DYNAMIC SELECTION OF MODELS

A DISSERTATION

SUBMITTED TO THE PROGRAM IN MEDICAL INFORMATION SCIENCES

AND TO THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Geoffrey William Rutledge

March, 1995

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

Ross D. Shachter
Department of Engineering-Economic Systems
Stanford University
(Principal Advisor)

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

Lawrence M. Fagan
Department of Medicine
Stanford University

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

Edward H. Shortliffe
Department of Medicine
Stanford University

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Abstract

In this dissertation, I develop an approach to high-stakes, model-based decision making under scarce computation resources, bringing together concepts and techniques from the disciplines of decision analysis, statistics, artificial intelligence, and simulation. I develop and implement a method to solve a time-critical decision problem in the domain of critical-care medicine. This method selects models that balance the prediction accuracy and the need for rapid action.

Under a computation-time constraint, the *optimal model* for a model-based control application is the model that maximizes the tradeoff of model benefit (a measure of how accurately the model predicts the effects of alternative control settings) and model cost (a measure of the length of the model-induced computation delay). This dissertation describes a real-time algorithm that selects, from a graph of models (GoM), a model that is accurate and that is computable within a time constraint. The dynamic-selection-of-models (DSM) algorithm is a *metalevel reasoning strateg*y that relies on a DSM metric to guide the search through a GoM that is organized according to the simplifying assumptions of the models. The DSM metric balances an estimate of the probability that a model will achieve the required prediction accuracy and the cost of the expected model-induced computation

delay. The DSM algorithm provides an approach to automated reasoning about complex systems that applies at any level of computation-resource or computation-time constraint.

The DSM algorithm solves the model-selection problem for a *ventilator-management advisor* (VMA). A VMA is a computer program that applies patient-specific models of physiology to interpret intensive-care unit (ICU) data and to predict the effects of alternative proposed treatments. *VentPlan* is a prototype VMA that implements a simplified model of physiology to monitor postoperative ICU patients; however, VentPlan's model is unable to make accurate predictions for patients with complex physiologic abnormalities, such as the abnormalities that occur in patients with asthma or pulmonary embolus. *VentSim* is a more detailed and more accurate model of cardiopulmonary physiology that represents a wider range of physiologic abnormalities, but VentSim is too computationally complex for use in a real-time VMA. Simplifications of VentSim may make accurate predictions for specific patients; alternative simplifications of VentSim represent a range of tradeoffs of prediction accuracy and computation complexity.

I implement the DSM algorithm in *Konan*, a program that selects patient-specific models from a GoM of alternative simplifications of the VentSim model. Konan demonstrates that the DSM algorithm selects models that balance the competing requirements for high prediction accuracy and for low computation complexity; these model selections allow a VMA to make real-time decisions for the control settings of a mechanical ventilator.

Acknowledgments

I am grateful for the help and support of many people. I would like to thank my principal advisor, Ross Shachter, for his help and guidance, and for many interesting discussions. I thank Larry Fagan, who has guided my path through the informatics program at Stanford. I especially thank Edward Shortliffe for his advice, and for creating the environment for research in medical informatics at Stanford.

I thank Lewis Sheiner for his keen insight and unrelenting intellectual curiosity. His involvement in this research was critical, particularly in the early phase of the project. I am grateful to George Thomsen for developing the first physiologic model for VentPlan; I thank him also for the many academic and nonacademic adventures that we shared—and for being my best friend. I thank Brad Farr for developing the preference model, and for encouraging me during many early morning bicycle rides through the Arastradero loop. I thank Eric Horvitz for encouraging my interest in reasoning under scarce computation resources, and for numerous interesting discussions on time-dependent utility. I am grateful to David Heckerman for helpful advice, and for defining a standard of excellence for research in medical informatics. Adam Galper graciously provided his routines for inference in belief networks.

This dissertation builds on the research that was carried out by the many contributors to the VentPlan project, including Jeanette Polaschek, Stig Andersen, Samson Tu, Richard Peverini, and Michael Kahn. Ingo Beinlich modified ALARM to create an early version of VentPlan's belief network. This research also benefited from discussions with Adam Seiver, Larry Widman, Tze Lai, Tom Raffin, Greg Cooper, Mike Higgins, Peter Solovitz, John Egar, Harold Lehman, Marty Chavez, Homer Chin, Jaap Suermondt, Yuval Shahar, Paul Dagum, Les Lenert, Malcolm Pradhan, Max Henrion, and Greg Provan. I thank Reed Gardner, Tom East, and Alan Morris for helpful discussions, and for their hospitality during my visit to LDS Hospital.

I thank Lyn Dupré for her review of this manuscript, and for her efforts to remove the bugs in my writing. I only wish that her book "Bugs in Writing" were available when I began my writing! I thank Darlene Vian for her friendship and for superb administrative guidance.

This research was supported by grant IRI-9108359 from the National Science Foundation, and by grants LM-07033, LM-04316, and LM-04136 from the National Library of Medicine. Additional support was provided by grant NSF IRI-9311950, and by a gift from Hewlett-Packard Corporation. Computing facilities were provided by the SUMEX-AIM resource, grant LM-05208, and by the CAMIS Resource, LM-05305.

I also owe a debt of gratitude to my parents (which I am unlikely ever to repay), and to my daughter Brooke, for reminding me that I was once a kid.

Finally, I reserve my deepest gratitude for my partner, Maria Tovar. Maria has guided me through many experiences in medical informatics, and in life. She provided me with physical, emotional, and intellectual support, without which I would not have completed this dissertation.

Table of Contents

List of Tables

List of Illustrations

Chapter 1

Introduction

In this dissertation, I develop automated methods to reason about complex systems under time constraints. I bring together concepts and techniques from the disciplines of decision analysis, statistics, artificial intelligence (AI), and simulation to develop an approach to high-stakes, model-based decision making under scarce computation resources. My hypothesis is that an algorithm for the dynamic selection of models under time constraints—a *metalevel reasoning strategy*—can select, from a set of models that vary in their complexity and accuracy, a model that is accurate and that is computable within a time constraint.

I demonstrate that this hypothesis is valid by developing the *dynamic-selection-of-models algorithm* to automate model selection for model-based controllers; I implement this algorithm in Konan, a program that makes time-critical decisions for the settings of a mechanical ventilator for patients in the intensive-care unit (ICU).

1.1 The Ventilator-Management Problem

Critically ill patients who are in the ICU frequently require artificial breathing support from a mechanical ventilator. Ventilators have multiple control settings that allow a clinician to adjust the rate of ventilation, the volume and pressure of gas delivered during each cycle, the positive end-expiratory pressure (*PEEP*)[1]and the fraction of oxygen (FIO_2) in the inspired air.[2] As a patient's condition changes, the clinician must adjust the settings of the ventilator to maintain the benefit of adequate ventilation and to reduce the risk of injury or toxicity from the ventilator treatment.[3] To decide how to adjust the ventilator, a clinician examines the patient, observes the ventilator settings and the resulting airflows and pressures of ventilation, and considers the results of radiographic studies and blood tests.

The clinician must reassess the ventilator settings with a frequency that depends on how stable the patient's condition is; typically, she reassesses at least every few hours. Whenever a sudden deterioration in a patient's physiologic state occurs, however, the clinician must reevaluate the ventilator settings within 1 to 2 minutes, or risk a further decline of the patient's state.

I use the term **ventilator-management advisor** (VMA) to refer to an automated model-based controller that monitors patients in the ICU and continuously reassesses the settings of a ventilator. A distinguishing feature of a VMA is that a VMA incorporates a prediction model to reason about the expected effects of alternative settings. *VentPlan* is a prototype VMA that incorporates a single, simplified model of patient physiology to predict the effects of the ventilator on a patient's physiologic state.

1.2 VentPlan

VentPlan monitors patients in the ICU who are receiving treatment with a ventilator, and recommends appropriate settings for the controls of the ventilator. VentPlan incorporates a mathematical model of physiology in a model-based control architecture, in which a quantitative model predicts the effects of proposed changes in the control settings of the

1. PEEP is the ventilator setting for the pressure in the mouth during expiration.
2. In addition, many ventilators have an adjustable pressure waveform or pattern of airflow during each ventilatory cycle [Kirby, 1984].
3. Side effects of ventilator therapy include collapse of a lung, chronic lung damage, and oxygen toxicity.

ventilator. A decision-theoretic preference model interprets the predicted effects of changes in the ventilator settings. The preference model ranks alternative settings, and VentPlan recommends those control setting that the preference model predicts will be optimal. VentPlan demonstrates that, for a restricted range of patients, a simplified quantitative prediction model of the ventilator–patient interaction allows VentPlan to make appropriate recommendations for the settings of a ventilator. A preliminary evaluation of VentPlan's control-setting recommendations showed that they were in good agreement with the settings actually implemented by the ICU clinicians, for a set of patients whose physiologic problems were well described by the VentPlan model (see Chapter 3 and [Thomsen, 1989; Farr, 1989; Rutledge, 1993a]).

Although VentPlan's physiologic model predicts accurately the response of patients with a restricted range of physiologic abnormalities,[4] we would not expect this simplified model to provide accurate predictions for other patients. For patients with more complex abnormalities, an improved VMA would need a model that includes a representation of a wider variety of physiologic abnormalities.

I expanded the VentPlan model in three areas to create *VentSim*, a model that represents a wider variety of physiologic abnormalities than does VentPlan (see Chapter 4 for a description of VentSim). The computation complexity of the VentSim model is, unfortunately, much greater than that of VentPlan's model. If we implemented VentSim in place of the VentPlan model, the resulting VMA would generate treatment recommendations after a computation delay of over 1 hour.[5] Even longer computation delays would occur if we implemented models that are more detailed than is VentSim.

I reasoned that, by applying various simplifying assumptions to the VentSim model, I could create a set of models that varied in their areas of simplification, and that were intermediate in complexity between VentSim and VentPlan. An improved VMA would select a model, for each patient, that balances the need for model detail to explain complex abnormalities,

4. The clinical problems for which VentPlan's model applies include conditions that involve abnormalities in the shunt fraction, the rate of oxygen consumption or of carbon-dioxide formation, the cardiac output, and the oxygen-carrying capacity of the blood.
5. For the purpose of comparison with VentPlan, this delay is based on an implementation of a VMA on the same hardware on which VentPlan was developed—a Macintosh IIci.

and the need for model simplicity (reduced computation complexity) to compute the predictions within the time constraint for decision making.

1.3 VMA Scenario: Postoperative ICU Patient

Consider the following decision-making scenario that illustrates how a VMA would reason about the selection of appropriate physiologic models of a patient. A previously well patient is transferred to the ICU after an operation to remove a cancerous prostate gland. In the ICU, a VMA monitors the patient's condition, and makes recommendations for the ventilator treatment. In Figure 1.1, I show a possible interface to such a VMA.

Several hours after the patient arrives in the ICU, the pulse oximeter records a dangerously low level of arterial-oxygen saturation (63%). The sensor is attached correctly, and a recent measurement of the cardiac output (a thermodilution cardiac output from a Swan–Ganz catheter in the pulmonary artery) reports a low-normal cardiac output (4.5 liters per minute).

The VMA notes the change in oxygenation status, and recognizes that the situation is urgent. There is no time to consider in detail the possible abnormalities that might be the cause of the low oxygen level. The VMA applies a simple, three-compartment model[6] that allows it to state that the patient has changed, and allows it to make, within a few seconds, a recommendation to increase the level of inspired oxygen. The patient's nurse notes the alert from the VMA, requests an arterial blood gas (abg) analysis,[7] and increases the FIO_2 setting on the ventilator (see Figure 1.2). He calls the surgeon to inform her of the change in the patient's condition. She agrees with the action taken to increase the inspired oxygen concentration, and asks him to repeat the abg in 20 minutes to verify that the patient is improved. By the time the abg results are available, the pulse oximeter shows an increase in the oxygen saturation. The nurse obtains a second abg measurement to confirm the patient's improved oxygenation status.

The VMA recognizes that the dangerous situation has improved, and the time urgency for further adjustments of the ventilator is lessened. It turns its attention to explaining the

6. I refer here to the three-compartment model of Riley and Cournand, as implemented in VentPlan.
7. An abg analysis measures the pH, and the partial pressures of oxygen (P_aO_2) and carbon dioxide (P_aCO_2) in arterial blood.

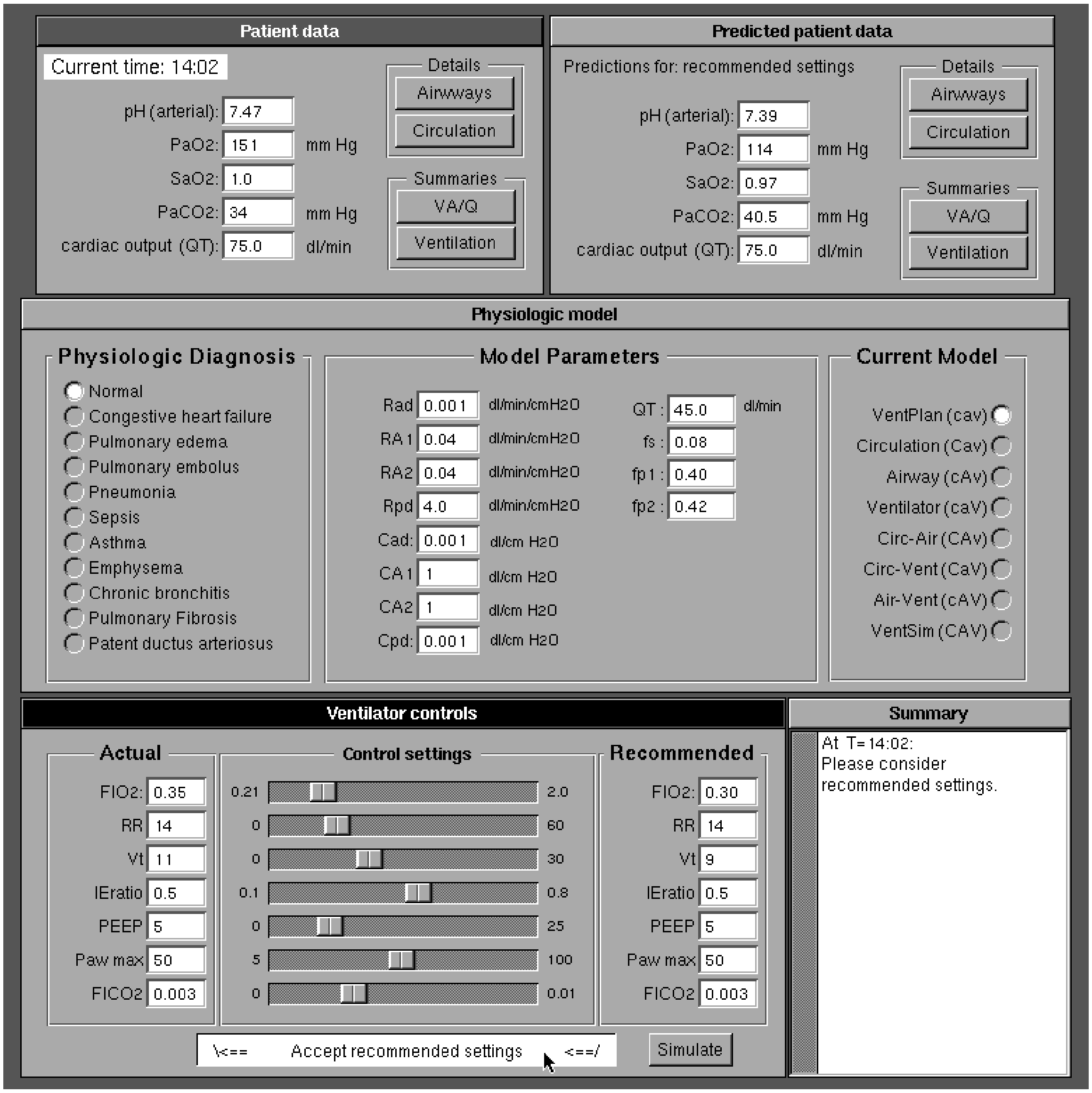

Figure 1.1 VMA scenario. The interface to the VMA shows current patient data in the upper-left panel, and the ventilator settings in the lower-left panel. The center panel shows the current model that the VMA is using, and the estimates of the model parameters, and includes an input for the clinician to assert a probable physiologic diagnosis. A text window in the lower-right corner shows the VMA interpretation of the data. The window in the upper-right corner shows the predicted effects of the recommended ventilator settings, based on the current model. The user has clicked on the button labeled "Accept recommended plan," which causes the VMA to transfer the recommended settings to the ventilator. The button labeled "Simulate" in the lower-left panel allows the user to simulate the effect of alternative settings by adjusting the ventilator controls and observing the effects in the "Predicted patient data" panel.

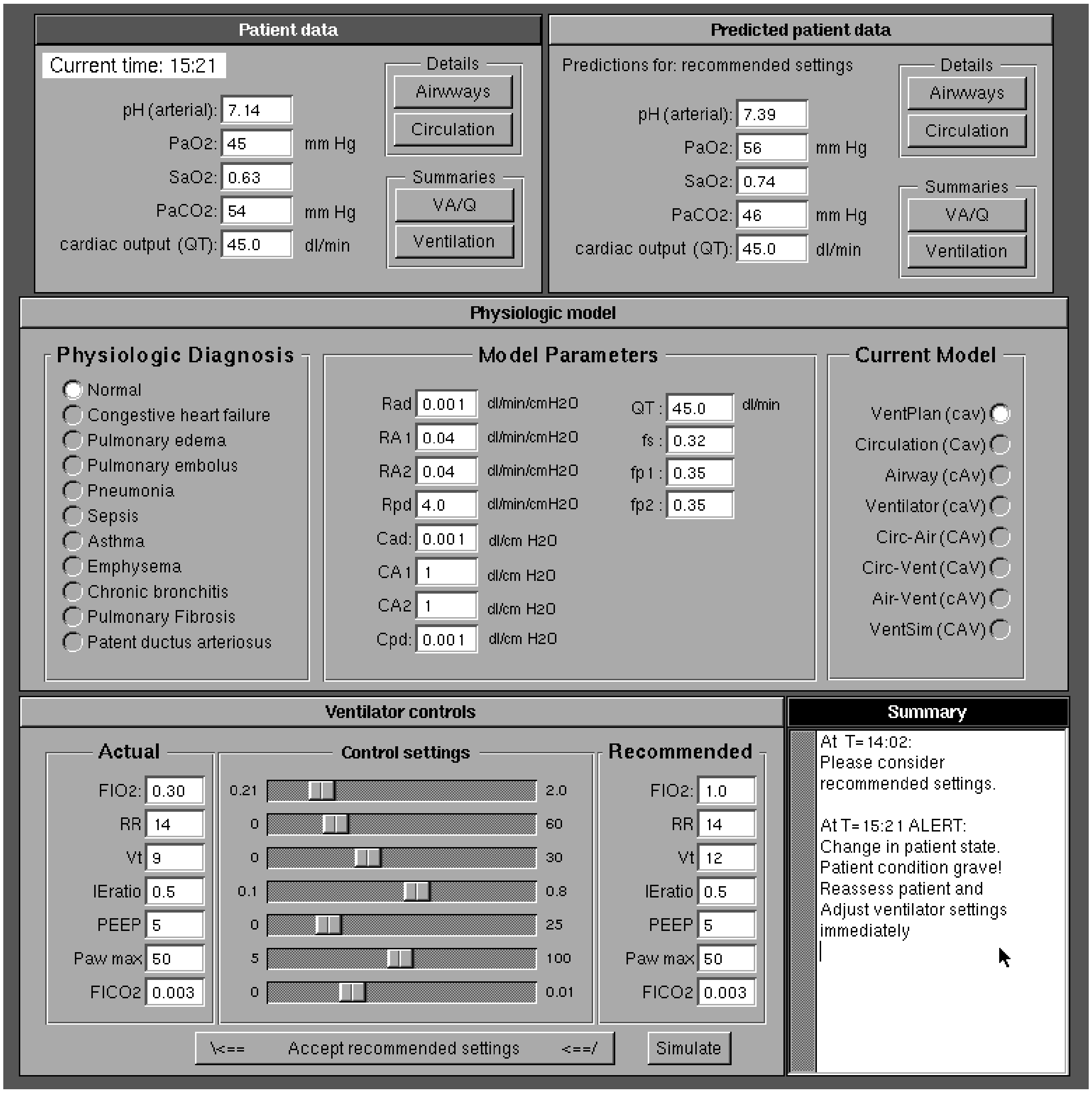

Figure 1.2 VMA scenario after a sudden deterioration in oxygenation status. The VMA notes a sudden fall in the level of oxygenation (the Patient-data panel shows P_aO_2 of 45 mm Hg), and reassesses the model. Because of the urgency of the situation, the VMA selects a simplified model (the *VentPlan (cav)* model, as shown in the far-right section of middle panel), and interprets the change in physiologic state as an increase in the shunt fraction. The VMA recommends an immediate increase in the FIO_2 to 1.0, and issues an alert, which appears in the lower right-hand window.

measurements of the partial pressure of oxygen (P_aO_2) from the abg's. The VMA notes that the simple model predicted a small rise in the P_aO_2, whereas a much larger rise actually occurred. The VMA considers alternative, more complex models to explain the observations of the P_aO_2. The VMA notes that a model with a more detailed representation of the ventilation–perfusion ($\dot{V}_A / \dot{Q}$) distribution[8] would explain the observations exactly. The VMA selects that model and predicts the effects of alternative levels of FIO_2. Based on the predictions for this patient, it recommends a lower level for the FIO_2 (0.60), which reduces the potential toxicity of oxygen and maintains the patient's oxygenation status at an acceptable level (see Figure 1.3). Although the VMA takes over 10 minutes to estimate the parameters of its model and to search for the recommended ventilator setting, the result becomes available while the surgeon is reexamining her patient.

The VMA generates a notice that, according to its analysis, the patient has an abnormal $\dot{V}_A / \dot{Q}$ distribution (see lower-right panel in Figure 1.3). The surgeon uses this information to confirm her clinical diagnosis that the patient has suffered a pulmonary embolus, and starts the patient on appropriate treatment for pulmonary embolus. She is concerned that the recommendation for the FIO_2 may be too low, however, so she asks the VMA to simulate alternative FIO_2 settings. She agrees that the predictions of the patient simulation are reasonable, but chooses a somewhat higher FIO_2 than the VMA recommends. She makes a note to herself to ask the ICU director when *her* preference model for setting the ventilator will be incorporated in the VMA.

1.4 Dynamic Selection of Models for a VMA

The scenario in Section 1.3 demonstrates some of the tradeoffs with which a successful VMA would have to reason to make appropriate modeling decisions. The less complex physiologic models allow the VMA to interpret new observations rapidly, and allow the VMA to explain the observations with reference to a simplified structural model. When the patient's state is suddenly critical, the VMA may have no choice but to make an immediate recommendation based on the predictions of a simplified model.

