pilot Training at Reese AFB, Texas in January 1979. After graduation in December 1979 he stayed at Reese AFB as a T-38 instructor until February 1982. He served as Accelerated Copilot Enrichment (ACE) Assistant Detachment Commander at Minot AFB, North Dakota as a T-38 instructor from February 1982 to April 1984. In April 1984 he attended F-111 and EF-111 qualification training at Mt. Home AFB, Idaho. His follow-on assignment was in the EF-111 at Upper Heyford, England assigned to the 42nd Electronic Combat Squadron. He also served as Wing Plans Officer attached to the 66th Electronic Combat Wing at Sembach AB, Germany. In May 1988, Major Zeitz was assigned to Aeronautical Systems Division (ASD) at Wright-Patterson AFB, Ohio and is currently the Chief of Tactical Requirements for ASD/XR. He attended AFIT as a part-time student, pursuing a Master's of Science degree in Electrical Engineering.

Permanent address: 913 Bayberry Ct.
Bellevue, NE. 68005