8. The $\dot{V}_A / \dot{Q}$ distribution is a description of the different proportions of ventilation (flow of air into each region of lung) and perfusion (flow of blood through each region) that occur in different regions of the lung.

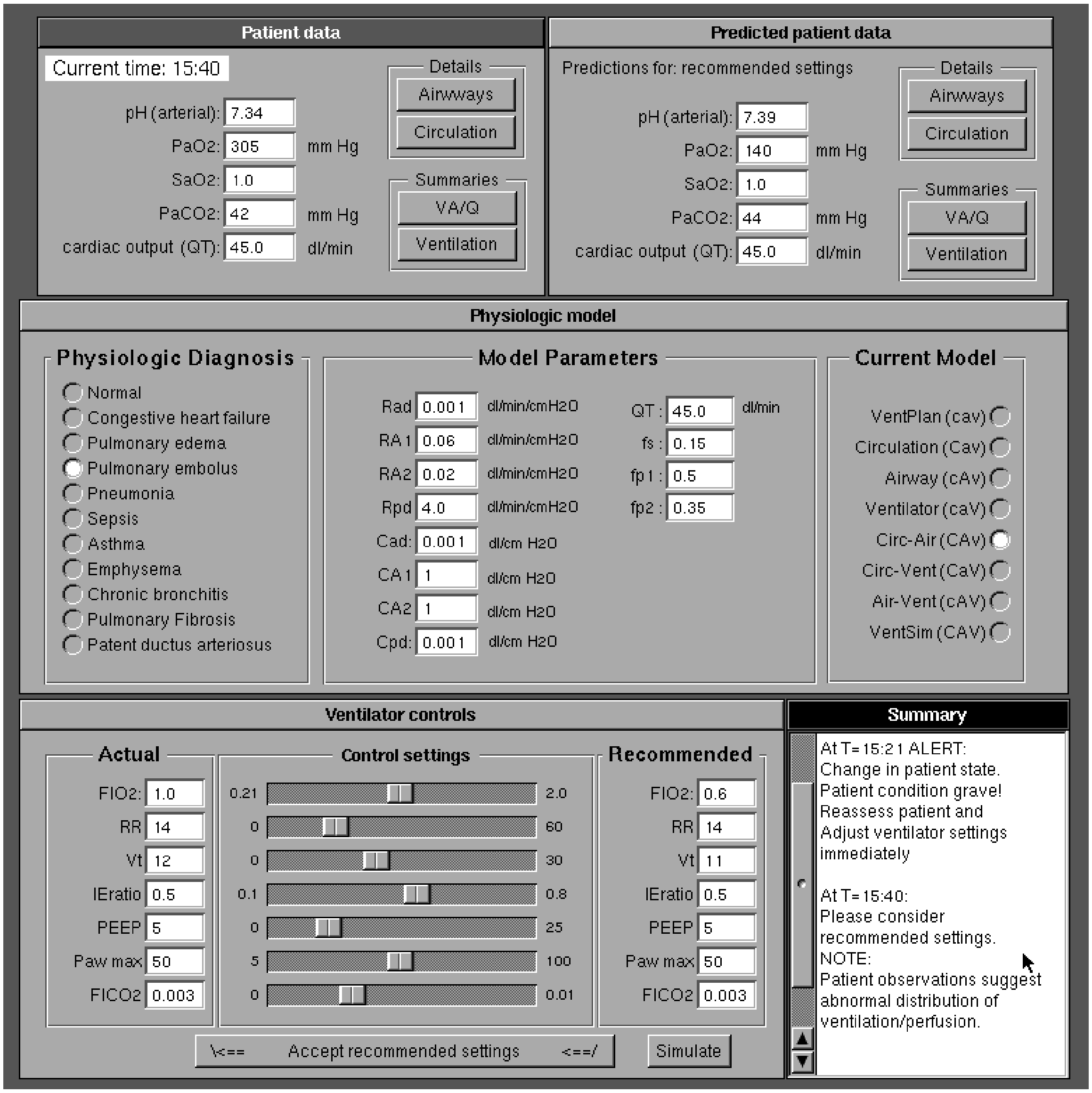

Figure 1.3 VMA scenario after an increase in FIO_2. The patient responded to an increase in the FIO_2 to 1.0, and the P_aO_2 is now 305 mm Hg. The time-criticality of immediate action is now reduced, and the VMA considers other, more complex models. The VMA determines that the CAv model, which has expanded circulation and airways components, is the least complex model that explains the observations. The VMA fits this model to the data, and applies the model to generate new recommended ventilator settings. The VMA also notes that the model predicts a markedly abnormal distribution of ventilation to perfusion in this patient (this analysis of the model parameters is available to the user through the Summaries: V_A/Q button in the upper left panel). The VMA issues a notice to the clinician, as seen in the lower-right panel.

As the patient's condition stabilizes, the time-criticality of further treatment recommendation is reduced, and the VMA can reconsider which model to select. If the simple model continues to explain the observations, then the VMA retains the simple model. However, if the simple model is inaccurate, the VMA searches for a minimally complex model that explains the observations. This minimally complex model should be a model that is detailed only in the areas corresponding to the physiologic abnormality that led to the observations that were not explained by the simple model. For example, in the scenario presented, the VMA prefers the least complex model that includes sufficient detail to represent an abnormal $\dot{V}_{\mathrm{A}} / \dot{Q}$ distribution.

The VMA scenario highlights the need for an automated method to find mathematical models that are at the right level of complexity for a time-critical task.

1.5 Mathematical Models

Modeling is at the heart of science and engineering. A review of the value of additional research to develop modeling methodologies noted that "mathematical models codify facts and help to confirm or reject hypotheses about complex systems. They reveal contradictions or incompleteness of data and hypotheses. They can often allow prediction of system performance under untested or presently untestable conditions. They may predict and supply the values of experimentally inaccessible variables" [NIH, 1989].

Scientists acquire insight by representing knowledge of physical and biological phenomena as a model. For example, James Clerk Maxwell developed a theory of electromagnetic phenomena that provides insight into the interactions between electric and magnetic fields. Maxwell described his electromagnetic theory in a series of equations that form a compact and powerful model of electromagnetic phenomena [Purcell, 1963]. Engineers develop models during the process of design—they use models to simulate the effects of alternative design options, and to test the performance of a design. For example, an engineer designing a bridge might first apply mathematical models of stress and mechanical performance to predict the strength of a specific design. He might then build a scale model (a physical replica) to verify the strength of the structure.

Studying the behavior of models of a system may be more useful than is studying the system itself. For example, the Navy teaches the diagnosis and maintenance of boiler plants by allowing trainees to interact with a numerical simulation of such plants [Hollan, 1984], and professors of anesthesiology teach the management of intraoperative crises by allowing anesthesiology residents to interact with a detailed mathematical and physical simulation of a patient in the operating room [Gaba, 1988].

1.6 Automated Model Selection

The problem of automating the selection of a model for a specific task has attracted considerable attention in the fields of AI and statistics (see Chapter 2 for a more detailed discussion of prior research in automated modeling).

Users of models of complex systems may avoid conceptual confusion if the model includes only the details needed to achieve acceptable accuracy. For example, a model used by physicians to assist them in understanding blood electrolyte abnormalities should be only sufficiently complex to include the causal pathways that explain the abnormality [Kuipers, 1984]. In other cases, a more detailed model is required to achieve greater accuracy. To predict the airflow over an airfoil at transonic speeds, for example, aerodynamicists require a highly detailed, computationally complex, finite-element model.

Greater accuracy (and complexity) may be essential in certain model-based applications; in others, accuracy may be sacrificed to increase simplicity, or to decrease the time taken to evaluate the model. The most important task of the model builder is to develop models that fulfil the complexity and tractability requirements of an application [Neelankavil, 1987]. Traditionally, model-building experts have hand crafted simulation models of complex domains to meet the required accuracy of a specified task with a minimum of complexity. The general problem of finding the appropriate assumptions and simplifications that will lead to a tractable yet accurate model has traditionally been a task for human experts, because developing such models requires knowledge of statistical and numerical methods, experience in model building, and expertise in the domain of the application.

Over the past decade, AI researchers have developed methods to automate the modeling of complex systems. These methods either select models from a set of candidates, or compose models from submodel components. In each case, these automated methods choose models based on qualitative information about the system, and on expert knowledge of model behavior. The AI methods rely on rule-based descriptions of expected model behavior to select models that match known features of the system to be modeled. We shall review these AI methods in greater detail in Section 2.6.

Researchers in statistics also have developed methods that select models based not on prior information, but rather on quantitative observations of a system. The statistical model-selection methods rely on quantitative assessments of how well alternative models are able to explain all observations. We shall review the statistical methods in greater detail in Section 2.7.

The methods of AI and statistical model builders are complementary. The AI methods assume that the only information needed to construct a model is prior information; then, they perform limited validation of their models against observations of the system. Statistical model builders assume that the information needed to build a model is available in the set of observations alone, and have few methods for incorporating prior information in the model-building process. Naturally, AI researchers apply their methods to domains where prior knowledge of the system is available—such as mechanics models of physical systems [Forbus, 1984; Falkenhainer, 1991; Gruber, 1993]. Similarly, statistical researchers apply statistical methods to build models from large observational data sets [Box, 1985].

For some problems, both prior knowledge of the domain and observations of the domain constrain the structure of the model. For these problems, combined methods that take advantage of both AI and statistical methods are needed. For example, models of human physiology must be consistent with a large body of knowledge about physiologic structure and behavior, and must explain specific quantitative observations. The problem of identifying a model that predicts accurately the physiologic behavior of a specific individual is particularly difficult when the observations are sparce and the choice of appropriate model must be based largely on prior information and qualitative observations.

A model-selection method for an automated ICU-monitoring program should select a model that explains a set of quantitative observations of a patient in the ICU, but should also take into consideration the qualitative information that is available. For example, qualitative information about a critically ill patient in the ICU might include that she is receiving treatment with a ventilator, has the clinical appearance of a patient with pneumonia, and has had a chest X-ray examination that confirms the presence of pneumonia. In addition, quantitative observations might include measurements of arterial partial pressure of oxygen and carbon dioxide, cardiac output, blood pressure, pulse rate, and mean airway pressure, all of which were obtained both before and after a change in the control settings of the mechanical ventilator.

Whenever model selection must take into account both prior information (such as “the patient has severe pneumonia”), and available quantitative observations, (such as “on the current ventilator settings, the partial pressure of oxygen is 78 mm Hg”), a combined AI and statistical method for model selection is needed. In Chapter 5, I shall develop a metric of model benefit that is based on a combination of AI and statistical approaches. This metric of model benefit will allow an automated model-selection algorithm to select models that are consistent both with prior information and with the quantitative observations of a system.

The Time Constraint for Model-Based Recommendations in the ICU

The large number of model evaluations that VentPlan performs during each cycle of data analysis[9] limits the computation complexity of any model that an improved version of VentPlan might implement. VentPlan’s simplified model leads to computation delays of 1 to 2 minutes, whereas highly detailed models of physiology could increase that delay to from 10 minutes to many hours.[10]

VentPlan shows that, for some patients, a simplified model, which leads to recommended actions within a short time, is ideal. For other patients, whose physiologic abnormalities are not well characterized by the VentPlan model, we do not expect VentPlan to make the best possible recommendations for the ventilator settings. For such patients, we would rather

9. VentPlan evaluates the model repeatedly during parameter estimation (this procedure finds patient-specific parameter values that cause the model to predict, as closely as possible, the patient data), then does so again during the control-setting optimization (this optimization searches the space of possible ventilator settings to find the settings for which the model predictions optimize the preference model). The total number of model evaluations is greater than 1000 during a single cycle of data interpretation and treatment recommendation.

10. The time delay induced by computationally complex models depends on many factors, which I discuss in greater detail in Section 6.2.

incur some additional computation delay, if the result were accurate patient predictions within time to be useful. In other words, we prefer models that lead to accurate patient predictions, but we also prefer models that cause only short computation delays. How should we determine which model is optimal for a given set of patient data?

1.7 The Optimal Model Under a Time Constraint

In a time-critical control application, we may prefer models that compute, in a short time, suboptimal control actions over models that compute, after a long delay, the action that would have been optimal had the delayed action been available at the time computation began. For a model-based controller that must compute control settings within a time constraint, the *optimal model* is a model that maximizes the tradeoff of model benefit (a measure of how accurately the model predicts the effects of alternative control settings) and model cost (due to the length of the computation delay).

We also shall develop, in Chapter 5, two criteria to identify models that are likely to be optimal under a time constraint. These criteria will assist an automated method for the dynamic selection of models. The first criterion, called the *prior dynamic-selection-of-models metric* ($\text{DSM}^{\text{prior}}$), provides an easy to compute measure of the probability that a model will make accurate predictions, given only the prior information about the system, and given knowledge of the level of model detail and of computation complexity. This measure identifies models that are unlikely to provide accurate predictions for a specific patient, or that would lead to excessive computation delays. The second criterion, called the *dynamic-selection-of-models metric* (DSM metric), updates the $\text{DSM}^{\text{prior}}$ with a statistical measure of the quality of the fit of the model to the quantitative observations, to determine the final model selection.

1.8 The Dynamic-Selection-of-Models Algorithm

Finding the optimal model under a time constraint is a difficult problem because, under that time constraint, there is no time to compare more accurate and less accurate models. In Chapter 5, I shall describe a heuristic algorithm, called the *dynamic-selection-of-models algorithm*, that selects, within a time constraint and from a set of alternative models, a model that optimizes the heuristic measures of the tradeoff of model accuracy and model computation complexity. This algorithm provides a method to allow a real-time VMA to perform model selections that will lead to model-based treatment recommendations within the time constraint of ICU decision making.

The dynamic-selection-of-models algorithm requires that the alternative models be organized as nodes in a graph, in which the edges are labeled with the simplifying assumptions that distinguish adjacent models (see Figure 2.1). To demonstrate the application of the algorithm to the time-constrained model selection for a VMA, we must have a graph of alternative models of cardiopulmonary physiology. In Chapter 6, I describe such a graph of models (GoM), in which the models vary in their level of detail. Each model makes simplifying assumptions that define the level of detail that the model must incorporate in each of its components. The least complex model in this set, the VentPlan model, has simplified versions of all three of VentSim's components. The most detailed model in this GoM, the VentSim model, includes the highest level of detail—among all the models in the GoM—in all of its components. VentSim is too computationally complex to be useful to a real-time VMA, but is a reference model against which we can compare the performance of other, less detailed and less computationally complex models.

Finally, we shall investigate the performance of the dynamic-selection-of-models algorithm by applying the algorithm to select models of cardiopulmonary physiology for a VMA, using data sets that correspond to simulated patients with various physiologic abnormalities (see Section 6.5). The results demonstrate that the dynamic-selection-of-models algorithm makes appropriate model selections in the high-stakes, time-critical decision-making environment of the ICU.

1.9 Guide for the Reader

In Chapter 2, I present a review of prior work on model selection and automated modeling. I begin with a review of recent work in AI to automate the model-building process, and compare that with the statistical view of model selection from data. This material is not essential to an understanding of the method for dynamic selection of models, and may be skipped without loss of continuity.

In Chapter 3, I present a discussion of the use of mathematical models in automated ICU applications, and then I present the results of the VentPlan project, which provided a starting point for this research. VentPlan is a prototype VMA that implements an architecture for combining differing models of physiology in a model-based control application.

In Chapter 4, I describe VentSim, an expanded model of cardiopulmonary physiology, and its implementation with a graphical user interface.

In Chapter 5, I develop a theory of model selection under computation-time constraints. I define optimal model selection under a time constraint, and present an heuristic approach to model selection under time constraints. I describe the dynamic-selection-of-models algorithm for performing model selections under time constraints.

In Chapter 6, I implement the DSM algorithm to select models for a VMA. First, I describe a set of physiologic models for use with a VMA, and show an organization of the models as a graph of models. I demonstrate the behavior of the model-selection algorithm on this graph of physiologic models.

In Chapter 7, I review the work presented in this dissertation, and then discuss my perspective on the significance and the contributions of this work. Finally, I point to future research directions.

Chapter 2

Automated Modeling

In this chapter, we review the terminology that we shall use to describe models, then we review prior work in AI on automated modeling. We then discuss prior work on the development of statistical model-selection criteria that researchers in statistics developed to evaluate how well alternative models explain sets of observations. We shall see that the differences among these criteria are equivalent to different assumptions about the prior probabilities of alternative models. In the last section, we consider a problem—modeling the cardiopulmonary status of patients in the ICU—that requires a combination of methods from AI and statistics to solve.

In this chapter, we discuss model selection as if there were no cost to the use of any model. In Chapter 5, we shall extend this discussion to consider the effect that a computation-time delay has on the benefit of a model in time-critical situations.

2.1 Terminology

Models are abstractions of *systems*. A **system** is some real or imagined part of the universe that we consider for the purpose of study and analysis. The largest and most complete system is the universe. We can consider any part of that universe as a system, and we can subdivide any system—no matter how small—into smaller components, each of which also constitutes a system. For example, an automobile is a system, an automobile engine is a system, and each cylinder of an automobile engine is a system. The smallest subatomic particle is a system that we can subdivide into smaller systems by proposing the existence of even smaller, sub-subatomic particles.

The defining feature of systems is that they are sources of data—systems allow observations that generate data. In other words, we can define a system as anything that allows observations of system variables. **Variables** correspond to features of a system: "we should pick out and study the facts that are relevant to some main interest that is already given.... The system now means, not a thing, but a list of variables." [Ashby, 1963, p. 54]

A **model** is an abstraction of a system; a model also generates data. If a model perfectly reproduces the data that are observed from a system, we say that, under the experimental conditions investigated, the model is accurate. The set of conditions under which we study the system is called the *experimental frame* [Zeigler, 1976].

An important principle of modeling is that each model has an associated experimental frame within which the model is useful. For example, consider a model of spacecraft motion for interplanetary travel. A model that is based on Newtonian principles has an associated experimental frame that restricts the speed of the spacecraft to a small fraction of the speed of light, c. For predictions of the position of a spacecraft that travels at a relative velocity greater than 0.1 c, we should not apply a Newtonian-motion model, since velocities greater than a small fraction of c are outside the model's experimental frame, and the model would make inaccurate predictions. A kinematic model that includes relativistic effects has an experimental frame that includes spacecraft velocities that approach c [Sears, 1964].

We cannot expect a model to be accurate outside the experimental frame for which it was designed. The description of the experimental frame for a model is as important to specify as is the structure of the model itself [Zeigler, 1984].

Types of Models

There are at least three types of models [Cellier, 1991]:

- *Conceptual models*. These models are cognitive constructs—they are not instantiated in a physical form or enumerated in a mathematical form. They are also called mental models.
- *Physical models*. These models are physical abstractions of a system. Examples include a balsa-wood miniature scale version of an airplane, a full-scale plywood mock-up of the U.S. space shuttle, a ball-and-stick representation of a molecule, and a set of branching glass tubes that match the shape and size of the human tracheobronchial tree.
- *Mathematical models*. As the name implies, these models are expressed as mathematical relationships. For example, Maxwell's equations form a mathematical model of electromagnetic fields [Purcell, 1963].

The term *mathematical model* includes discrete-event and continuous-event models, in addition to continuous-state and discrete-state models. That is, the state variables and the independent variable (typically, time) may be represented either discretely or continuously. In this dissertation, I emphasize continuous-state, continuous-time models, as these models offer the most powerful method to represent the behavior of physical and biological systems. Nevertheless, the issues of model simplification, model complexity, and automated design apply to all types of models.

2.2 Conceptual Models Versus Mathematical Models

We are unable to create mathematical models of systems that include all know interactions of all components, because there are innumerable subdivisions of every system. For example, a complete conceptual model of the human body would have to represent the physical, chemical, electromagnetic, and quantum interactions of all atoms, molecules, tissues, organs, and organ systems in the body. We cannot construct such a model, but we can generate a conceptual model by thinking about the complete conceptual model.

A mathematical model of a complex system is a description of certain aspects of a more complete conceptual model. Any realizable mathematical model must simplify a conceptual model to retain only those components and interactions that have meaningful effects on the model's predictions. Whether or not parts of a conceptual model will have meaningful effects on the predictions of a model depends on the purpose of the model. For example, consider the full conceptual model of an automobile. This model includes all functional components of the vehicle, such as the engine, drivetrain, suspension, and body. Each component has subcomponents that are also modeled at a level of detail that includes all known interactions of each part and subpart.

The conceptual model is arbitrarily complex, since it contains all interactions that we understand to occur from the macroscopic level down to the quantum level. Insignificant interactions, or inappropriately detailed interactions, are part of the conceptual model. We may choose to think about the conditions under which they have a noteworthy effect, but we would not include such interactions in any model that we implemented.

Now consider just the engine component of the conceptual model of an automobile. This component includes interactions that lead to no significant effects—for example, the interaction of the earth's magnetic field with ferrous engine parts. Other interactions are present at an atomic level of detail, but are more conveniently described at the macroscopic level. For example, the effects of temperature on path motions of each iron atom within the engine block are present in the conceptual model, but the temperature-dependent properties of the iron alloy in the engine block are described more usefully at a macroscopic level.

The most complex mathematical model of a system that can be implemented is called the **base model** of that system [Zeigler, 1976]. We apply simplifying assumptions to the base model to remove unnecessary complexity from the conceptual model. For example, we assume that the automobile is to be used in a terrestrial environment that does not include intense magnetic fields, so that we can safely ignore the interactions of metal parts with the ambient magnetic field.

A model builder chooses simplifying assumptions that reduce the complexity of the full conceptual model to create a less complex base model. For most applications, the base model is still unnecessarily complex, and the model builder will choose additional assumptions that simplify the base model to create less complex models that are suitable for specific tasks. Zeigler refers to these simplified models as *lumped models,* because these models often lump together model variables and parameters in parts of the model that are less relevant to the task at hand [Zeigler, 1976]. For example, in a model of an automobile that simulates the effects of motor-vehicle collisions, the detailed internal interactions of the engine might be lumped into a single variable that describes the engine mass.

2.3 Mathematical Models

Mathematical models represent the state variables and a time variable as either discrete or continuous quantities. Qualitative mathematical models use a discrete-time, discrete-state (or finite-state) representation [Forbus, 1984; Kuipers, 1986]. Models of periodic systems often implement a discrete-time variable, so that the periodic behavior is represented once per cycle. For example, a biological model of population growth of a species would naturally select a discrete time interval equal to the length of the reproductive cycle. Such a model would predict the population at the beginning of each cycle as a function of the state variables of the previpreviousous cycle [Stewart, 1988].

This dissertation addresses the problem of building and selecting mathematical models that include continuous-time, continuous-state representations. These *dynamic state-space models* are the most complete, and potentially the most accurate models for the simulation of complex systems [Cellier, 1991; Press, 1988].

Models that apply arbitrary mathematical descriptions to sets of data, and that are not based on the structure of the system, are called *empiric* models. For example, Draper examined several models of the relationship of temperature to the probability of primary O-ring erosion for the space shuttle. The only knowledge of the relationship of probability of O-ring failure to temperature was that the failure probability increased monotonically with decreasing temperature. The alternative models were varying combinations of logit, probit, and complementary log-log functions of temperature and the second power of temperature [Draper, 1993]. As Draper points out, the lack of knowledge of the mechanisms that were modeled led to substantial uncertainty about which prediction model would be most accurate.[1]

1. Draper suggests that the predictions should be integrated across all reasonable models to assess the true prediction variance due to uncertainty in model structure.

Models that represent abstractions of some or all of the structure of the system are called *structural* models. One advantage of structural models is that knowledge of mechanisms of system behavior allows parsimonious models to predict complex behavior [Cellier, 1991; Zeigler, 1976]. A more powerful reason to prefer structural over empiric models is that inferred values for model parameters that are unmeasured may provide insight into the mechanisms of a system.

For example, the multicompartment model of iron metabolism represents an abstraction based on knowledge of the metabolic pathways of iron absorption from the intestines, incorporation into hemoglobin, and elimination through the reticuloendothelial system [Franzone, 1982]. Inferred values for the model parameters that specify the rate of elimination of iron are a direct measure of the activity of the reticuloendothelial system. These inferred features of iron metabolism allow accurate classification of alternative types of disorders of hemoglobin metabolism [Barosi, 1985].

2.4 Space of Models

The **space** of possible models has multiple dimensions: scope, domain, granularity, and accuracy [Weld, 1992]. The **scope** of a system refers to the size of the list of observable variables. An automobile is a system with a scope that is larger than that of an internal combustion engine. We should not choose to develop a model that has a larger scope than does the system under consideration. The **domain** of a model is the set of variables from the system that the model includes. The terminologies of Weld and Zeigler are related, in that the scope plus the domain together define the experimental frame for a model. **Granularity**, or *resolution*, refers to the number of levels that the variables in the model can have. At one extreme, model variables can be continuous numbers; for example, the speed of a spacecraft is often modeled as a real number. At the other extreme, variables may be restricted to a small number of values; for example, qualitative models of fluid dynamics typically restrict the values for the rates of flow to be either negative, zero, or positive [Forbus, 1984]. In between these extremes, semiquantitative approaches offer an intermediate granularity [Widman, 1989].

2.5 Reasoning About Models

Reasoning about models is a difficult task, because the number of possible models is large. Greater accuracy (and complexity) may be essential in certain applications; in others, accuracy may be sacrificed to increase simplicity, or to decrease the time taken to evaluate the model. As Simon said [Simon, 1990], "Intelligent approximation, not brute-force computation, is still the key to effective modeling." Traditionally, model-building experts have hand-crafted simulation models of complex domains to meet the required accuracy of a specified task with a minimum of complexity. Expert model builders reduce the complexity of the modeling problem by applying assumptions that simplify the modeling task. The **simplifying assumptions** are constraints that reduce the size of the space of possible models. Weld points out that the technique of applying simplifying assumptions is analogous to the abstraction methods implemented in the fields of planning and search [Weld, 1992].

A model represents a hypothesis about the behavior of the corresponding system. If the hypothesis is true, then the model will predict accurately the system behavior within the model's experimental frame. For example, Meyer and other researchers constructed the FERRARI model of lunar gravitation to predict the effects of nonspherical lunar shape on stationary orbits of lunar satellites. If the FERRARI-model hypothesis were true, then FERRARI would predict the locations of lunar-orbiting satellites, within the accuracy of tracking radar observations, during orbits that last greater than 1 year [Meyer, 1994]. We can test the validity of the FERRARI-model hypothesis by applying statistical methods to observations of lunar satellites.

Researchers have developed both artificial intelligence (AI) and statistical approaches to the selection of models. Statisticians often require models to explain large sets of observations when little knowledge of an appropriate model structure is available. Lacking prior knowledge of the system structure, statistical model builders apply purely descriptive mathematical models to explain the observations. Researchers in the field of AI address the opposite problem: They apply only their knowledge of a domain to construct a model; few quantitative observations constrain the model. Researchers have not yet combined statistical and AI methods to solve model-building problems that are constrained both by quantitative observations and by domain knowledge.

2.6 The AI Perspective: Automated Modeling

AI researchers have described two approaches to reasoning about model building: *selection* of a model from a predetermined candidate set of models, and *composition* of a model by automated selection of the individual model components. Both AI methods for automated modeling perform model selection. These methods perform a search through the space of possible models to select a model that has no violated constraints. The constraints are based on prior domain knowledge, which the AI methods represent symbolically. The methods modify an initial model selection if subsequent observations show that the model makes inaccurate predictions. The process of moving from an initial model to a more accurate model is called **model refinement**.

2.6.1 Model Selection in the Graph of Models

A logical first step in developing a model-selection method is to organize the models in a manner that helps the model search. The **graph of models** (GoM) is a formalism for organizing models that represents graphically the assumptions of each model (see Figure 2.1). The GoM contains the set of all models, which must be enumerated in advance. The nodes of a GoM represent models, and the edges represent the changes in assumptions between adjacent models. The GoM was first implemented in PROMPT, a system that reasoned about structural design problems [Murthy, 1987]. In PROMPT, modeling assumptions were represented explicitly in the form of **consistency rules** and **parameter-change rules**.

2.6.1.1 Consistency Rules

The consistency rules represented the assumptions of the model; they were a set of constraints on the system state that had to be satisfied for a model to be accurate. For example, a model of fluid dynamics that made the laminar-flow assumption had a consistency rule stating that the Reynolds number of all flows under consideration had to be less than 2300.[2] Consistency rules were specified for each model in the GoM by the model builder; they were based on her modeling expertise and domain knowledge.

2. The flow of a fluid is laminar if the Reynolds number is greater than 2300; otherwise, the flow may become turbulent.

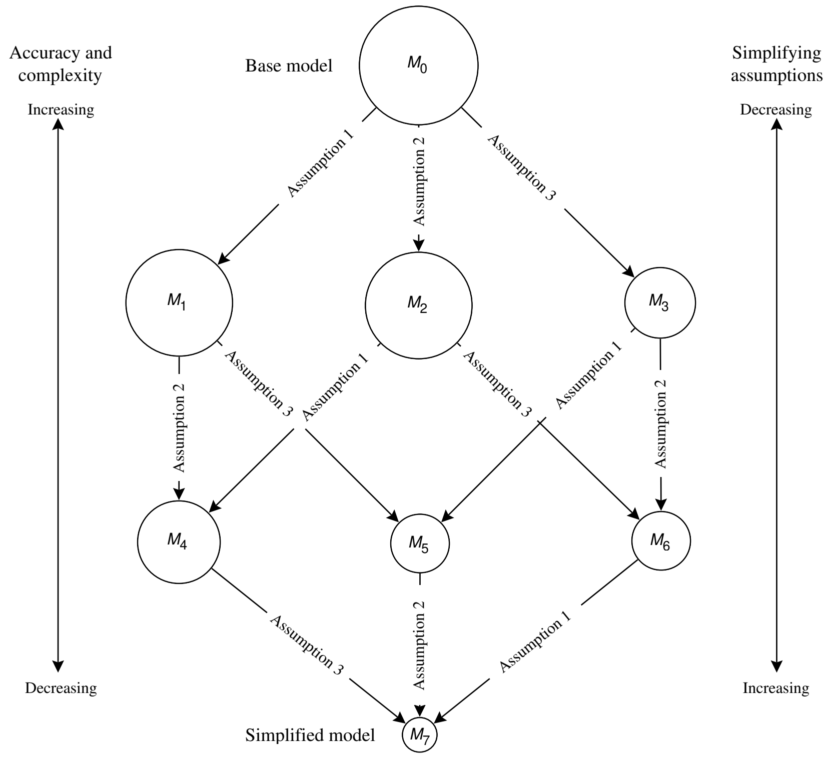

Figure 2.1 Graph of models. The base model (M_0) is the most complex, and is expected to be accurate in all experimental frames of interest. When simplifying assumptions are valid, one or more less complex lumped models (M_1 through M_7) also may be accurate. Arrows are labeled with the names of the simplifying assumptions that distinguish adjacent models. This GoM is a generalized version of the GoM for a ventilator-management advisor that is shown in Figure 6.3.

The GoM formulation enabled PROMPT to select models that had valid consistency rules (simplifying assumptions), and that were therefore internally consistent—that is, the structure of the model was consistent with the observations. If subsequent observations invalidated a modeling assumption, then the model was internally inconsistent, and the GoM guided a search for an alternative model. For example, if a flow was observed to have a Reynold's number of greater than 2300, then the consistency rule indicated that the laminar-flow assumption was invalid. The edge leading from the currently selected model that was labeled with the laminar-flow assumption led directly to a model that retracted the laminar-flow assumption.

2.6.1.2 Parameter-Change Rules

Parameter-change rules are a method to implement model refinement in the GoM. In Addanki's formulation, parameters included variables of a model that took numerical values.[3] Parameter-change rules represented the expected changes in model predictions that would occur if a switch were made to an adjacent model in the GoM. For example, a GoM for the design of an automobile transmission included two models that differed in their representation of the effects of friction. An edge connected the model that included no representation of friction to the model that included a characterization of coulomb friction; this edge was labeled with a parameter-change rule that concluded what would be the effects of switching from the no friction to the coulomb-friction model. The rule stated, "If two solid objects are in contact, then moving from the no friction model to the coulomb-friction model will lead to the prediction of an increased force parallel to the surfaces of contact."

An internally consistent model was *externally consistent* if there were no discrepancies[4] between observations and model predictions. If a discrepancy occurred, a model was externally inconsistent, and PROMPT attempted model refinement. The parameter-change rules suggested an alternative model, for which the rules indicated that the expected effect of moving to the alternative would reduce the discrepancy.

Unfortunately, there was no guarantee that the changes encoded in the rules would occur as predicted by the model builder. For complex models, the effects of switching models might be complex, and might be difficult to describe as a set of simple parameter-change rules. Nevertheless, parameter-change rules did perform model refinement for problems in various physical domains, including thermodynamics, hydrodynamics, and mechanics [Addanki, 1989].

---

3. Note that Addanki's use of the term *parameter* conflicts with the definition that I presented in Section 2.2.
4. A common approach is to define an arbitrary minimum discrepancy that must be exceeded before the program declares an external inconsistency. The problem of defining the minimal acceptable prediction accuracy for an application was not solved by Addanki.

2.6.2 Compositional Modeling

The GoM approach searched a set of fully enumerated models. The compositional-modeling approach overcame the requirement of the GoM to enumerate all models in advance, and allowed a search through a much larger space of possible models. Compositional modeling applied explicit modeling assumptions to decompose domain knowledge into partially independent model fragments, then composed a **scenario model** by collecting combinations of model fragments. Falkenhainer defined each model component, or elementary domain model, with a set of assumptions, operating conditions, and relationships (either qualitative or quantitative) that the fragment specified when it was applied. Each model fragment had an ontological assumption: the ontologies considered were contained-stuff, energy-flow, molecular-collection, and mechanics. Falkenhainer gave an example of a quantitative scenario model that his method composed to simulate the time-varying flow of oil among three connected tanks [Falkenhainer, 1991].

Compositional modeling assumes that we can build models from a set of elementary domain model fragments that are created at different granularities, with different ontologies and with different operating regions. More significantly, compositional modeling assumes that the interactions among model fragments are well defined and declarable in the form of explicit assumptions. For physical systems, this decomposition of component models is possible.

For biologic systems that have a high degree of interdependence among components, and that contain many homeostatic mechanisms, this decomposition may be more difficult. For example, a compositional-modeling approach to construct a scenario model for the cardiopulmonary system might reason about components for the circulatory system and for the lungs. These two components are tightly linked, however, as the circulatory system passes through the lungs, and a circulatory-model fragment must match the specific structural detail of the circulation of blood in the lung-model fragment. It would make no sense to compose a model by selecting both a lung component and a circulatory component, without a detailed consideration of how the two would work together. In effect, the granularity for the components of this scenario model should be the entire cardiopulmonary system.

2.6.3 Model-Sensitivity Analysis

Both the GoM approach (as implemented in PROMPT) and compositional modeling reasoned symbolically with sets of rules that defined circumstances under which models or model fragments were accurate to perform model selection. Another approach is to reason about alternative models based on only an analysis of model behavior. The task of analyzing the effects of switching models is called **model-sensitivity analysis** (MSA). MSA enables a principled resolution of conflicts between observations and model predictions. That is, MSA analyzes or evaluates the models under consideration directly, and does not rely on a representation of an expert's knowledge of expected model behavior [Weld, 1992].

MSA may be accomplished by any of a number of techniques, including symbolic reasoning about model structure, numerical analysis of model predictions, and a combination of symbolic and numeric techniques. The most direct and conceptually simple method to accomplish MSA is to compare symbolically the closed-form solutions of each model. Unfortunately, we are unable to find closed-form solutions for many models whose structure is defined by complex dynamic constraints. Whenever closed-form solutions are not available, numerical simulation will give results for MSA. The numerical solutions for MSA apply to only the initial model conditions that were solved, and the solutions may not generalize to other initial conditions. However, if the task for model refinement involves finding a model to use under a known set of conditions, the numeric solution to MSA may be acceptable.

In general, MSA is difficult, and there is no universal method to perform MSA without evaluating the models under all possible initial conditions [Weld, 1992].

Fitting Approximations

An intriguing method that makes MSA practical involves restricting the relationships among models, so that each pair of models under consideration has a **fitting-approximation** relationship. An approximate model is a fitting approximation of a detailed model if, as one detailed-model parameter approaches a limiting value, the detailed model has behavior that approaches the behavior of the approximate model. For a less complex model M_k and a more complex model M_l, the fitting approximation relationship is true if there exists an approximation limit, and a parameter p_{k+j}, such that

$$\lim_{p_{k+j} \to \text{limit}} M_l(x_1, \ldots, x_n, p_1, \ldots, p_k, \ldots, p_{k+j}, \ldots, p_l) = M_k(x_1, \ldots, x_n, p_1, \ldots, p_k), \quad (2.1)$$

in which $x_1, \ldots, x_n$ are the control variables, $p_1, \ldots, p_n$ are the model parameters, p_{k+j} is the fitting-approximation parameter, and $k + j \leq l$.

Table 2.1 Examples of fitting approximations[a].

Domain	Simplifying assumption	Fitting-approximation parameter	Approximati on limit
Physics	Inelastic string	Hooke's constant	∞
Physics	Massless object	Mass	0
Physics	Nonrelativistic motion	Speed of light	∞
Electronic circuits	Constant resistivity	Temperature-coefficient of resistivity	0
Thermodynamics	Perfect thermal conductivity	Coefficient of thermal conductivity	0
Hydrodynamics	Momentumless flow	Fluid density	0
Economics	Perfect macroeconomic competition	Number of firms in Cournot-theory model	∞

a. Adapted from [Weld, 1992].

The concept of fitting approximations is general, and many simplifying assumptions that are applied to models can be expressed in the form of fitting approximations. Table 2.1 lists several simplifying assumptions and their corresponding fitting approximations for models in the domains of physics, hydrodynamics, thermodynamics, electronic circuits, and economics. For example, consider a model that describes Hooke's law

$$y = M_1(F, k_H) = \frac{F}{k_h}, \quad (2.2)$$

in which M_l is a model that predicts the string elongation, y, as a function of force applied, F, and Hooke's constant for the string, k_H.

In the limit, as Hooke's constant, k_H, approaches infinity, the elongation approaches zero, and this relatively more detailed model is approximated by the simpler model

$$y = M_k(F) = 0. \tag{2.3}$$

To discover the effect of switching between models, we perform MSA by taking the partial derivative of the detailed-model with respect to the fitting parameter:

$$\frac{\partial y}{\partial k_h} = \frac{\partial M_l(F, k_H)}{\partial k_H} < 0. \tag{2.4}$$

The partial derivative of the model prediction (y) with respect to the fitting-approximation parameter (k_H) is negative, so the prediction for y will decrease as k_H increases. MSA predicts that y will decrease in models that assert the inelastic-string assumption.

Not all simplifying assumptions can be expressed as fitting approximations. For example, for a model of string length as a function of applied force, the simplifying assumption of an unbreakable string cannot be expressed as a fitting approximation. A more detailed model that has tensile strength as a fitting parameter predicts discontinuous behavior whenever the applied force exceeds the minimum breaking tension. As the breaking tension approaches infinity, the detailed model predicts discontinuous behavior for forces that approach infinity, whereas the simplified model that asserts *unbreakable string* predicts continuous behavior.

Other simplifications that involve approximate computation methods may not be represented as fitting approximations. For example, a piecewise-linear approximation of a dynamic system cannot be expressed as a fitting approximation, because the number of linear-approximating segments is not a parameter of the more detailed model [Sacks, 1988; Sacks, 1991].

Weld demonstrated the feasibility and the difficulty of MSA by implementing a program, called SAM, that performed discrepancy-driven model refinement in a GoM. SAM selected models from a GoM of ordinary differential equation (ODE) models, but required that the GoM include a corresponding qualitative differential equation (QDE) model for each ODE model. Whenever a discrepancy between an ODE model and an observation occurred, SAM attempted MSA by simulating each QDE model before and after moving the fitting-approximation parameters toward their approximation limits. SAM suffered from three problems. First, SAM required that the QDE models have unique qualitative behaviors, yet most complex QDE models do not meet this criterion. Second, the MSA algorithm was incomplete, because SAM applied DQ analysis to the results of qualitative simulations [Weld, 1988]. Finally, as Weld described, the tedium of creating linked ODE–QDE models precluded an implementation of more than two nodes in the GoM.

The AI approaches to modeling complex systems involve methods that reason about model structure and behavior; these methods combine knowledge of the system with some measure of model behavior (or of model-component behavior) to infer the structure of models that are accurate within the experimental frame under consideration. By contrast, the statistical approach to modeling typically considers only the quantitative behavior of alternative models, without considering the effect of model structure on model behavior. Statistical methods focus on *model selection*.

2.7 The Statistical Perspective: Model Selection

The classical statistical paradigm assumes that observations are generated by stochastic sampling of a known model. That is, the classical paradigm assumes that noisy observations of a system under study are the result of stochastic sampling from some model that is observably identical to the system. The goal of statistical methods is to discover the structure and the parameters of the *correct model* of a system. This paradigm suggests that a correct model exists, even though, “All models are wrong, therefore we cannot proclaim a correct one” [Box, 1985].

Box goes on to note that, because all models represent abstractions that approximate the observed behavior of a system, the issue becomes whether or not a model approximates the system well enough to be *useful*. A model does not have to be correct to be useful.

2.7.1 Statistical Model Terminology

A model M has a corresponding state vector Z that includes vectors of parameters, Θ, control variables, X, and observable variables, Y. For models that have a steady-state solution, model predictions are a function of Θ and X:

$$\hat{Y}_i = M_i(X, \hat{\Theta}_i), \tag{2.5}$$

in which $\hat{Y}_i$ is the vector of model predictions, and $\hat{\Theta}_i$ is the estimate of the vector of adjustable model parameters. The subscript i is an index that refers to one model in a set of models
$\mathtt{M} = \{M_0, M_1, \ldots, M_k\}$.

The difference between an observation of Y and the model prediction $\hat{Y}_i$ is the error vector ε_i:

$$Y = \hat{Y}_i + \varepsilon_i. \tag{2.6}$$

The vector Θ_i is a set of free parameters whose values may be adjusted to cause a model to predict values for $\hat{Y}_i$ that correspond as closely as possible to the observations of Y. For statistical models, the **dimension** of a model is the length of Θ. A major challenge in modeling a system is to determine the appropriate model dimension from a set of observations.

2.7.2 Model-Selection Criteria: Maximum Likelihood

Builders of statistical models first define a set of alternative models, then apply various model-selection criteria to choose a model that most closely reproduces the observations. A common method is to evaluate exhaustively a selection criterion for all the alternative models, then to select that model with the most favorable criterion.

Model-selection criteria are related to the concept of **model likelihood**. The likelihood of a model, $p(Y \mid M, \Theta)$, is the probability density that a set of observations, Y, would occur if the model, M, and its associated parameters, Θ, were a true representation of the system that generated the data. For the case of independent, normally distributed observation errors, the likelihood for the ith model is

$$L_i = p(Y|M_i, \Theta_i) = \prod_{j=1}^{N} e^{-\frac{1}{2}\left(\frac{y_{o_j} - \hat{y}_{i_j}}{\sigma_j}\right)^2}, \tag{2.7}$$

in which $\hat{y}_{i_j}$ is the ith-model estimate for y_j, y_{o_j} is an observation of y_j, and σ_j is the standard deviation of the observation errors for y_j. The term $\Delta\sigma$ is a small interval that converts the normal density function to a point probability; it is often ignored, because it is an arbitrary constant.

This definition of L_i assumes that the errors in the observations are independent and are normally distributed, and have a known variance. The definition also assumes that the observations used to compute the probability of each model are not first used to compute the parameters of each model.

If we take twice the negative logarithm of the likelihood, and rearrange, we obtain

$$-2\log L_i = \sum_{j=1}^{N} \left(\frac{y_{o_j} - \hat{y}_{i_j}}{\sigma_j}\right)^2, \tag{2.8}$$

which reduces to

$$\log L_i = -\frac{1}{2} R_i \quad . \tag{2.9}$$

R_i is the weighted sum of squared residuals for the ith model. The *maximum likelihood of the ith model*, $L_i^{\max}$, is the model likelihood evaluated at the maximum-likelihood values of its parameters, which occurs when R_i is a minimum.

Models with a higher dimension are able to explain more features in a set of data than are models with a lower dimension. If the dimension of a model exceeds the dimension of the system that generated the data, then the additional dimensions of the model will cause the model to *overfit* the data. For example, Figure 2.2 shows the results of fitting a seventh-order polynomial model to observations of a first-order polynomial system. The seventh-order polynomial model overfits the data, and makes poor predictions for the behavior of the linear system outside the range of observations.

In general, models will have a greater likelihood ($L_i^{\max}$) as their dimension increases, as long as the dimension of the model is less than the dimension of the data. Models that have a dimension that is higher than the dimension of the system that generated the data will reproduce the noise in the observations. A model-selection method based on $L_i^{\max}$ alone leads to selection of models that have an inappropriately high dimension. We would like to choose the model with a dimension that matches the dimension of the system that generated the data, rather than one that matches the dimension of the data.

The problem of finding a model that is at the right level of detail to explain fully a set of observations without explaining random observation errors has received considerable attention in the statistical model-selection literature. A model-selection method that simply minimizes $-L_i^{\max}$ is unsuitable. Numerous model-selection criteria attempt to modify $L_i^{\max}$ to create a criterion that leads to selection of models that are at the right level of detail.

An estimate of the maximum likelihood forms the basis of all model-selection criteria, because $L_i^{\max}$ is an essential measure of the goodness of fit of a model to the observations. Measures of goodness of fit of a model to the observations do not necessarily demonstrate that a model has good predictive power, but a model that is unable to explain the observations is unlikely to make reliable predictions for subsequent observations [Box, 1985].

2.7.3 Statistical Model–Selection Criteria

Numerous statistical model-selection criteria adjust measures of the likelihood to compensate for the tendency of higher-dimension models to overfit the data. These adjustments are penalty terms that increase in magnitude as the number of free parameters increases.

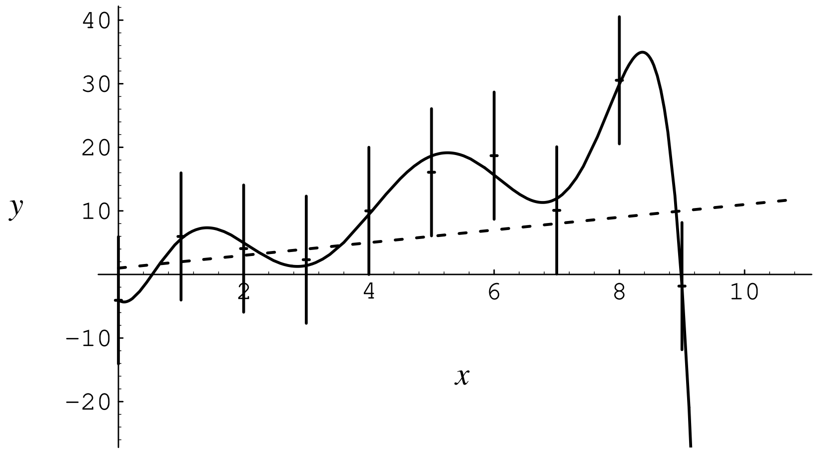

Figure 2.2 Maximum-likelihood fit of a polynomial model (M_7) to observations of a linear system (S). Data are sampled from the system $S = y = 1 + x$ (shown as dashed line) with an unbiased Gaussian sampling error of standard deviation $\sigma = 10$. Error bars on the observations show $\pm\ 1\sigma$. M_7 is a seventh-order polynomial model of form $M_i = y = \mathrm{a}_0\, x^0 + \mathrm{a}_1\, x^1\ +\ \mathrm{a}_2\, x^2 + ... + \mathrm{a}_i\, x^i$, $i = 0, 1, ..., N - 1$. The best fit of M_7 to the observations is plotted as a solid line. Although M_7 predicts all observations accurately, M_7 does not make accurate predictions of the behavior of S beyond the range of observations ($x < 0$, $x > 9$). The minimum sum of squared residuals, R_i, for fits of the polynomial models M_i to these data decrease progressively as i increases, until $R_i = 0$ at $M_i = M_N$. (For $i = 0, ..., 9$, the R_is are 9.76, 7.43, 5.86, 4.6, 3.26, 3.11, 1.15, 0.219, 0.077, 0.000). M_N fits the data exactly—and reproduces the random noise in the observations.

Many criteria match the pattern of the **penalized likelihood criteria** (PLC)

$$-\,2\log L_i^{\max} + \text{penalty terms}\,, \tag{2.10}$$

in which $L_i^{\max}$ is the likelihood of the ith model evaluated at the maximum-likelihood values of the parameters, m_i is the number of fitted parameters in the ith model, N is the number of observations, and the penalty terms vary among the various criteria [Sclove, 1994].

Table 2.2 Summary of information criteria.[a]

Criterion: $-2 \log L_i^{max}$ + penalty terms	penalty terms
Akaike information criterion (AIC)	$2\, m_i$
Bayesian information criterion (BIC)	$m_i \log N$
Modified Bayesian information criterion (MBIC)	$m_i (\log N - \log 2\pi)$
Kashyap information criterion (KIC)	$m_i \log N + \log \hat{\boldsymbol{I}}$

a. All criteria are of the form $-2 \log L_i^{max}$ +penalty terms. L_i^{max} is the likelihood of the *i*th model evaluated at the maximum-likelihood estimates of the parameters. $|\hat{\boldsymbol{I}}|$ is the observed information matrix, evaluated at the maximum-likelihood estimates of the parameters [Sclove, 1994]. Log is the natural logarithm.
The criteria are discussed in more detail in Sections 2.7.3.2 – 2.7.3.5, and the penalty terms are compared in Figure 2.3.

Akaike derived the first PLC by applying a maximum-entropy principle that involved minimizing the Kullback information measure [Akaike, 1973; Rissanen 1978]. Akaike named his selection criterion the **Akaike information criterion** (AIC); other researchers now apply the term *information criterion* to model-selection criteria that are derived from other, non–information-theoretic approaches.

The AIC, BIC, MBIC and KIC all have interpretations as measures of the posterior probability of a model given the observations (see Table 2.2). An expansion of this posterior probability shows the relationships among these criteria.

2.7.3.1 Expansion of the Posterior Model Probability

Consider the set of mutually exclusive and collectively exhaustive models $\mathtt{M} = \{M_0, M_1, \ldots, M_k\}$. Each model M_i predicts a set of observable variables Y. Let the hypothesis that a model M_i is correct be denoted M_i^c, $0 \le i \le k$. Then, M_i^c is true if M_i predicts behavior indistinguishable from that of the system being modeled. That is, M_i^c is true if and only if there exist a set of parameters, Θ_i such that

$$Y_c = \hat{Y}_i = M_i(\Theta_i, X) = Y - \varepsilon, \text{ for all } X, \tag{2.11}$$

where Y_c is the set of the true values of the l observable variables $\{y_{c_1}, y_{c_2}, \ldots, y_{c_l}\}$, X is the set of u control variables $\{x_1, x_2, \ldots, x_u\}$, Y is the vector of steady-state observations $\{y_1, y_2, \ldots, y_l\}$ that correspond to X, and ε is a vector of random variables. The symbol Y represents a vector of observations that correspond to a single setting of the vector of the control variables, X. The boldface symbols $\boldsymbol{Y}$ and $\boldsymbol{X}$ represent sets of values of the vectors Y and X.

The idea behind a Bayesian analysis of the model-selection problem is that the ideal model to select is the *maximum a posteriori* (MAP) model—the model that has the highest posterior probability of being a correct model. We can compute the a posteriori probability that the ith model is correct by applying Bayes' rule to derive

$$p(M_i^c | \boldsymbol{Y}) = Cp(M_i^c)p(\boldsymbol{Y}|M_i^c), \tag{2.12}$$

in which C is a scaling constant, equal to $\frac{1}{p(\boldsymbol{Y})}$.

The posterior probability is proportional to the product of the prior probability of a model, $p(M_i^c)$, and the conditional probability of the data given that the model is correct, $p(\boldsymbol{Y} | M_i^c)$. This formula assumes that the models are mutually exclusive and collectively exhaustive, so we can determine the value of the scaling constant, C, by setting the sum of posterior probabilities to 1. If the models are not collectively exhaustive, the sum of posterior probabilities of all models in the set $\mathtt{M}$ may be set to $(1-r)$, where r is the probability that no model in $\mathtt{M}$ is correct.

The calculation of $p(\boldsymbol{Y} | M_i^c)$ requires the evaluation of the integral

$$p(\boldsymbol{Y}|M_i^c) = \int p(\Theta_i|M_i^c)p(\boldsymbol{Y}|\Theta_i, M_i^c)d\Theta_i, \tag{2.13}$$

because the parameters of each model are not specified at the time of model selection. Laplace's method leads to an approximate analytic solution of Equation 2.13:

$$p(\boldsymbol{Y}|M_i^c) \cong (2\pi)^{m_i/2}|\hat{\boldsymbol{I}}_i|^{1/2}p(\boldsymbol{Y}|\hat{\Theta}_i, M_i^c)p(\hat{\Theta}_i|M_i^c), \tag{2.14}$$

in which $|\hat{\boldsymbol{I}}_i|$ is the determinant of the observed information matrix,[5] and $\hat{\Theta}_i$ is the vector of maximum-likelihood estimates of the parameters [Draper, 1993]. The Laplace method assumes only that the likelihood is peaked in the region of the maximum-likelihood value, which is true when N is at least of moderate size. The approximation has an error of order $O(N^{-1})$ [Kass, 1993; Draper, 1993].

Taking –2 times the natural logarithm of the left- and right-hand sides of Equation 2.14 gives

$$-2\log p(\boldsymbol{Y}|M_i^{\mathrm{c}}) \cong -2\log p(\boldsymbol{Y}|\hat{\Theta}_i, M_i^{\mathrm{c}}) - \log|\hat{\boldsymbol{I}}_i| - m_i \log 2\pi - 2\log p(\hat{\Theta}_i|M_i^{\mathrm{c}}). \quad (2.15)$$

Rearranging, and substituting $L_i^{\max}$ for $p(\boldsymbol{Y}|\hat{\Theta}_i, M_i^{\mathrm{c}})$ gives

$$-2\log p(\boldsymbol{Y}|M_i^{\mathrm{c}}) \cong -2\log L_i^{\max} + \log|\hat{\boldsymbol{I}}_i| - m_i \log 2\pi - 2\log p(\hat{\Theta}_i|M_i^{\mathrm{c}}). \quad (2.16)$$

For large N, $\log|\hat{\boldsymbol{I}}_i| \cong m_i \log N$, and the effect of the prior parameter distributions is overshadowed by the data, so

$$-2\log p(\boldsymbol{Y}|M_i^{\mathrm{c}}) \cong -2\log L_i^{\max} + m_i \log N - m_i \log 2\pi. \quad (2.17)$$

Comparison of Equations 2.16 and 2.17 with the information criteria in Table 2.1 shows that different information criteria include different elements of the expansion of the posterior probability of a model given the observations. All criteria include the first term $-2\log L_i^{\max}$, but the alternative criteria differ in which terms of Equation 2.16 and 2.17 they include in the likelihood penalty.

2.7.3.2 Akaike Information Criterion

The AIC is $-2\log L_i^{\max} + 2m_i$ —it sets the likelihood-penalty term to $2m_i$. The AIC approximates the terms $\log N - \log 2\pi$ (from Equation 2.17) as 2, which is true for sample sizes that have $N \approx 50$ (see Figure 2.3). The AIC penalty does not increase as N increases;

5. $\hat{\boldsymbol{I}}_i$ is the inverse of the negative Hessian matrix of second partial derivatives of the log likelihood, evaluated at the maximum likelihood estimates of the parameters.

so, for larger sample sizes, the AIC tends to overestimate the dimension of a model [Hannan, 1981; Shibata, 1976]. To counteract this weakness of the AIC, Bhansali proposed to increase the constant 2 in the AIC to 3 or 4 [Bhansali, 1977].

2.7.3.3 Bayesian Information Criterion

As pointed out elsewhere, the optimal value for the likelihood-penalty term cannot be expressed as a constant [Atkinson, 1980]. Schwarz noted that, under the assumption of nowhere-vanishing priors, the log posterior probability of a model asymptotically approaches $\log L_i^{max} - 1/2\ m_i \log N$. He ignored terms that were independent of N in the expansion of the log posterior probability $p(\boldsymbol{Y}|M_i^c)$, noting that, in the large sample limit, the effect of these terms vanishes. He suggested using a likelihood-penalty term of $m_i \log N$ for selecting the MAP model. The information criterion based on this penalty term is the **Bayesian information criterion** (BIC). Compared to the AIC, the BIC imposes a larger penalty on higher-dimensional models whenever $\log N > 2$ (that is, whenever $N \geq 8$). The BIC is asymptotically optimal, because, as the number of observations becomes large, the dimension of the model selected by minimization of the BIC converges to the dimension of the system that generated the data [Schwarz, 1978].

2.7.3.4 Modified Bayesian Information Criterion

The expansion of negative twice the log posterior probability includes a penalty term that varies with sample size ($m_i \log N$), and a penalty term that is independent of sample size ($-m_i \log 2\pi$). Although this second term is asymptotically unimportant, it will have a noteworthy effect on the magnitude of the log posterior probability for small and moderate sample sizes. Draper suggest that this term should be included in a Bayesian model-selection criterion, and that its inclusion improves model-selection performance for real problems [Draper, 1993]. I call the information criterion with likelihood-penalty terms $+m_i \log N - m_i \log 2\pi$ the **modified Bayesian information criterion** (MBIC).

2.7.3.5 Kashyap Information Criterion

Kashyap also considered the expansion of posterior model probability to suggest the *Kashyap information criterion* (KIC). He included in the penalty the logarithm of the determinant of the observed information matrix, $\log |\boldsymbol{I}|$, in an effort to improve the small-

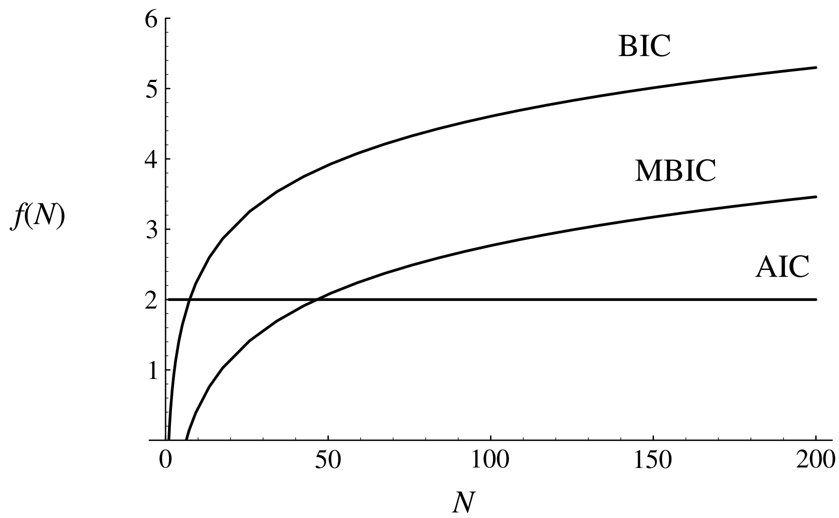

Figure 2.3 Comparison of likelihood penalties. I plot the rate of increase of the likelihood penalty (as the number of free parameters increases) as a function of the number of observations, for the BIC, MBIC and the AIC (see Table 2.1). Likelihood penalty = $m_i\, f(N)$; m_i: number of free parameters; N: number of observations; AIC: Akaike information criterion, for which $f(N) = 2$; BIC: Bayesian information criterion, for which $f(N) = \log N$; MBIC: modified BIC, for which $f(N) = \log N - \log 2\pi$.

sample performance of the KIC. Unfortunately, in the setting of small sample sizes, the estimate $|\boldsymbol{I}|$ is unreliable, which makes the benefit of this penalty term questionable [Kashyap, 1982; Sclove, 1994].

2.7.3.6 Minimum Description Length Criterion

In the original derivation of the AIC, Akaike evaluated the cross-entropy of the candidate models and of the correct model by minimizing the expectation of the differences in the Kullback information between the candidate models and the correct model. Rissanen pointed out that this approach is biased (that is, the approach is not asymptotically optimal as N increases), and proposed an alternative method based on the **minimum description length** (MDL) principle. This information-theoretic approach leads to expressions that describe the minimum model dimension that is required to explain the information content in a set of observations.

The MDL principle leads to model-selection criteria that are similar to the criteria derived from the Bayesian analysis. For example, for linear autoregressive moving-average (ARMA) models, the MDL principle leads to a model-selection criterion of the form

$$\log L_i^{\max} + \sum_j \log\left(\Theta_{i_j}^2 \frac{\partial^2}{\partial\Theta_{i_j}^2}\log L_i^{\max}\right) + m_i \log N, \tag{2.18}$$

in which the subscript j is the index of the $\{1,...,m_i\}$ parameters Θ_i.

2.7.4 Comparison of Information Criteria

The AIC, BIC, and MBIC all set likelihood penalties that increase linearly as the model dimension increases. The rate of increase in penalty as the number of fitted parameters increases depends on the sample size (N), and varies among the criteria. Figure 2.3 plots the rates of increase in penalty for these three information criteria for values of N between 1 and 200.

The likelihood penalties of both Bayesian criteria become progressively larger as N increases; for very large N, the small fixed difference between them (which is equal to m_i $\log 2\pi$) is unimportant, and $\lim_{N\to\infty}\frac{\text{BIC}}{\text{MBIC}} = 1$. Both the BIC and MBIC select a model of the correct dimension in the limit, for large N.

The AIC favors higher-dimensional models than does the BIC for all values of $N > 8$. Simulation studies of the AIC suggest that, for model-selection problems with moderate sample sizes, the AIC works well. Figure 2.3 shows that, for $N \approx 50$, the AIC and MBIC have approximately the same magnitude of penalty terms. These results suggest that the BIC favors models with fewer dimensions than are present in the system that generates the observations (under the assumption that the prior distribution on the set of alternative models is uniform).

Comparison of Equation 2.18 with Equation 2.16 shows that the selection criterion based on MDL principles has a striking similarity to the criterion based on the posterior model probability. Under the assumption that the term in the second derivative of the log likelihood is ignored, the MDL criterion is identical to the BIC.

2.7.5 Prior Probability of a Model

The information criteria are derived from the posterior model probability, but they ignore the prior probability of the models. For large N, the priors are in fact unimportant, and the criteria converge on the correct model dimension, as long as the priors are nowhere vanishing [Schwarz, 1978]. For smaller sample sizes, the priors may be more important. The posterior-probability analysis (as summarized in Equations 2.16 and 2.17) ignores the prior distribution; the use of the MBIC for model selection with moderate sample sizes should be valid as long as the prior distributions are uniform.

The use of an information criterion other than the MBIC makes an implicit assumption that the prior distribution on the alternative models is not uniform. The following derivation shows what the implied assumptions are.

Taking the natural logarithm of the expression for the posterior probability of a model (Equation 2.12) gives

$$\log p(M_i^c|\boldsymbol{Y}) = \log p(M_i^c) + \log p(\boldsymbol{Y}|M_i^c) + \log C. \tag{2.19}$$

If the prior distribution on the models is uniform (the priors are uninformative), then $p(M_i^c)$ is a constant, and

$$\log p(M_i^c|\boldsymbol{Y}) = \log C^\circ + \log p(\boldsymbol{Y}|M_i^c), \tag{2.20}$$

in which $\log C^\circ = \log C + \log p(M_i^c)$.

The information criterion (IC) is an estimate of $-2\log p(\boldsymbol{Y}|M_i^c)$, so

$$\log p(M_i^c|\boldsymbol{Y}) = \log C^\circ - \frac{1}{2}\mathrm{IC}. \tag{2.21}$$

The information criterion is at a minimum for the MAP model if the prior distribution on the models is uniform.

The information criteria described in the previous section delete or add terms in the expansion of the log posterior probability of the models:

$$\mathrm{IC} = -2\left(\log p(\boldsymbol{Y}|M_i^{\mathrm{c}}) - \text{missing terms}\right). \tag{2.22}$$

If the priors on the models *are* uniform, then the information criterion is not at a minimum for the MAP model. Furthermore, if

$$\log p(M_i^{\mathrm{c}}|\boldsymbol{Y}) = \log p(M_i^{\mathrm{c}}) - \frac{1}{2}\mathrm{IC} + \log C, \tag{2.23}$$

then

$$\log p(M_i^{\mathrm{c}}) = \log C^{\circ} + \text{missing terms}. \tag{2.24}$$

That is, deleting a term from the expression for the information criterion has the effect of making the assumption that the log prior is equal to a constant plus the missing term. For example, consider the missing terms in the BIC estimate of $\log p(\boldsymbol{Y}|M_i^{\mathrm{c}})$. The BIC ignores the penalty term $-m_i \log 2\pi$ (see Equation 2.17), which is equivalent to the addition of $-m_i \log 2\pi$ to the log of uniform priors. The use of the BIC to select the MAP model implies that the prior probabilities of the models are proportional to $(2\pi)^{-m_i}$.

Akaike points out that changing the AIC likelihood penalty from $2m_i$ to $4m_i$ has the same effect on the log posterior probability as does adding $2m_i$ to the log prior. This change in the AIC is equivalent to asserting that the prior distribution is proportional to e^{2m_i} [Akaike, 1979]. The differences among alternative model-selection criteria imply different prior distributions for the models. For large sample sizes, these differences in prior distributions are unimportant.

2.8 Effect of Prior Distribution on the MAP Model

The model-selection criteria presented in Section 2.7 are approximations of a measure of the log conditional probability of a model given the observations. As such, these criteria are

MAP model-selection criteria only under specific prior distributions for the set of alternative models.

Information criteria that are based on the expansion of $\log p(\boldsymbol{Y}|M_i^c)$ ignore the contribution of the prior distribution ($p(M_i^c)$) to the posterior probability of a model (see Table 2.1). For example, the MBIC includes three terms in the expansion of $-2\log p(\boldsymbol{Y}|M_i^c)$:

$$\text{MBIC} = -2\log L_i^{\max} + m_i \log N - m_i \log 2\pi, \tag{2.25}$$

but includes no term for the effect of the prior distribution $p(M_i^c)$ on the posterior probability of a model.

The contribution of the prior probability to the posterior probability of a model is unimportant under two conditions. First, in the case of a uniform prior distribution, the prior probabilities of all models are the same, and the prior distribution has no effect on which model has the highest posterior probability. Second, if the number of observations is large, and if the prior distribution is nowhere vanishing, then the prior has a negligible contribution to the posterior probability. That is, as the number of observations becomes large, the likelihoods of the incorrect models become small, and the data overwhelm the priors [Schwarz, 1978; Akaike, 1979]. The reason the prior must be nowhere vanishing is that, whenever the prior for a model vanishes ($p(M_i^c) = 0$), the posterior probability of the model vanishes also ($p(M_i^c|\boldsymbol{Y}) = 0$), no matter how large the likelihood is for that model.

For smaller sample sizes, the prior distribution may have a noteworthy effect on the posterior probability distribution for the models, and the model-selection criteria that assume large sample sizes do not apply. The prior distribution of the models should be an important component of a model-selection criterion when N is small; the addition of prior information may allow confident MAP model selection when the likelihoods of the alternative models are similar.

2.9 Summary and Conclusions

In this chapter, we discussed two widely differing methods for reasoning about models. The AI approach to automated modeling allows for the selection (or the composition) of models when no, or few, observations of the system behavior are available. The AI approach applies qualitative knowledge of the domain to assess which alternative model is most likely to provide an accurate representation of the system. AI methods may consider sparse quantitative observations in a model verification step, but provide no method to incorporate a large set of quantitative observations in the assessment of the most probable model.

In the statistical approach to automated modeling, model-selection criteria estimate the distribution of probable models from a set of quantitative observations, under the assumption that prior knowledge about which model to select is unimportant. That is, a major assumption of the statistical perspective is that a sufficient number of quantitative observations will specify which model is most likely to represent the system that generated the observations. From a Bayesian perspective, the model-selection criteria are approximations of the log posterior probability of a model, under the assumption that the prior distributions on the models are uninformative. The differences among the criteria are equivalent to varied assumptions for the prior distribution on the set of alternative models. However, the statistical model-selection criteria provide no method to incorporate explicit prior distributions on the set of alternative models in situations where we know what those distributions are.

For modeling complex systems about which we have prior knowledge, and for which we have quantitative observations, neither an AI nor a statistical method alone is adequate. For example, to find an appropriate model of cardiopulmonary physiology for a patient who is in the ICU, we require a method that combines knowledge of the probable underlying physiologic abnormalities[6] with numerous quantitative observations of the patient state.[7] We should apply AI methods to reason with knowledge of such a complex system and to create an explicit prior distribution on the probability that each model will be an accurate

6. An example of knowledge of underlying abnormalities is a clinician's bedside assessment that a patient has asthma (narrowing of the airways)—which indicates that a model of that patient should include a detailed description of the airways.
7. Examples of quantitative observations of patients in the ICU include the partial pressures of oxygen and carbon dioxide in the arterial and venous blood, and the cardiac output.

predictor of system behavior. We can then combine this prior distribution with the quantitative assessment of model performance, as provided by the statistical model-selection criteria. In Chapter 5, I present a method that combines AI and statistics approaches to define a measure of the benefit of a model (see Section 5.6).

In the next chapter, I present background work on the VentPlan project that establishes the need to apply more detailed and computationally complex models of physiology to critically ill patients in the ICU. I then describe a set of such models in Chapter 4, before returning to describe a method to select a model that balances the benefit of model accuracy with the cost of model-induced computation delay.

Chapter 3

Model-Based Ventilator Management

Researchers have implemented numerous approaches to automate the interpretation of ICU-patient data and the generation of treatment advice. In this chapter, I review these approaches, present background on the use of mathematical models of physiologic systems, and then describe an automated **ventilator-management advisor** (VMA), called VentPlan, that assists the monitoring and care of patients who are treated with a ventilator in the intensive-care unit (ICU). The VentPlan prototype demonstrates the need to incorporate more detailed physiologic models in a real-time ICU program. The need to incorporate computationally complex physiologic models in an improved VMA led me to develop the method for the dynamic selection of models under time constraints that I shall present in Chapters 5 and 6.

3.1 Quantitative and Qualitative Methods for Interpretation of ICU Data

Numerous techniques have been developed to interpret the data collected in ICU [Gravenstein, 1983]. These techniques are derived from different problem-solving approaches that have been applied to medical decision-making problems [Shortliffe, 1979].

Numeric methods interpret quantitative observations, such as the arterial partial pressure of oxygen, but cannot easily incorporate qualitative observations, such as "the patient has pneumonia." For example, Coleman and colleagues applied a purely numeric method to abstract, from 11 cardiovascular and metabolic measurements, a state-space vector that classified a patient's physiology. Although this classification led to estimates of the probability of survival, the classification was unable to incorporate other qualitative observations [Coleman, 1990].

To take advantage of qualitative information, ICU-data interpretation programs require some form of symbolic or qualitative representation. Several programs included qualitative information in the interpretation of patient data by applying *rule-based symbolic models*. For example, VM interpreted cardiovascular and ventilator measurements to make recommendations for changing the settings of the ventilator [Fagan, 1984], and VQ-Attending provided critiques of suggested changes in settings of a ventilator [Miller, 1985]. These programs recognized specific situations in which they could make recommendations for incremental changes to the ventilator settings. These changes were incremental because the programs were unable to predict quantitatively the effects of ventilator settings. The models implicit in these programs were not able to resolve situations in which conflicting evidence suggested opposite courses of action.

Models to Support Ventilator Management

Automated methods that monitor the physiologic status of patients in the ICU must reason about, and respond to, changes in numerous observable patient variables—such as cardiac output, airway pressure, systolic and diastolic blood pressures, heart rate, and arterial and venous partial pressures of oxygen and carbon dioxide. Alternative approaches to physiologic modeling may represent these entities either as *discrete variables* or as *continuous variables*.

Discrete Versus Continuous Model Variables

Rule-based, protocol-based, and qualitative-model–based approaches to reasoning are *discrete methods*, because they represent the physiologic state of patients with a discrete representation for each variable. Discrete methods interpret the values for each variable as one of a small number of discrete states, such as "high," "normal," and "low." Discrete methods also recommend actions as discrete changes in the values of one or more of the control settings (control variables) of the ventilator. For example, a discrete method might recommend the actions "increase the fraction of inspired oxygen" and "decrease the volume of each breath."

Researchers developed qualitative models of physiology to predict future changes in physiologic state, but found that the qualitative-modeling methodology was not suitable for the prediction of changes in state of a physiologic system [Kuipers, 1985; Uckun, 1993]. The problem that a discrete representation of the physiologic state of ICU patients causes is that the number of possible combinations of discrete states of the observable patient variables is large.[1] This combinatoric problem is made worse because the categories for each variable depend on the *context* in which we interpret the observed variables. For example, an elderly patient who has atherosclerosis may have higher-than-average systolic blood pressures, which value we should consider "normal" for her. If the same patient were to develop a cerebral infarction (stroke), then we would expect her to have an even higher blood pressure, and that high value would be normal in that situation. In practice, the circumstances of the patient define a context that sets the range of values for the discrete categories of each variable.

Protocols for ventilator management provide recommendations for discrete changes in ventilator settings based on discrete variables that describe a patient's physiology. For example, a protocol might indicate that, in a stated context, "If $P_aO_2 < 55$ mm Hg, then increase FIO_2 by 0.05"). The large number of possible discrete patient contexts makes the number of combinations of discrete physiologic variables enormous. Nevertheless, Morris and colleagues developed computer-based protocols for adjusting the ventilator for patients who have adult respiratory distress syndrome (ARDS); these protocols appears to provide improved care for such patients [Thomsen, 1989; Sittig, 1989; Morris, 1994]. The success of the protocol-based approach to ventilator management demonstrates the problem of the combinatorics of multiple contexts—a protocol applies to the management of only a single observed variable for a narrowly defined group of patients who suffer from one pathophysiologic problem [Henderson, 1991; East, 1992].

1. For n variables that may take on one of m states, there are n^m possible combinations of states. Nine observable variables plus four control variables, with three states per variable, results in 2197 combinations.

A fundamental difficulty with protocol-based management of ventilator settings is that protocols hide the underlying knowledge on which they base their recommended treatments. Protocols that guide medical treatments suffer from additional disadvantages: They are labor intensive to create, they may contain internal inconsistencies, and they may become obsolete as soon as new information or new modalities of therapy become available.

To reason about the effect of changes in ventilator settings, a VMA should implement models that make accurate, quantitative, patient-specific predictions of the effects of alternative settings of the ventilator. Only mathematical models that include a continuous representation of their variables can make such predictions. In the remainder of this chapter, I describe VentPlan, an ICU application that is based on a quantitative mathematical model of cardiopulmonary physiology.

3.2 VentPlan: A Prototype Ventilator-Management Advisor[2]

VentPlan is a prototype VMA that interprets physiologic measurements of patients in an ICU who are being treated with a ventilator, and recommends settings for four controls of the ventilator: fraction of inspired oxygen (FIO_2), tidal volume (V_T), ventilator rate (RR), and positive end-expiratory pressure ($PEEP$). VentPlan calculates recommended settings for the controls of the ventilator by evaluating the predicted effects of alternative ventilator settings.

VentPlan relies on a physiologic model to interpret clinical observations and monitored data, and to predict the effects of alternative treatments. VentPlan implements a model that is simplified, to minimize the computation-time delay resulting from repeated model evaluation. The real-time requirements of ICU-patient care impose a limit on the computation time that is available for a VMA's physiologic model to make predictions. During a single cycle of data analysis, VentPlan evaluates its mathematical model over 1000 times. If a single evaluation of VentPlan's model were to take 1 second of computation

2. VentPlan was developed by numerous researchers at Stanford University's Section on Medical Informatics: Geoffrey Rutledge, George Thomsen, and Brad Farr, with the assistance of Lawrence Fagan, Lewis Sheiner, Jeanette Polaschek, Maria Tovar, Richard Peverini, Michael Kahn, and Ingo Beinlich. Much of the work described in this chapter has been presented elsewhere [Rutledge, 1989; Rutledge, 1990; Rutledge, 1991; Rutledge, 1993a; Thomsen, 1989; Farr, 1989; Tovar, 1991].

time, VentPlan would require 10 minutes to complete a cycle of data interpretation and ventilator-setting recommendation. In order to achieve a response time of less than 2 minutes, we implemented a simplified model—one that represents only fundamental physiologic principles—as a first step.[3]

VentPlan demonstrates that a quantitative physiologic model can form the basis of an intelligent ventilator monitor, and, in so doing, highlights the need for accurate, patient-specific models that are computable within the time constraints of ICU-patient problems. VentPlan motivates us to ask: How can an automated system find a model that contains sufficient detail to explain a patient's physiologic abnormalities accurately, and that also is simple enough to make predictions within the time available? The performance of VentPlan's physiologic model gives us a reference against which we can evaluate more detailed models that impose longer computation delays to calculate treatment recommendations.

3.3 VentPlan Architecture

VentPlan has four components (see Figure 3.1):

1. A *belief network* calculates probability distributions of shared physiologic model parameters from qualitative inputs (such as "pulmonary edema is TRUE") and from semiquantitative inputs (such as "the central venous pressure is high {10 to 25 centimeters of water}").

2. A *mathematical-modeling module* implements a quantitative model of the circulation of oxygen and carbon dioxide. This physiologic model predicts the effects of alternative ventilator-control settings, and allows VentPlan to search for the settings that optimize the patient's predicted respiratory status. The mathematical-modeling module combines quantitative patient observations with probability distributions calculated by the belief network to estimate the patient's physiologic parameters, which makes the model predictions patient specific.

3. A *plan evaluator* ranks therapy plans (proposed ventilator settings) and their predicted effects, based on a multiattribute value model.

3. The real-time constraint for computation of a new recommended action varies as a function of the patient's rate of change in physiologic state. VentPlan makes the assumption that a response within 2 minutes of the occurrence of new observations will satisfy this constraint.

4. A *graphical interface* displays the monitored data from which the patient-specific physiologic parameters are estimated, shows the current and recommended ventilator settings, and allows the user to interact with the mathematical model to simulate alternative ventilator settings.

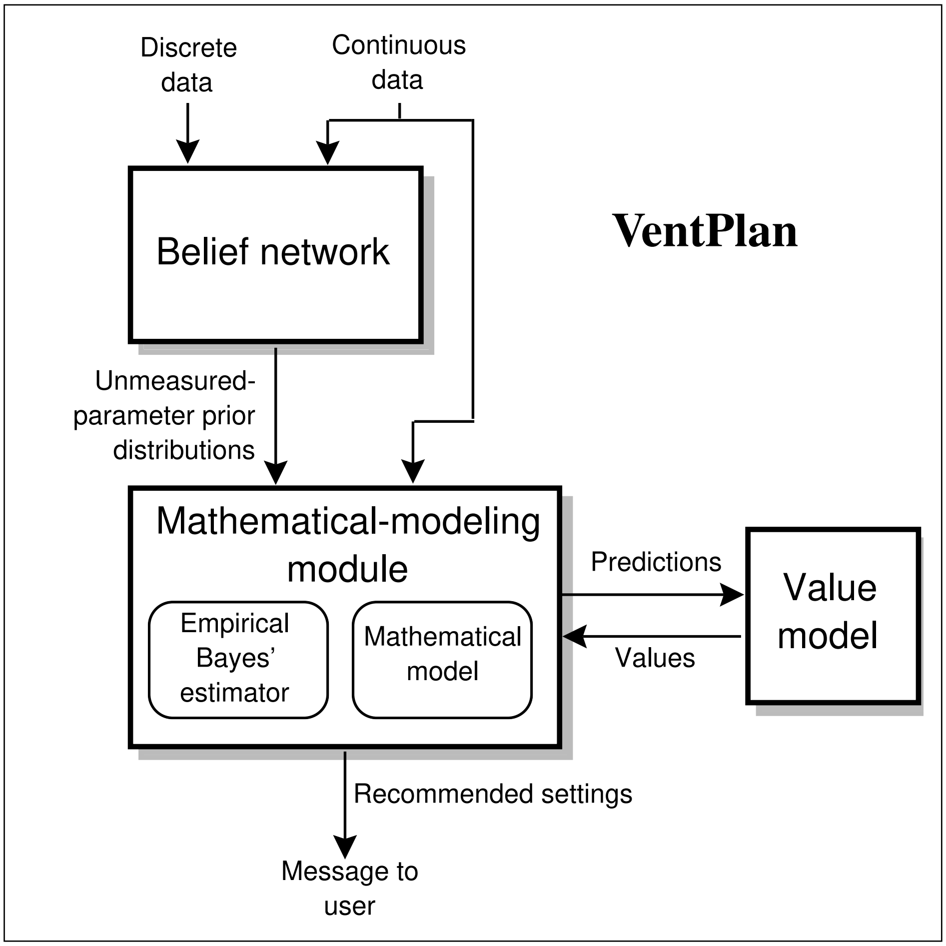

Figure 3.1 VentPlan architecture. Shadowed rectangles show the relationships of the belief network, plan evaluator, and mathematical-modeling module. Arrows show the flow of information among these components during each cycle of data interpretation. Initially, discrete data (such as the diagnosis) and numeric data not included in the mathematical model are given to the belief network for evaluation. The belief network converts the numeric data to categorical values, then evaluates the network to calculate probability distributions for the unmeasured parameters of the mathematical model. The empirical Bayes' estimator combines these prior distributions with other numeric data, such as the arterial blood-gas report, to estimate the physiological model parameters. The mathematical model simulates the effects of possible ventilator settings and the plan evaluator ranks the settings. Adapted from [Rutledge, 1990, p. 785].

3.3.1 Belief Network

Other researchers have demonstrated the usefulness of belief networks to solve medical problems, including hematopathologic disease diagnosis [Heckerman, 1989], electromyogram interpretation [Andersen, 1989], and anesthesia monitoring [Beinlich, 1989]. We use a belief network, called VPnet, to calculate probability distributions for the set of shared physiologic model parameters.[4] VPnet represents knowledge of pathophysiologic disease states in a causal probabilistic framework. The assumptions of the belief network—the assertions of conditional independence—are represented explicitly in a graphical structure. The network structure resolves conflicts in the inputs by combining evidence in internal nodes to assess the distributions on the parameter nodes. These parameter-node distributions are then used by VentPlan's mathematical-modeling module during the parameter-estimation procedure.

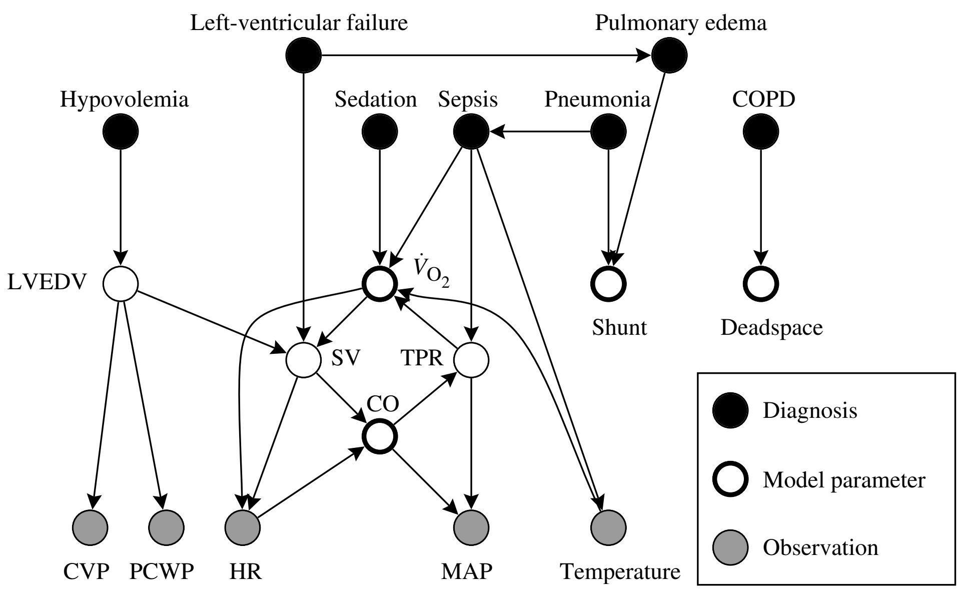

Figure 3.2 VPnet. The belief network represents causal relationships with diagnosis nodes, model-parameter nodes (intermediate nodes) and observation nodes (measured-variable nodes). CVP: central venous pressure; LVEDV: left-ventricular end-diastolic volume; PCWP: pulmonary capillary wedge pressure; HR: heart rate; SV: stroke volume; CO: cardiac output; TPR: total peripheral resistance; $\dot{V}_{O_2}$: oxygen consumption; MAP: mean arterial pressure; COPD: chronic obstructive pulmonary disease. Adapted from [Rutledge, 1990, p. 786].

4. The shared physiologic parameters are cardiac output ($\dot{Q}_T$), shunt ($\dot{Q}_s$), deadspace (V_{ds}), oxygen consumption($\dot{V}_{O_2}$), and respiratory quotient (RQ).

VPnet Structure

VPnet includes seven diagnosis nodes, five monitored data nodes, and seven intermediate nodes (Figure 3.2). The diagnoses represented are conditions that commonly affect patients who receive ventilator therapy in the ICU. The diagnoses are represented as TRUE–FALSE nodes, with prior probabilities based on the distribution of pathophysiologic disease states of ICU patients. The intermediate nodes combine the diagnosis with the data from hemodynamic monitoring. Four intermediate nodes correspond to parameters of the quantitative physiological model; these parameters are the cardiac output ($\dot{Q}_T$), the shunt ($\dot{Q}_s$), the oxygen consumption ($\dot{V}_{O_2}$), and the deadspace (V_{ds}). Three intermediate nodes correspond to physiologic concepts necessary for the interpretation of the hemodynamic data, but are not included directly in the mathematical model: these parameters are the left-ventricular end-diastolic volume, the stroke volume, and the total peripheral resistance. The measured-variable nodes correspond to clinical variables that are monitored in the ICU; we include in VPnet only those measured variables that are not included in the mathematical model.

For example, inputs to VPnet might be "congestive heart failure TRUE," "V_T 800 milliliters," and "temperature 35.5° C." VPnet classifies this tidal volume as "normal," and this temperature as "low." The corresponding output would be the probability distribution for each of the parameters shared with the mathematical model—for example, $\dot{Q}_T$ 4.4 ± 2.3 liters per minute (mean ± variance), and $\dot{V}_{O_2}$ 180 ± 120 milliliters per minute.

3.3.2 Communication of VPnet Results to the Mathematical-Modeling Module

VPnet calculates prior probability distributions on the physiologic parameters by converting the network's discrete parameter-node distributions to equivalent normal distributions. The means and variances of these distributions are passed to the mathematical-modeling module for use during the estimation of patient-specific parameter values.

3.3.3 Mathematical-Modeling Module

VentPlan's mathematical-modeling module implements a mathematical model of cardiopulmonary physiology. A set of linked first-order differential equations describes a parameterized compartment model of exchanges of oxygen and carbon dioxide in the lungs and tissues, and of transport of these gases through the body. We derived this model from the classic three-compartment model of pulmonary gas exchange [Riley, 1949].

VentPlan calls on the mathematical-modeling module for three tasks:

1. *Parameter estimation:* We apply the technique of empirical Bayes' estimation [Sheiner, 1982] to adjust the population prior distribution[5] of each model parameter to obtain an approximate posterior distribution in light of the measured data. The prior parameter distributions are calculated by the belief network. Thus, the estimate for a parameter is strongly influenced by the clinical context when the observations do not restrict the parameter to a single value.

2. *Prediction:* Based on the updated parameter distributions and on the control variables (the settings of the ventilator), the patient-specific model predicts the distribution of values for each dependent variable of the model for any specified future time. The dependent variables are the oxygen and carbon-dioxide partial pressures in each compartment.

3. *Optimization*: The mathematical-modeling module works with the plan evaluator to optimize the control variables. Using the separation principle, and assuming the parameters are equal to the mode of their posterior distributions, we search for the settings of the ventilator that result in the patient state with the highest expected value, as determined by the plan evaluator.[6]

3.3.4 Plan Evaluator

The plan evaluator is based on a multiattribute value model that provides a relative ranking of plans and of their predicted consequences [Farr, 1989]. The attributes of the value model are the proposed FIO_2 and *PEEP* settings of the ventilator, and selected predictions of the model. The model predictions used are the arterial partial pressure of carbon dioxide (P_aCO_2) and the O_2 delivery. We calculate O_2 delivery as the product of $\dot{Q}_T$ and arterial O_2 saturation (S_aO_2). We determine a value for each of the attributes using a value function that we derived by direct assessment from physicians experienced in ventilator therapy. We

5. The population prior distribution of each parameter describes the unconditional probability distribution of the parameter in the population to which the patient belongs; VPnet calculates this distribution.
6. We set the model parameters to the mode of their posterior distributions to calculate the mode of the dependent-variable distributions. We estimate the variance of the dependent-variable distributions as a function of the variance of the posterior parameter distributions and of the variance due to unwarranted assumptions and model error [Sheiner, 1982]

weight these values, and then sum them to obtain an overall value for the plan. This value assumes that the predictions for an alternative plan are certain. To take into account the uncertainty of the model predictions, the plan evaluator calculates the expected value of each plan from the distributions for the predictions of each attribute, by making the assumption that these distributions are independent. The mathematical-modeling module calls on the plan evaluator during the optimization task to give the calculated value of each proposed setting of the ventilator.

3.3.5 Control Algorithm

A control algorithm reads in new data, checks for inconsistent or redundant data, directs the belief network and mathematical-modeling module to perform their functions as needed, compares the current ventilator settings with the calculated optimal settings, and generates text messages for the user based on this comparison. The portion of the control algorithm shown in Figure 3.2 describes how patient data are provided to the belief network and to the mathematical-modeling module. New data for input nodes in the belief network are first compared to their previous values. If any input to the network has a new discrete value, the network recalculates the probability distributions of the shared physiologic parameters. Note that, because network node values are categorical, a patient observation must change substantially to be classified to an alternate discrete value. For example, a measurement for the central venous pressure of 5 mm Hg would be classified as “normal,” and would have to fall to less than 0 to be classified as “low,” or to rise to greater than 10 to be classified as “high.”

A new observation corresponding to a parameter or dependent variable in the mathematical-modeling module—such as a new value for the arterial partial pressure of oxygen (P_aO_2)—triggers a call to the mathematical-modeling module to update the patient-specific model. After each refinement of the patient-specific model, the algorithm calls the mathematical-modeling module again to search for the optimal setting. This setting is presented to the user as the recommended ventilator setting. The control algorithm compares the calculated value of the recommended setting to that of the current setting. If the difference exceeds a predefined threshold, a message to the user indicates that VentPlan has found a setting that may have significant advantages over the current setting; otherwise, a message indicates that the current and recommended settings are similar. VentPlan continuously monitors the stream of patient data, reapplying this control sequence to refine the model and to recalculate the recommended settings as new data become available.

When a change in the ventilator settings occurs, the mathematical-modeling module predicts the effect of the new settings. The plan evaluator then calculates the expected value of the new settings, and the algorithm compares this value with the values of the previously recommended and current settings. A warning is issued if the new settings have a predicted value lower than that of the settings in effect; otherwise, the new settings are compared with the recommended settings, and a message is posted indicating whether the new settings are predicted to be as good as the recommended setting.

3.3.5.1 Inconsistent Observations

We assume that the underlying physiologic entities represented by the model parameters are constant. VentPlan compares each new measurement for a variable or parameter of the mathematical model with the calculated prediction interval. If a new measurement is outside the model-prediction interval, then VentPlan analyzes the discrepancy as follows. If the datum falls outside the physiologic range for that measurement, then it is spurious, and VentPlan discards it. If the datum is within the physiologic range, and if no change in the ventilator settings has occurred, then either the measurement is in error, or the underlying physiologic state has changed over time. Evaluating an unusual observation to determine whether or not it is in error is beyond the scope of VentPlan, since multiple clinical factors that are not inputs to the system would have to be considered. For example, a patient with a sudden, unexpected drop in P_aO_2 should be evaluated at the bedside for clinical evidence that might suggest a pulmonary embolus.

When an observation is inconsistent, VentPlan posts a notice that the new measurement is incompatible with the previous observations, and allows the user to assert that the datum is in error or that the patient's state has changed. In the absence of user input, VentPlan proceeds on the assumption that the datum reflects a true change in patient state; it refits the model to the recent data, using the prior parameter distributions previously computed. Thus, VentPlan discards the previous quantitative observations (at least one of which is inconsistent), but maintains the prior clinical context, as computed by the belief network.

3.3.5.2 Prediction Errors

The occurrence of a prediction error after a change in ventilator setting indicates that the physiologic model is in error. Ideally, a VMA would reason with such prediction errors to find a model that more accurately explains the observations. Since we have not yet implemented a model-selection mechanism, VentPlan alerts the user to any prediction errors, then refits the model to the new observations. Because VentPlan's model is refit to the measurements taken after the change in ventilator settings, it continues to match the observed behavior. The model predicts accurately the effect of small changes in ventilator settings, however, so it should continue to predict the correct direction— if not the correct magnitude—of the optimal ventilator-setting change.

3.4 The VentPlan Interface

The VentPlan interface is a monitor panel that displays pulse-oximeter data, arterial-blood gas (abg) analysis, and measurements of the $\dot{Q}_T$, along with the current and recommended ventilator settings and a text message interpreting the difference between these settings. A graphical display presents the time-ordered patient data on which are based the estimates of the physiologic-model parameters, and allows the user to superimpose the model predictions on graphs of the data. The user may inspect the model parameters and predictions for the model variables in a separate panel. The interface allows users to modify the data display, to begin a simulation, or to enter a new diagnosis. The interface mimics the control panel of the Puritan-Bennet model 7200 ventilator, and includes the controls for adjusting the settings on the ventilator, although we do not allow the user to adjust the physical ventilator through the VentPlan interface.

The user asserts or retracts diagnoses by clicking on buttons corresponding to the diagnoses recognized by the belief network. Adding a new diagnosis or retracting a previous diagnosis causes a reevaluation of the belief network and a reinterpretation of the patient observations. The user can access a patient simulator to explore the effect of adjusting the controls of the simulated ventilator: VentPlan will invoke the patient-specific model to predict the effects of the ventilator settings on the patient's physiology [Tovar, 1991].

3.5 Implementation

VPnet uses the Lauritzen–Spiegelhalter algorithm for network evaluation [Lauritzen, 1988]; each evaluation takes less than 2 seconds on the Macintosh IIci. The parameter-estimation and model-simulation procedures of the mathematical-modeling module are implemented with standard numerical techniques [Press, 1988]. The computation time required to complete one cycle of belief-network evaluation, parameter estimation, and search for optimal ventilator settings varies according to the number of patient-data observations and according to the goodness of fit of the mathematical model to the data. We limit the maximum duration of this cycle to approximately 2 minutes, by restricting the number of steps that can be taken during the parameter-estimation procedure. For the ICU-patient data that we have evaluated, the cycle duration has been less than 1 minute.

3.6 Evaluation

The evaluation of VentPlan requires several steps. The first step is to validate each of the system components. The second step is to evaluate the architecture for combining the components. The third step is to compare the overall program to alternative approaches for accomplishing the same task. We have so far performed initial validation of each of the VentPlan components, and have examined retrospectively the recommendations that the prototype made for postoperative patients in a surgical ICU.

3.6.1 Validation of the Belief Network

The belief network represents a probabilistic model of how pathophysiologic disease states (input nodes) influence physiologic parameters (intermediate nodes). There is no gold standard for interpreting the calculated distributions for the intermediate nodes of our network. Instead, we rely on subjective assessments of the calculated distributions, as determined by our clinical experts. A systematic evaluation of our belief network's performance is not feasible, since VPnet has over 400 million possible combinations of the states of diagnosis and report nodes. The approach that we took was to identify and evaluate smaller regions of the network that are sparsely connected to other regions. We also evaluated the network's performance on sets of data corresponding to probable clinical scenarios, and on sets of data from real clinical cases. We have presented the results of this evaluation elsewhere [Rutledge, 1990].

3.6.2 Validation of the Preference Model

The preference model provides a relative ranking of proposed ventilator settings. The model is a *valid representation* of the expert's preferences if, for any patient, the ventilator setting ranked highest by the model is also the ventilator setting selected as optimal by the expert from whom the model was elicited. To test this validity, then, we require a method to determine the expert's optimal setting for a given patient. Since we cannot arbitrarily adjust the ventilator while it is supporting a real patient, we instead ask the expert to interact with a model of the patient. The model dictates how the simulated patient would respond to the ventilator, and the expert adjusts the settings until she is satisfied that they are optimal. Note that, if we were to compare the setting ranked highest according to the model with that ranked highest by a different expert, or with that ranked highest by a panel of experts, then we would be evaluating the interexpert variability in preferences for setting the ventilator.

Farr presented an evaluation of the preference model [Farr, 1989]. He showed that the preference model was able to predict the ventilator settings that were considered optimal by different assessors, and that different experts preferred the settings that were optimal according to their own preference models over the settings that were predicted to be optimal by the models that were derived from other experts.

3.6.3 Validation of the Mathematical Model

We studied the ability of VentPlan's mathematical model to predict the effects of ventilator-setting changes that were made routinely during the care of postoperative patients in the surgical ICU. We examined retrospectively the records for 10 patients chosen randomly from those patients who were admitted to the Palo Alto Veterans Administration Medical Center (PAVAMC) surgical ICU over a 2-month period. The procedures that these patients underwent were coronary-artery bypass grafting (5), cardiac valve replacement (2), sternal rewiring (1), exploratory laparotomy (1), and subdural hematoma evacuation (1). One patient recovering from coronary-artery bypass grafting was transferred to the surgical ICU after suffering a cardiac arrest on the ward.

We analyzed data from the online ICU-data system from the time of admission until the ventilator rate was reduced below 4 breaths per minute (nine patients), or the patient expired (one patient). These records include minute-to-minute observations from the bedside monitors and from the ventilator. We constructed data files for input to VentPlan that included observations of O_2 saturation (pulse oximeter), heart rate (electrocardiogram monitor), mean arterial pressure (indwelling arterial catheter), central-venous and pulmonary-artery pressure (central-venous or Swan–Ganz catheter), and temperature (rectal probe). We manually added the observations of the abg analysis and of the cardiac output obtained by thermodilution (Swan–Ganz catheter). We supplied the mathematical model with a standard set of prior parameter distributions for all cases, and did not reevaluate these distributions based on inputs to the belief network. We isolated the mathematical model to test its ability to make predictions without benefit of the parameter distributions computed by the belief network. The VentPlan predictions had no effect on the frequency or magnitude of changes to the ventilator controls, since we analyzed the online data record retrospectively.

We compared the model predictions for P_aCO_2 and P_aO_2 with the abg measurements that were taken after each ventilator-setting change. We included an observation in the evaluation if the time from the ventilator-setting change to the second abg was at least 20 minutes, and if the time from the abg before the change to the abg after the change was less than 3 hours (34 instances).

The correlation coefficients for predictions of P_aO_2 and P_aCO_2 were 0.77 and 0.61, respectively. The average prediction errors for P_aO_2 and P_aCO_2 were 17.6 and 4.8 mm Hg, respectively, and the standard errors for P_aO_2 and P_aCO_2 were 23.5 and 6.3 mm Hg, respectively. We show a a graph of the predictions of the P_aO_2 versus the measured P_aO_2 in Figure 3.3.

3.6.4 Evaluation of VentPlan's Recommended Settings

We studied 10 patient-data records to compare, each time that the ICU team adjusted the ventilator, the recommendation made by VentPlan with the ventilator-setting change that was implemented by the ICU team. This set of patient-data records was distinct from the set that we used for the validation of the mathematical-model predictions. We found 55 ventilator adjustments during 355 hours of monitoring of these 10 patients. The ventilator changes were made to the FIO_2 in 42 cases and to the minute ventilation ($\dot{V}$)—either the

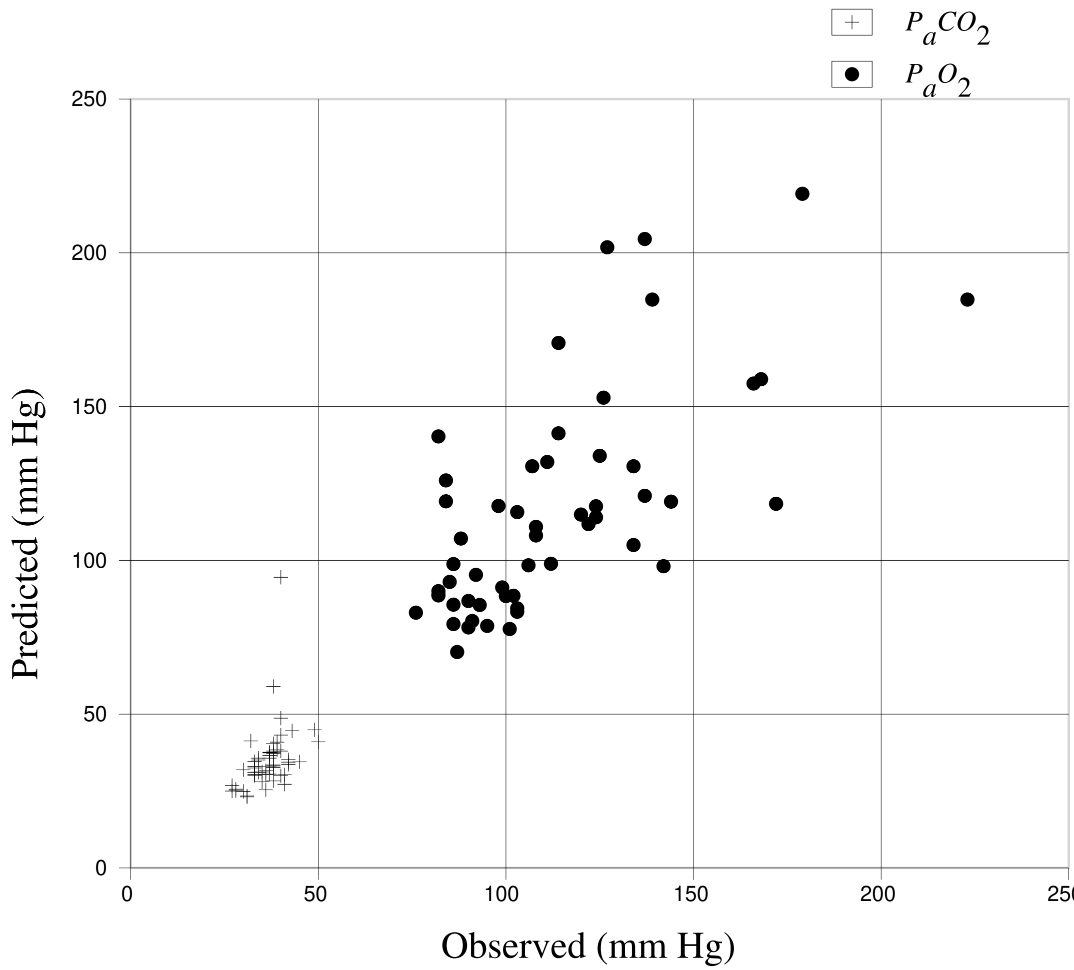

Figure 3.3 Model predictions versus patient observations. Predictions for P_aO_2 and P_aCO_2 after changes in the ventilator settings are plotted against the subsequent measurements of P_aO_2 and P_aCO_2. The correlation coefficients are for P_aO_2, 0.77, and for P_aCO_2, 0.61.

RR or the V_T—in 20 cases.[7] The average change in FIO_2 was 0.10, and the average change in $\dot{V}$ was 24 deciliters per minute. Figure 3.4 shows the distribution of the magnitude of the observed ventilator-setting changes. For all patients, the *PEEP* remained at 5 cms. of water at all times.

Of 55 adjustments to the FIO_2 that were observed, VentPlan disagreed with the direction of the change in only two cases.[8] Of 29 adjustments to the minute ventilation (either the V_T or the *RR*), VentPlan disagreed with the setting changes made in seven cases. In each case of

7. The number of setting changes is less than the total of the FIO_2 plus $\dot{V}$ changes, because both the FIO_2 and the $\dot{V}$ were adjusted at the same time in seven cases.

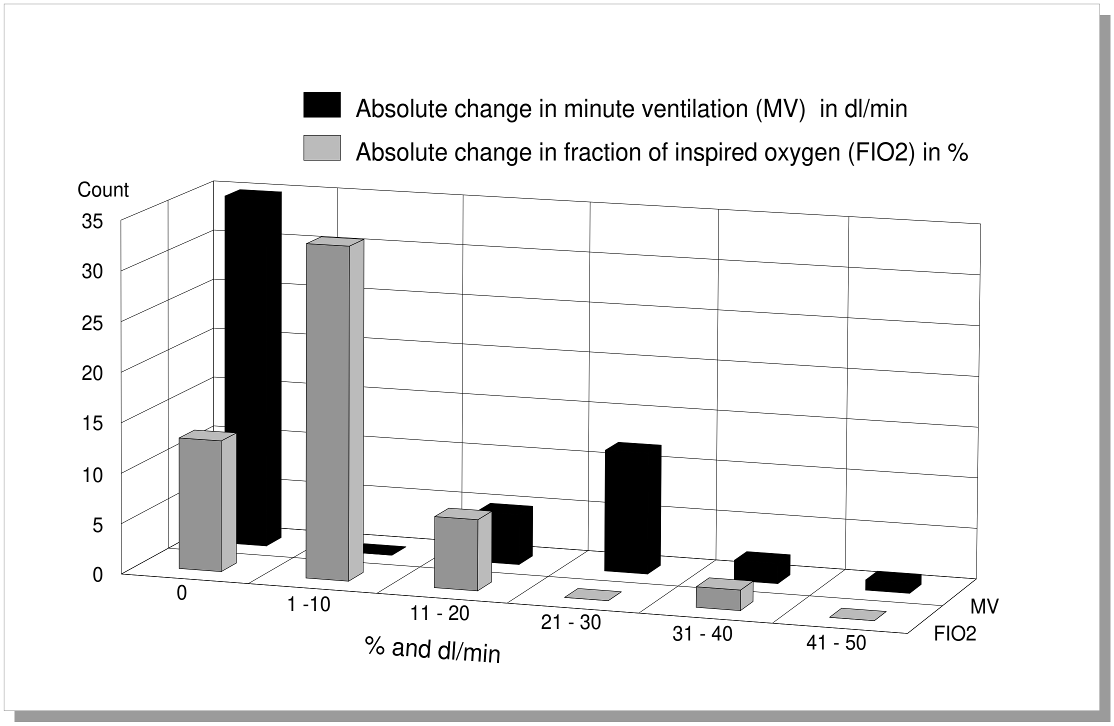

Figure 3.4 Distribution of magnitude of ventilator-setting changes. The observations of 55 ventilator-setting changes are shown.

disagreement, VentPlan's recommendation was to change the minute ventilation so as to normalize the P_aCO_2, whereas the actual change implemented induced an increased hypocapnea or hypercapnea. Figure 3.5 shows the relationship of actual changes in FIO_2 to the changes in FIO_2 recommended by VentPlan.

We have not yet performed a prospective study to evaluate the effects on patient care or patient outcomes of having Ventplan recommendations available to the ICU team during the ventilator-setting decision-making process.

3.7 Discussion

VentPlan extends design principles introduced in several system architectures. Like VM [Fagan, 1980; Fagan, 1984], VentPlan establishes a patient context that it updates

8. In one case, for a patient with an elevated P_aO_2 (492 mm Hg), the ventilator was increased from an FIO_2 of 0.7 to 0.8, whereas VentPlan recommended a decrease in the FIO_2 to 0.40. In the other case, for a patient with a P_aO_2 of 93, the ventilator was reduced from an FIO_2 of 0.40 to 0.35, whereas VentPlan recommended an increase in FIO_2 to 0.45.

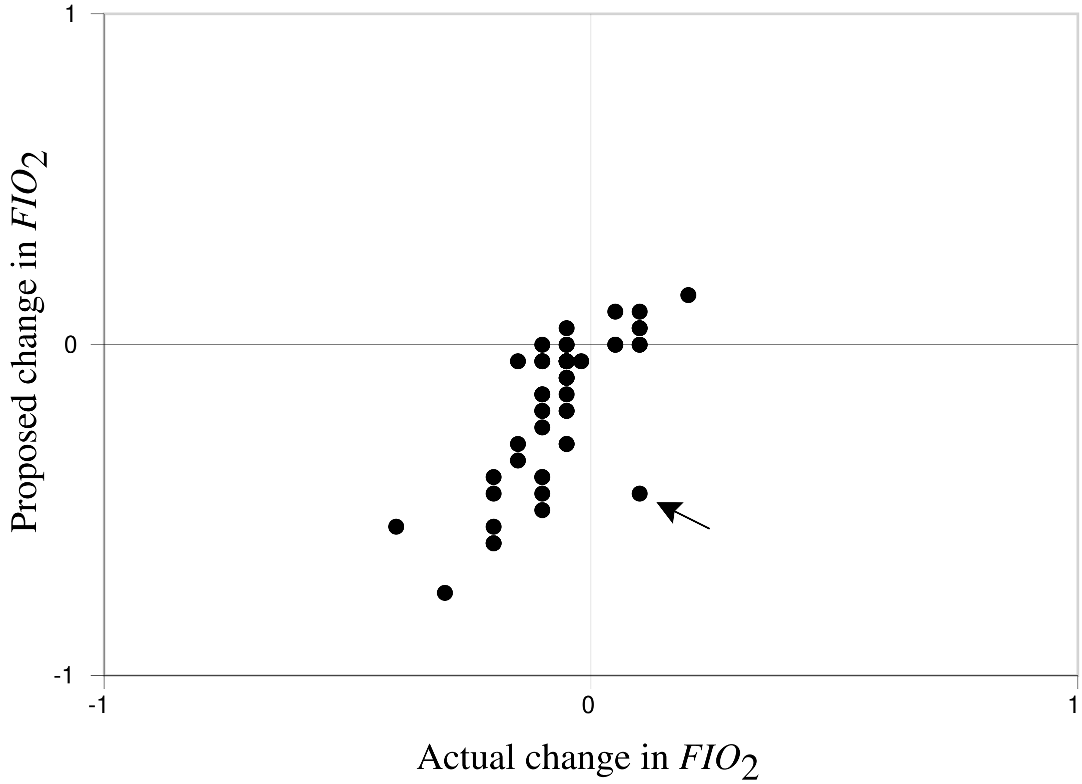

Figure 3.5 Proposed versus actual ventilator-setting changes. The recommendations for the change in FIO_2 calculated by VentPlan are plotted against the actual changes in FIO_2 implemented by the ICU team. The outlying point—indicated by an arrow—corresponds to a patient with an elevated P_aO_2 and an FIO_2 of 0.7, for whom VentPlan recommended a decrease in FIO_2, whereas the ICU team increased the FIO_2. Linear regression equation for these data is
Proposed change in FIO_2 = –0.08 + 1.48 Actual change in FIO_2. The correlation coefficient, r, is 0.73. For these data without the one outlying point, $r = 0.80$.

continually, using new data as they become available. Current findings are compared with explicit expectations generated from this patient context. Abnormal findings trigger an error analysis and, if they are not withdrawn, a change of context. A related combination of techniques was presented in the planning architecture of ONYX [Langlotz, 1985].

VentPlan exploits the advantages of both quantitative and qualitative techniques: Time and nonlinear relationships are handled by the numerical routines of the mathematical model, whereas qualitative data are processed by the probabilistic routines of the causal model. The mathematical model is, by itself, unable to interpret categorical information about the clinical context, and is unable to interpret data measurements for variables that are not included in it. The belief network solves this problem by converting the diagnosis, and other measured variables, into a form that the mathematical-modeling module can use directly.

We have presented the qualitative and quantitative components of our architecture as separate modules communicating through a probabilistic representation of shared model parameters. An equivalent perspective is to consider that the belief network and mathematical models are complementary views of the same set of physiological processes. According to this perspective, the two models are independent, but we can evaluate them sequentially to take advantage of the strengths of each (Figure 3.1).

3.7.1 Changes in Patient State Over Time

We assume that the physiologic parameters do not change over time. We do not represent or attempt to model changes in physiologic state that occur slowly, so VentPlan's recommended ventilator plan consists of a single recommendation for the setting of the ventilator. Until there is a change in the assessment of the physiologic state, the calculated optimal setting will remain the same. VentPlan does adjust its recommendation, however, as soon as a change is signaled by the monitored data or is suggested by a new diagnosis provided to the belief network. If the physiologic state changes slowly, the parameter estimates follow these changes, lagging behind the actual values.

3.7.2 Mathematical-Model Predictions

The model that we have implemented in VentPlan is highly simplified, yet provides reasonable prediction accuracy for postoperative ICU patients. The magnitude of the recommended changes in ventilator settings is greater than that of the changes that were implemented by the ICU team (see Figure 3.5).

For patients with more complex pathophysiologic abnormalities, we do not expect VentPlan's model to make accurate predictions. Nevertheless, the predictions of VentPlan's model may be sufficiently accurate for small changes in ventilator settings to allow VentPlan to make recommendations for smaller changes in settings.[9] The effect of adding

9. Rudowski showed that, given less powerful models and less accurate predictions, a VMA can arrive at the optimal settings after making a series of smaller ventilator-setting changes [Rudowski, 1990].

more accurate models to VentPlan would be to allow VentPlan to make recommendations for larger changes in ventilator settings, for patients with a wider variety of abnormalities.

3.7.3 Evaluation of VMA Recommendations

Evaluation of VMA ventilator-setting recommendations is difficult, since there is often disagreement among experts as to the optimal course of action for a given patient. As Figure 3.5 shows, we demonstrated that the changes in ventilator settings that VentPlan recommended correlated well with the changes that were actually implemented by the patient-care team.

3.7.4 Pragmatic Concerns

Quantitative observations in the ICU are subject to multiple sources of observation error. For example, pulse oximeter observations of arterial oxygen saturation may be falsely low [Gillies, 1993; Oniki, 1993], and charted ventilator settings may differ from the actual setting of the ventilator [Gardner, 1991; Young, 1995]. A practical VMA will require a more robust mechanism for assessing whether a change in an observation of a physiologic variable is explained by a change in the underlying physiology, or represents noise in the observation.

There are numerous obstacles to the successful implementation in the ICU of a VMA such as VentPlan. More complex mathematical models would be needed to explain the pathophysiologic abnormalities of certain ICU patients. To apply a more detailed model to a specific patient, we must specify values for the greater number of parameters of the more complex model, even when there are few observations that restrict the parameter values. As in VentPlan, a VMA could apply a belief network to provide prior parameter estimates to specify parameter values when quantitative observations are unavailable.

3.8 Summary

Quantitative physiologic models provide automated methods for the interpretation of ICU data with a powerful method to integrate multiple observations of ICU patients and provide the only method to make quantitative patient-specific predictions of the effects of alternative ventilator settings. A ventilator-management advisor (VMA) is a computer program that interprets observations of patients in the ICU and makes recommendations for adjustments to the settings of a mechanical ventilator, based on the predictions of a quantitative physiologic model.

VentPlan is a prototype VMA that interprets patient physiologic measurements, ventilator settings, and clinical diagnoses by incorporating these quantitative and qualitative data in a patient-specific physiologic model. A belief network converts clinical diagnoses to distributions on physiologic parameters, a mathematical-modeling module applies a patient-specific mathematical model of cardiopulmonary physiology to predict the effects of alternative ventilator settings, and a decision-theoretic plan evaluator ranks the predicted effects of alternative ventilator settings according to a multiattribute-value model that specifies physician preferences for ventilator treatments. VentPlan applies its preference model to the current settings and to the calculated optimal settings, and issues a recommendation for adjusting the ventilator.

Although the prototype demonstrates the feasibility of the VentPlan architecture, a VMA that is to be applied to the wide variety of ICU patients would have to incorporate physiologic models that are more detailed than is VentPlan's model. In Chapter 4, we shall discuss a more detailed physiologic model, called VentSim, that would be suitable for use in an advanced VMA, if its computation complexity were not limiting. In Chapters 5 and 6, we shall see how a dynamic-selection-of-models method allows a VMA to minimize the problem of increased computation complexity despite the severe time constraint for decision making.

Chapter 4

Mathematical Models of Cardiopulmonary Physiology

Automated methods that provide treatment advice require a model of physiology to assist them in the interpretation of ICU-patient data, and to guide them in the selection of treatments. VentPlan implements a highly simplified model to make recommendations for the ventilator setting for post-operative ICU patients, but VentPlan's model does not include enough detail to predict accurately the responses of many patients to changes in the ventilator settings. VentSim is an interactive ICU-patient simulator that incorporates a physiologic model to predict the effects of ventilator settings for critically ill patients in the ICU. The VentSim model is too complex computationally for use at the inner loop of a VMA; however, it provides a reference model against which we can compare the prediction accuracy of other, more simplified models.

In this chapter, I review research on the development of mathematical models of cardiopulmonary physiology (Section 4.1), then review the VentPlan model (Section 4.2), and describe the structure and implementation of the VentSim model (Section 4.3).

4.1 Mathematical Models of Physiology

Mathematical models are powerful tools for simulating the quantitative time-dependent behavior of complex, dynamic systems, such as the human cardiopulmonary system [Cellier, 1991]. One of the first mathematical models of cardiopulmonary physiology (the physiology of the respiratory and the circulatory systems) was the three-compartment model of Riley and Cournand [Riley, 1949]. The conceptual simplicity of this model, and its explanatory power, make it a classic model that educators continue to teach to students of physiology. Before the advent of digital computers, there was no method to implement and test complex mathematical models other than symbolic analysis or graphical analysis. Mechanical and electrical analog computer implementations led to qualitative descriptions of model behavior, but were limited by their inability to provide quantitative predictions [Snyder, 1969; Hill, 1973].

With the advent of digital computing devices, numeric simulation of more complex models became possible. As the capabilities of digital computing devices increased, larger and more detailed research models of physiology were developed [Guyton, 1984]. An ever-increasing number of researchers have worked to develop new and more detailed models of physiologic systems. For example, the size of the research community that is dedicated to the subject of physiologic modeling of the respiratory system is reflected by the rate of publication of research articles. Figure 4.1 presents observations of the rate of addition of articles to the MEDLINE database, from 1966, when the National Library of Medicine began indexing articles in biomedicine, to 1993. During this period, the number of research articles on the subject of simulation of the respiratory system increased exponentially, whereas the rate of addition of articles on all subjects increased only linearly. The data show that the rate of addition of articles on this subject has an exponential doubling time of approximately 5.0 years.

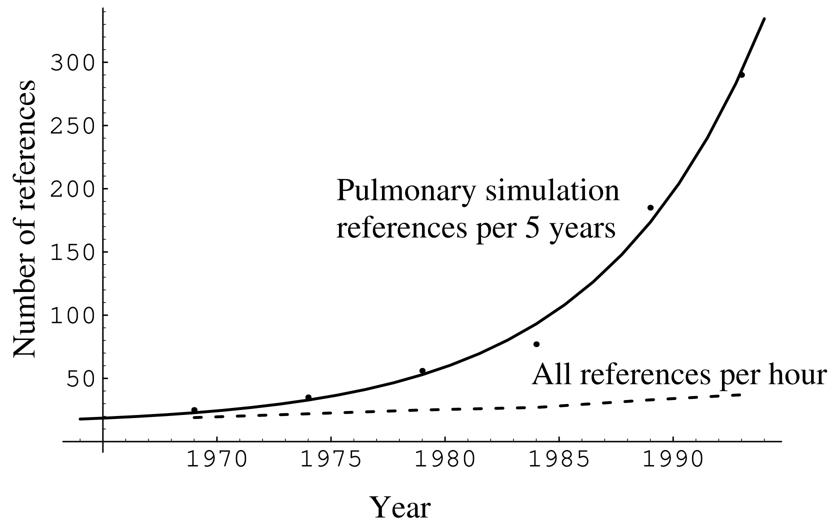

Figure 4.1 Rate of addition of new references to MEDLINE. The average rates of addition to the MEDLINE database of references on the subjects of simulation of the respiratory system are plotted as points on the solid line. The solid line is the exponential function $b + e^{(Year - 1964)/kt}$, in which b=2.563 and kt=7.205, which corresponds to a doubling time of 4.99 years. The rate of addition of all articles to the MEDLINE database is plotted as a dashed line; note that the scales for the two plots differ by a factor of 43,800 (total articles per hour versus articles per 5 years). Simulation-related MEDLINE articles were counted with search statements that found all articles with a major subject-heading match to the term SIMULATION, and with a subject heading match to one of the following terms: LUNG, RESPIRAT#, PULMON#. Each point represents an average of data for the preceding 5 years (for the points plotted at 1974, 1979, 1984, and 1989) or for the preceding 4 years (for the points plotted at 1969 and 1993).

Researchers are now so confident of the faithfulness of their models that they look for explanations to unexpected clinical observations through model simulations. For example, physiologists had thought that no one could climb to the top of Mount Everest without supplemental oxygen. When two climbers reached the summit without supplemental oxygen in 1978, West applied a mathematical model to demonstrate how that feat was possible [West, 1980].

Researcher have studied most of the physiologic mechanisms that can be studied directly. Physiologists now turn to models to help elucidate the interactions among mechanisms that are not amenable to direct observation. To quote Guyton, “probably the most important role that can be played by the use of complex mathematical models is to analyze and to understand mechanisms that are beyond the attack of presently available direct experimental methods” [Guyton, 1984].

Physiologic models fit into one of two classes: (1) comprehensive models that simulate interactions and regulatory mechanisms among a set of physiologic organ systems, and (2) models that explore, in detail, a restricted set of physiologic concepts.

4.1.1 Comprehensive Models

Comprehensive models include a wide coverage of physiologic concepts and feedback mechanisms from two or more organ systems. An important use of these models is to study interactions among physiologic systems [Sun, 1992; Barnea, 1993]. An early and influential project in this area was Guyton's study of the behavior of a comprehensive model of renal and cardiovascular physiology. His work led to the understanding that long-term blood-pressure control is mediated almost entirely by the kidneys [Guyton, 1983; Guyton, 1991].

Physiologists use comprehensive models to teach physiologic concepts to students of medicine and physiology. Since the advent of inexpensive desktop computers in the 1980s, many teaching models have been developed; examples include models in the areas of cardiovascular and respiratory physiology [Dickinson, 1977; Lefevre, 1988], of anesthesia treatments [Heffels, 1990; Gaba, 1988; Schwid, 1987], and of ventilator management [Petrini, 1986; Boyle, 1991; Gaar, 1991; Rutledge, 1994b].

HUMAN is a remarkably complete microcomputer-based model that allows students to perform a variety of physiologic experiments without the concern for the risks and expense of animal or human experimentation. In contrast to the research-model goal of absolute accuracy and faithful model response, teaching models require a less accurate response. The authors of HUMAN state that, in certain areas, they sacrifice model fidelity for robustness, at a cost of the loss of faithful transient responses. They suggest that the strong homeostatic mechanisms in the biologic system that they model lead to the appearance of reasonable behavior in the simulations [Coleman, 1983].

Comprehensive models of physiology contain far more parameters than could be measured in any single individual. Model builders attempt to find sets of parameters that cause the model to generate realistic or reasonable behavior for a characteristic patient or set of patients. The models are tolerant of variations in individual parameter values because they represent strong homeostatic mechanisms that the models represent. In effect, the models mimic the tolerance to variability that biologic systems demonstrate. An example of this biologic tolerance to variability is the fact that an 80-year-old man may have one-half of the cardiac output of a 20-year-old man, but not be in heart failure [Guyton, 1984].

4.1.2 Detailed Models

The second class of models explores a restricted area of physiology in detail to study a single physiologic mechanism. For example, detailed models of the human airways led researchers to insights regarding distribution of airway resistance in normal and diseased lungs [Weibel, 1989; Wiggs, 1990], and to a more complete understanding of gas exchange in the respiratory system [Tomlinson, 1993]. Researchers have answered a variety of questions about the human cardiovascular system by building detailed models of specific aspects of cardiovascular physiology.[1]

4.1.2.1 Detailed Models of Respiratory Physiology

Weibel described a reference model for the branching of the airways from the bronchus to the terminal bronchioles. This structural lung model predicted the distribution of pulmonary ventilation by simulating the behavior of a hierarchical branching structure of the airways. Weibel estimated the airway-model parameters by studying postmortem human lung specimens. He derived values for the airway-model parameters that are typical for human lungs, then answered questions of the form, “If the airways were structured as defined by these parameter values, what would be the distribution of airflow?”

1. Other examples of detailed models of specific aspects of cardiovascular physiology include models of diastolic cardiac dynamics [Summers, 1992], counterpulsation [Bai, 1992], arterial-graft effect on cardiovascular function [Helal, 1994], coronary-sinus–blood flow [Schreiner, 1992], increased gravity [Moore, 1992] and microgravity [Li, 1993].

Wagner described a second example of a detailed research model of the distribution of ventilation ($\dot{V}_A$) and perfusion ($\dot{Q}$) in the lungs [Wagner, 1974]. This model, which addressed a narrow area of physiology, enabled him to explain the causes of abnormal gas exchange in diseases such as asthma, emphysema, hepatic cirrhosis, and others [Wagner, 1987; Dantzker, 1987; Lee, 1987; Ringsted, 1989; Roca, 1988; Roisin, 1990; Torres, 1989; Ballester, 1990; Castaing, 1989].

The method that Wagner implemented—called the multiple inert-gas–elimination technique (MIGET)—is based on the fact that inert gases are eliminated from the lungs at rates that vary according to two factors: (1) the gas solubility in blood, and (2) the ratio of ventilation-to-perfusion in each region of the lungs [Wagner, 1974]. Wagner injected patients intravenously with inert gases that vary in their solubilities, then measured the rates of elimination of each gas in the expired air. He derived directly, from the MIGET data, values for the ventilation to perfusion ratios ($\dot{V}_A / \dot{Q}$) for 50 discrete compartments in his model of the lungs. He found a discrete distribution of $\dot{V}_A / \dot{Q}$ that caused the model to reproduce the MIGET data, although many such distributions existed for any set of data.

The parameters of Wagner's 50-compartment model are dramatically underconstrained by the MIGET data, because the model has 49 fitted parameters, and only seven observations. That is, an infinite number of parameter combinations within a 42-dimensional space would explain these data. Wagner implemented conventional least-squares estimation with enforced smoothing to compute *representative* patient-specific parameter estimates. He and his colleagues demonstrated that the fitting technique computed a $\dot{V}_A / \dot{Q}$ distribution that had similar structure to the average distribution among all possible solutions [Kapitan, 1987].

Lim demonstrated that a statistical model with six lung compartments offered a complete description of MIGET data, but he left unanswered the question of whether a smaller number of compartments would also explain the inert-gas data [Lim, 1990]. Kaufman, Lee, and Patterson demonstrated that, often, thy could reproduce the MIGET data by using a three-compartment model that included a shunt compartment plus two lung compartments (this structure is similar to that of the VentPlan model). In all cases, they reproduced the data by using a four-compartment model that included shunt plus three lung compartments. These authors demonstrated that the many-compartment models predicted the same

MIGET data for unimodal, bimodal, and trimodal distributions of $\dot{V}_A / \dot{Q}$. For a single set of MIGET data, multiple qualitatively different distributions of $\dot{V}_A / \dot{Q}$ for Wagner's 50-compartment model were possible, because the parameters of Wagner's model were underconstrained. Lee and colleagues concluded that a four-compartment model offers the advantage of a greatly reduced number of parameters whose values are determined uniquely by available data [Kaufman, 1987].

4.1.2.2 Application of Ventilation–Perfusion Models in Clinical Practice

Physicians do not routinely perform MIGET studies on patients in the ICU, so measurement of $\dot{V}_A / \dot{Q}$ distributions are not readily available for patients who receive ventilator treatment. Measurements of the partial pressure of arterial oxygen and carbon dioxide (P_aO_2 and P_aCO_2) are available as a matter of routine for ICU patients, and are usually obtained after a change in the ventilator settings. Lee and colleagues demonstrated that the observations of P_aO_2 alone at varying FIO_2 are sufficient to determine the values for the parameters of the four-compartment model [Lee, 1987].

Research on models of ventilation and perfusion suggest that an automated method could assess the type of gas-exchange abnormality in a patient by fitting the set of observations of P_aO_2 at varying FIO_2 to the four-compartment model, to compute a discrete $\dot{V}_A / \dot{Q}$ distribution that explains the observations. This distribution (a shunt fraction, and blood-flow fractions to a low $\dot{V}_A / \dot{Q}$ region and a normal $\dot{V}_A / \dot{Q}$ region) provides a quantitative estimate of the magnitude of the $\dot{V}_A / \dot{Q}$ abnormality, and allows a the four-compartment model to predict the response of the patient to other levels of FIO_2.

4.1.3 Models for a VMA

The architecture for a VMA, as I described in Chapter 3, requires that the VMA fit the parameters of a physiologic model to observations from an ICU patient, then simulate the effects of alternative ventilator settings during a search for optimal settings. This cycle must occur within 1 to 2 minutes in the real-time setting of ICU-patient care.

A major impediment to the application of more detailed—and more realistic—physiologic models in a VMA is the computation-resource limitation imposed by the real-time nature of patient-management decisions in the ICU. Highly detailed cardiopulmonary models are too computation intensive for a VMA. For example, Weibel's model of the airways required a lengthy computation to simulate the effects of a single set of model parameters. Another difficulty with highly detailed models is that they include parameters that cannot be measured, or that are not typically observed in ICU patients.

A cardiopulmonary model suitable for a VMA must balance the need to include physiologic interactions that affect observable variables against the computation-resource limitations that are imposed by the time-critical need for treatment recommendations. If more complex models are needed to represent a specific patient's physiologic abnormality accurately, then whether that more accurate model will be useful depends on the degree of inaccuracy of the less detailed model, and on the computation-time delay that the more detailed model would impose on a treatment recommendation.

Fortunately, to build a VMA, we do not need to implement a model that includes all known aspects of physiology. The experiments of the VentPlan project demonstrated that a VMA can apply a relatively simple model to interpret observations of ICU patients. An alternative approach to controlling the settings of the ventilator demonstrates that a VMA can be constructed with models that are even less accurate than is the VentPlan model. In a program called KUSIVAR, Rudowski and colleagues implemented linear-regression models to predict the changes in oxygen and carbon-dioxide concentrations that occur after adjustments in ventilator settings [Rudowski, 1989; Rudowski, 1990]. The authors implemented these models within a rule-based expert system that interpreted the physiologic context of the patient observations. An evaluation of the ability of these linear regression models to predict the P_aO_2 and P_aCO_2 revealed a weak correlation of the model predictions to the observations. The regression models were unable to find any correlations for six of 20 patients studied, and the overall correlation coefficient, r, was 0.28. To apply these models, Rudowski and colleagues assumed that there would be feedback observations after each small change in settings, so KUSIVAR would track continuously the observed physiologic behavior.

The reason for the failure of statistical models seems clear: If a model contains no information about the structure of the system, the data must specify the structure of the system. In the ICU domain, the data are often incomplete and noisy, so they are not sufficient to define an accurate prediction model. A similar problem plagues recent work to define a dynamic state–space model from ICU-data alone [Dagum, 1994].

In the ICU domain, we should include knowledge of cardiopulmonary physiology in a prediction model to improve the model's accuracy. VentPlan represents an initial attempt to incorporate this knowledge in a physiologic model. VentSim adds, to the VentPlan model, a greater level of detail of physiologic knowledge.

4.2 VentPlan: A Simplified Physiologic Model

The VentPlan model is a three-compartment model of physiology that describes the transportation of oxygen and carbon dioxide through the body, as described in Chapter 3. The VentPlan model represents the flow of oxygen and carbon dioxide in a simplified circulation, with a single compartment in which gas–blood interactions occur. The model makes the rather extreme simplifying assumption that the airways are normal, so all imbalances in $\dot{V}_A / \dot{Q}$ distribution are explained with single compartments for gas exchange, deadspace, and shunt.

VentPlan recalibrates its model for each new set of patient observations; therefore, the model always predicts the observed patient state, and makes highly confident predictions for the effects of small changes in control settings. For patients who have abnormalities that the model cannot represent, the prediction accuracy falls as the magnitude of the changes in ventilator settings increases.VentPlan provides ventilator-management decision support by interpreting a relatively simple patient model. A VMA would be more powerful if it were able to make more accurate predictions of the effects of changes in ventilator settings.

4.3 VentSim: An Expanded Physiologic Model

VentSim is a detailed model of cardiopulmonary physiology that simulates the effect of changing the control setting of a mechanical ventilator for ICU patients. Like VentPlan, VentSim is a continuous-time, continuous-state model that consists of a set of linked first-order differential equations that describes the circulation of oxygen and carbon dioxide through compartments of the body.

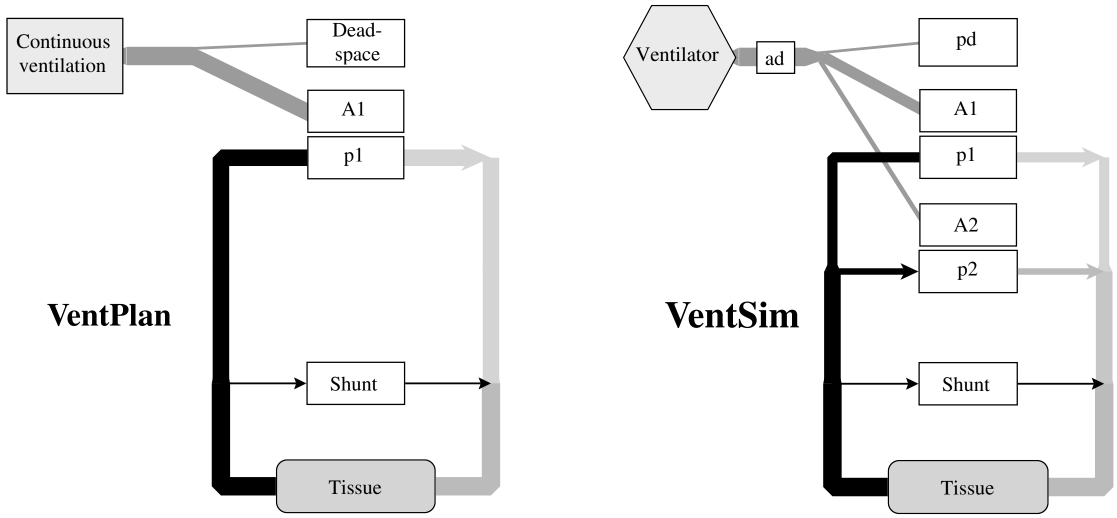

Figure 4.2 Comparison of VentPlan and VentSim models. Blood carries oxygen and carbon dioxide in a circuit, as shown by arrows. VentPlan computes the total alveolar ventilation from the ventilator settings, then divides the ventilation between the deadspace and a single alveolar compartment (A1) that exchanges gas with a single pulmonary blood-flow compartment (p1). VentSim contains a simulation model of a volume-controlled positive pressure ventilator that interacts through a series anatomic deadspace (ad), with three ventilated alveolar compartments: a nonperfused region (physiologic deadspace, pd), and two perfused alveolar compartments (A1, A2). The distribution of ventilation among the pd, A1, and A2 compartments depends on the resistance and compliance of each compartment, and varies with the frequency of ventilation. Each ventilated alveolar compartment (A1, A2) exchanges gases with a corresponding perfusion compartment (p1, p2).

VentSim incorporates more detailed versions of the ventilator, airways, and circulation components of the VentPlan model (see Figure 4.2). As a result of the greater detail in the VentSim model, VentSim predicts the effects of a wider variety of physiologic abnormalities than does VentPlan. In this section, I describe each of these components in detail. In Chapter 6, I show how models that incorporate components from both VentPlan and from VentSim are intermediate in their ability to explain diverse physiologic abnormalities, and also intermediate in their computation complexity.

4.3.1 Ventilator Component

VentPlan computes the effect of ventilation directly from the ventilator settings, and contains no explicit representation of a ventilator. VentPlan computes the continuous alveolar ventilation as the ventilator rate times the set tidal volume reduced by the total deadspace volume [$\dot{V} = RR \times (V_{\mathrm{Tset}} - V_{\mathrm{ds}})$]. By contrast, the ventilator component in

VentSim explicitly simulates a volume-cycled, constant-flow ventilator. The mechanical analog of the ventilator simulation is a rigid bellows with adjustable movement of a plunger. In VentSim's default configuration, the plunger moves at constant velocity, and compresses the desired tidal volume during the first part of the inspiration cycle. The simulator leaves a short inspiratory hold time after the plunger stops, to allow remaining bellows pressure to equilibrate with the airways component.

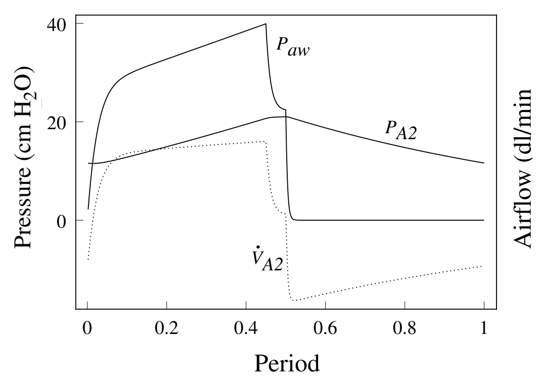

Figure 4.3 Sample ventilator simulation. The graph shows the ventilator pressure at the mouth (P_{aw}), the pressure in one alveolar lung compartment (P_{A2}) and the airflow into one alveolar compartment ($\dot{V}_{A2}$) during one cycle of ventilation.

During expiration, the ventilator pressure drops to the value set for positive end-expiratory pressure (PEEP), and outflow of air from the patient is limited by a variable outflow resistance (retard setting). Sample pressures and airflows during one cycle of ventilation of a simulated patient are shown in Figure 4.3.

Adjustable parameters of the VentSim ventilator allow it to simulate many volume-cycled constant-flow ventilators.[2] The default configuration for VentSim is for the constant mandated volume (CMV) mode of simulation. Differential-equation modeling makes it straightforward to adapt the VentSim ventilator component to simulate other mechanical ventilators for which a complete description is available.

2. The ventilator-specific parameters include maximum positive pressure, inspiratory hold time, and expiratory retard.

4.3.2 Airway Component

VentSim's airway component has four compartments: a series anatomic deadspace, a parallel physiologic deadspace, and two alveolar compartments (see Figure 4.2). Each compartment has an associated airway resistance and a lung compliance. The airway component interacts with the ventilator component to predict the pressures, airflows, and volumes of ventilation at each point in the ventilator cycle. VentSim computes the tidal volumes for each airway compartment during the simulation, and, when all tidal volumes are unchanged during successive ventilator cycles, VentSim detects that the simulator has reached a cycling steady state.

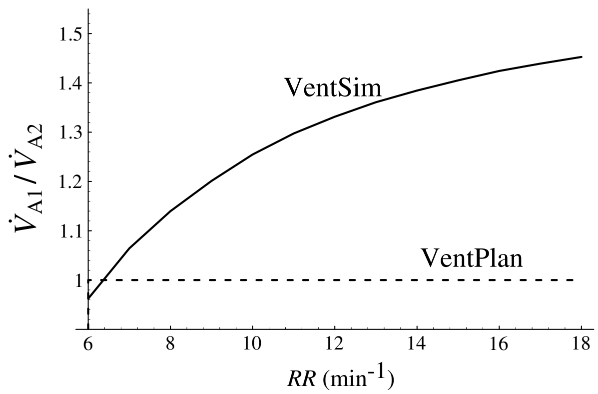

Figure 4.4 Effect of asymmetric resistance-compliance (RC) relationship on distribution of ventilation. The continuous line shows the ratio of ventilation in two alveolar compartments for a simulated patient with asymmetric RC values. The dashed line shows the constant, symmetric ventilation of the single alveolar compartment of the VentPlan model. V_{A1}, V_{A2}, ventilation of the A1 and A2 alveolar compartments shown in Figure 4.2; *RR*, frequency of ventilation.

If the two alveolar compartments have different resistance or compliance values, then the distribution of ventilation is asymmetric. If the products of resistance and compliance (the RC time constants[3]) of the two compartments differ, then the distribution of ventilation varies as a function of frequency of ventilation.

3. The resonant frequency of a compartment is defined by the compartment's resistance and compliance. The RC time constant is the inverse of the resonant frequency (the period). A driving force at a frequency greater than the resonant frequency will be cause less cyclic flow than a similar force at the resonant frequency.

Figure 4.4 shows an example of the effect of frequency on the distribution of ventilation in a simulated patient who has regions of the lungs with asymmetric RC characteristics. As the frequency of ventilation changes from 6 to 16 per minute, the ratio of ventilation in the two alveolar compartments changes from 1 to 1.4. This effect can be important in explaining the response of certain patients to changes in ventilation frequency [Weibel,1989].

4.3.3 Circulation Component

The circulation component of VentSim has two perfusion compartments (p1, p2) that correspond to the two ventilation compartments (A1, A2), in addition to a shunt and a tissue compartment (see Figure 4.2). The presence of a second perfused compartment that participates in gas exchange allows VentSim to represent asymmetric ventilation–perfusion distributions ($\dot{V}_A/\dot{Q}$).

The ability to represent asymmetric $\dot{V}_A/\dot{Q}$ is essential if a model is to describe accurately the effect of changes in inspired oxygen on the oxygen saturation. For example, for a simulated patient with severe asthma, a three-compartment model (such as VentPlan's) underestimates the fall in oxygen saturation as the fraction of inspired oxygen is reduced (see Figure 4.5).

There is a ventilation–perfusion ratio for each of the approximately 3×10^8 alveoli in the lungs. Taken together, the ventilation–perfusion ratios form a nearly continuous distribution for $\dot{V}_A/\dot{Q}$. The VentSim model provides a first-order approximation to asymmetric $\dot{V}_A/\dot{Q}$ by representing the distribution as $\{\dot{V}_{A1}/\dot{Q}_{p1}, \dot{V}_{A2}/\dot{Q}_{p2}\}$.

4.3.4 Oxygen-Content Relationship

The relationship of partial pressure of oxygen (P_{O2}) to oxygen concentration (C_{O2}) is nonlinear due to the presence of hemoglobin (Hb) in the blood. Hb binds oxygen avidly, so the content of oxygen in the blood is determined by the amount of Hb in the blood. Each gram of Hb can bind up to 1.34 ml of O_2 at standard temperature and pressure[4] (STP). The amount of oxygen bound to Hb is modeled in VentSim with the Hill equation [Hill, 1973].

4. Standard temperature is 20 degrees Celsius; standard pressure is 760 mm Hg.

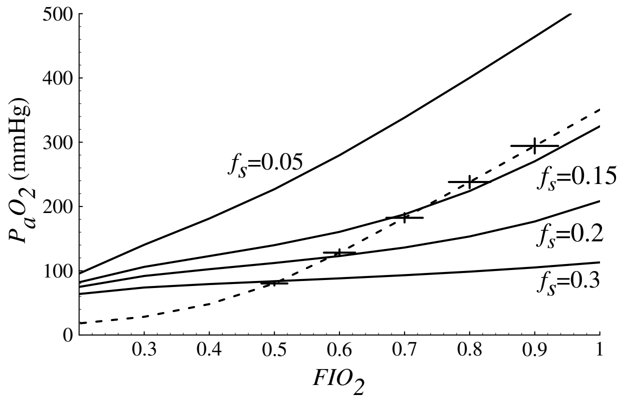

Figure 4.5 Effect of additional circulation compartments. The dashed line shows the VentSim model with parameters set to simulate a patient with a ventilation–perfusion mismatch (a moderate asymmetry in distribution of ventilation to the A1, A2 compartments, and of perfusion to the p1, p2 compartments shown in Figure 4.2). The continuous lines show the VentPlan model as the shunt fraction, f_s, varies. Sampled data are shown as crosses on the dashed line. No value of f_s allows the VentPlan model to predict the observations for P_aO_2 at all values of FIO_2.

In addition to the oxygen bound to Hb, there is an amount of oxygen in solution that increases linearly as the partial pressure of oxygen increases.

VentSim models the relationship as total O_2 *content* = *Hill* O_2 *content* + *dissolved* O_2, in which

$$Hill\,O_2\,content = \frac{\left(\frac{PO_2}{P_{50}}\right)^n}{\left(1+\frac{PO_2}{P_{50}}\right)^n}\text{,and} \tag{4.1}$$

$$Dissolved\,O_2 = Hb \times O_2\,per\,Hb + PO_2 \times O_2\,per\,Torr\,. \tag{4.2}$$

That is,

$$O_2\,content = \frac{\left(\frac{PO_2}{P_{50}}\right)^n}{\left(1+\frac{PO_2}{P_{50}}\right)^n} \times Hb \times O_2\,per\,Hb + PO_2 \times O_2\,per\,Torr\,, \tag{4.3}$$

in which P_{50} is the partial pressure at which 50 percent of the Hb capacity is bound to oxygen, *n* is the Hill constant (typically, 2.54), *Hb* is the concentration of Hb in the blood (in gm/dl), O_2 *per Hb* is the capacity of hemoglobin to bind oxygen (1.34 ml O_2 per gm Hb, and O_2 *per Torr* is the amount of oxygen dissolved in blood per unit of partial pressure of oxygen (0.003 ml O_2 per mm Hg). The relationship of O_2 *content* to PO_2 is shown graphically in Figure 4.6.

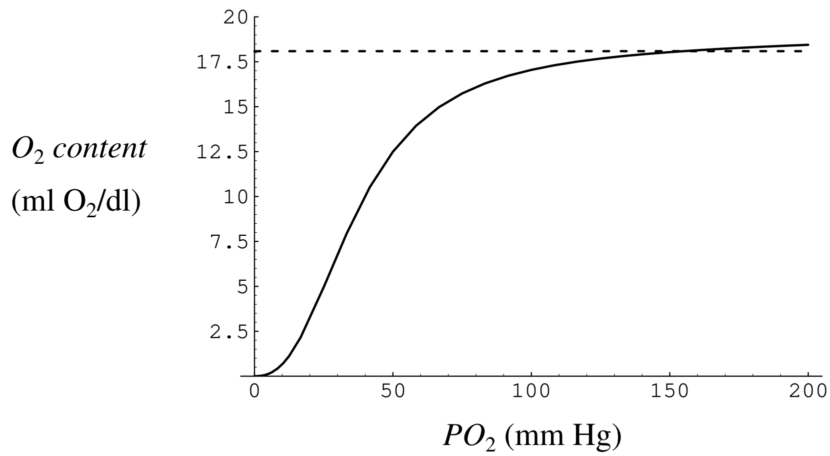

Figure 4.6 Oxygen-content relationship. The oxygen content, in ml O_2 per dl of blood is plotted against the partial pressure of oxygen (in mm Hg). Blood Hb is set to 13.5 gm/dl. P_{50} = 37 mm Hg; n = 2.54. The maximum amount of oxygen that can be carried by hemoglobin is shown as a dashed line (13.5 gm Hb multiplied by 1.34 ml/gm = 18.09 ml O_2). At oxygen tensions above 155 mm Hg, the oxygen content exceeds the total oxygen-carrying capacity of hemoglobin, due to the increasing content of dissolved oxygen at higher partial pressures of oxygen.

VentSim models patient-to-patient variations in this relationship with different values of P_{50} and *n.* VentSim does not include explicit models of several factors that influence the oxygen-content relationship (such as temperature, pH, 2-3-DPG concentration, and carbon-dioxide concentration).

To solve the oxygen-content relationship for a PO_2 value that corresponds to a given CO_2 value, we invert Equation 4.3. VentSim inverts this relationship numerically by applying the Brent method[5] to search for the value of PO_2 that satisfies Equation 4.3 for a given CO_2.

5. The *Brent method* (more fully, the *Van Wijngaarden–Dekker–Brent method*) combines root bracketing, bisection, and inverse quadratic interpolation; it is guaranteed to converge if a root exists within the interval, and if the function can be evaluated within the interval [Press, 1988]

4.3.5 VentSim Implementation

The VentSim implementation is a C program that integrates the difference equations that correspond to the differential equations. The full VentSim model has a total of 143 variables. The key model parameters, control variables, and prediction variables are shown in Table 4.1.

A graphical interface[6] allows users to study the behavior of the model by observing the effects of adjusting the ventilator under various assumptions about the physiology of a patient (Figure 4.7). The user inspects and adjusts model parameters and ventilator settings, and then observes the time-varying model predictions. The interface also allows a user to enter patient-specific model parameters, or to select a diagnosis from a list to set the model parameters to values that are typical for a diagnosis.

Table 4.1 VentSim and VentPlan variables.[a]

Model	Model parameters	Prediction variables	Control variables
VentPlan	$\dot{V}_{O_2}$, RQ, Q_T , f_s , V_{ds} HCO_2 , Hb	P_aO_2 , P_aCO_2 , pH_a P_vO_2 , P_vCO_2 , pH_v , $\dot{V}_{A1}$	V_{Tset} , RR , FIO_2 , $PEEP$
VentSim	$\dot{V}_{O_2}$, RQ, Q_T , f_s , f_{p1} , V_{ad} , V_{pd} , R_{ad} , R_{pd} , R_{A1} , R_{A2} , C_{A1} C_{A2} , C_{pd} , HCO_2 , Hb	P_aO_2 , P_aCO_2 , pH_a , P_vO_2 P_v CO_2 , pH_v , P_{aw} , V_{tidal} $\dot{V}_{A1}$, $\dot{V}_{A2}$, Q_{p1} , Q_{p2} , Q_s	V_{Tset} , RR , FIO_2 , $PEEP$ P_{max} , $IEratio$

a. Variables: V, volume; $\dot{V}$, dV/dt; P, pressure; R, resistance; C, compliance; Q, blood flow; f: fraction; $\dot{V}_{A1}$, alveolar compartment ventilation; $\dot{V}_{O_2}$, metabolic rate; RQ, respiratory quotient; Q_T, cardiac output; f_s, shunt fraction; HCO_2, serum bicarbonate; Hb, hemoglobin concentration; V_{tidal}, delivered tidal volume; V_{Tset}, set tidal volume; RR, set rate of ventilation; FIO_2, set fraction of inspired oxygen; $PEEP$, set positive end-expiratory pressure; P_{max}, set maximum positive pressure; $IEratio$, set inspiratory–expiratory ratio. Subscripts: s, shunt; a, arterial; v, mixed venous; ds, total deadspace;, ad, anatomic deadspace; pd, physiologic deadspace; aw, airway; A1 and A2, ventilated alveolar compartments; p1 and p2, perfused pulmonary compartments.

4.3.5.1 VentSim Solution Method

The VentSim model has no steady-state solution, because the ventilator component has a periodic, nonlinear driving function. As a result of this cyclic influence of the ventilator

6. I created the graphical interface for VentSim in NeXTStep, using Interface Builder.

component, VentSim reaches a steady state that is periodic, and there is no single steady-state solution.To find the model solution, we must integrate the model equations numerically. Unfortunately, the VentSim model equations are stiff— an airway component with extremely short time constants (due to low resistance and low compliance of the anatomic deadspace) interacts with a circulation component that has much longer time constants. As a result, numeric integration of the full model requires that we iterate small step sizes for long time periods; evaluation of the VentSim model is computation intensive.

VentSim computes approximate solutions by separating the slow and fast time-constant components of the VentSim model. That is, VentSim performs a breath-to-breath simulation of the ventilator–airway interaction by integrating the airway and ventilator components numerically, using a time step that matches the RC time constant of the airways,[7] until these components achieve a cycling steady state. At the cycling steady state, the observed tidal volumes allow VentSim to calculate an equivalent continuous ventilation. VentSim then makes the assumption that the alveolar ventilation is continuous and unchanging, to integrate the circulatory component with a longer time step,[8] until the circulatory component reaches a steady state. This technique is an approximation similar to the method described by Widman [Widman, 1989].

The VentSim implementation allows users to observe the time course of change in oxygen and carbon-dioxide concentrations. If we wish to compute only the steady state of the model, then direct solution of the equations that describe the steady state is a much faster method. To compute the steady-state solution of VentSim, we find the ventilator component's cycling steady state, derive the equivalent continuous ventilation for each alveolar compartment, then solve for the steady state of the circulation component by searching for the roots of the equilibrium equations.[9] Because of the nonlinearity in the relationship of oxygen pressure to concentration, VentSim implements numeric root finding to search for the roots of the equilibrium equations, using the Brent method.[10]

---

7. The time step that VentSim applies for numeric integration of the airway and ventilator components is 1 msec.
8. The time step that VentSim applies for numeric integration of the circulatory component is 200 msec.
9. At steady state, for each gas (oxygen and carbon dioxide), the flows into and out of each pulmonary-circulation compartment must be equal, and the sum of the flows must be equal to an amount that is set by the metabolic rate and the respiratory quotient.
10. VentSim also applies the Brent method to solve the inverse of the oxygen-content relationship.

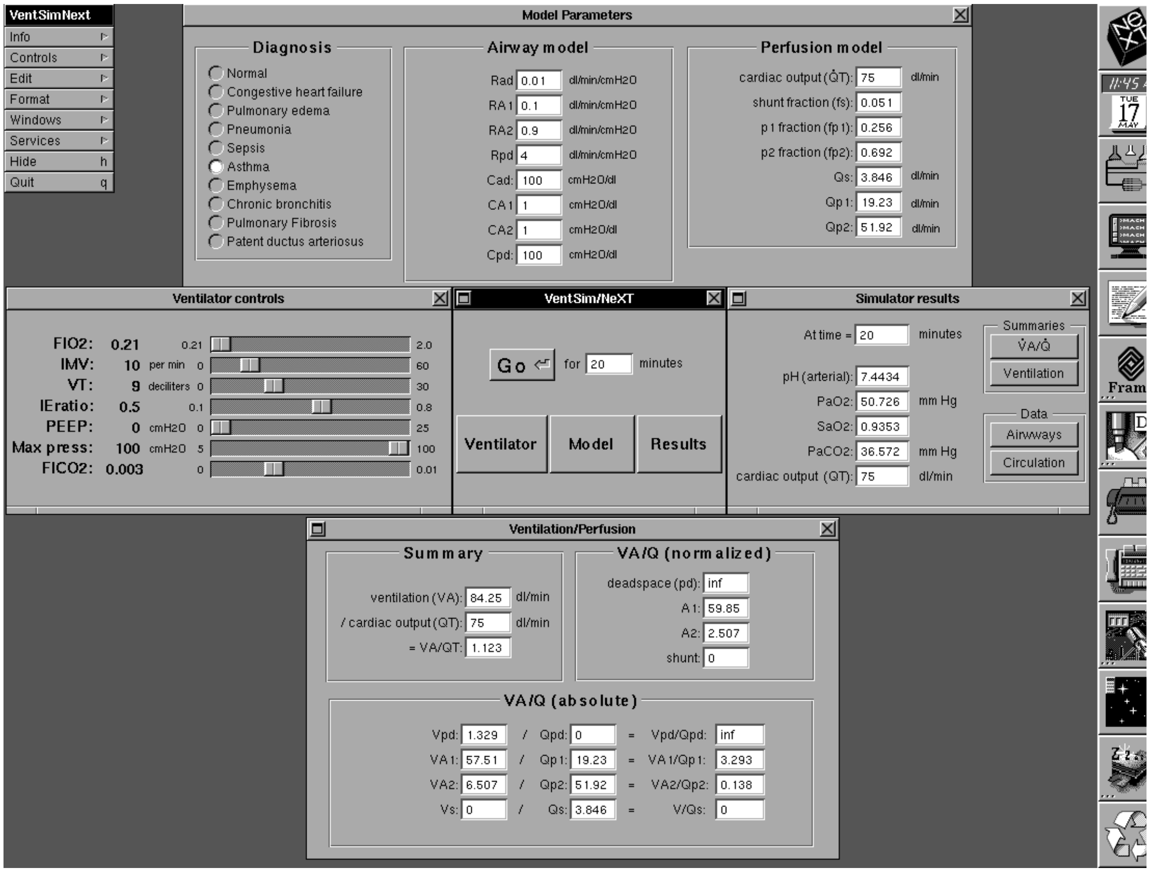

Figure 4.7 VentSim interface. The center three windows show (from left to right) the ventilator controls, the simulator controls, and the simulator results. The button labeled "Model" on the center window brings up the top window, in which the user may adjust the numeric values of model parameters. Buttons on the "Simulator results" window bring up other windows in the lower part of the screen that provide additional details or summaries of model predictions. The VentSim model is implemented in the C programming language; the interface is a NeXTSTEP program.

Table 4.2 Model-computation time (seconds).[a]

Task	VentPlan (analytic solution)	VentSim (analytic solution)	VentSim (numeric simulation)
Model evaluation	0.05	1.96	53
Data-set interpretation	60	7,100	190,800

a. The model evaluation times are the times for the models to compute the effects of a change in ventilator settings when the values of all model parameters are known. VentPlan and VentSim perform model evaluation by solving for the roots of the steady-state equations (analytic solution); VentSim evaluates certain models by doing numeric simulation for 20 minutes of simulated time. The data-set interpretation times are the times that are required to fit each model's parameters to a set of 10 quantitative observations. VentPlan requires approximately 1200 model evaluations to perform this fitting for four parameters. The data-set interpretation times for VentSim are based on the assumption that 3600 model evaluations are required to fit nine model parameters to the set of 10 observations. All times are measured with the Unix function getrusage(), running under the NextSTEP-3.2, on a NeXTstation (Motorola68040 processor with a clock rate of 25 MHz).

Because the ventilator provides a cycling input to the airway component, the full model predicts that, as time increases, the variables of the model will not approach a single value, but rather will oscillate around average steady-state values. If we assert the assumption of continuous ventilation, the model predicts that all variables of the model will reach constant steady state values—the average values for these variables during the cycling steady state.

Table 4.2 compares the computation times for the VentPlan and the VentSim models. For the implementation of these models in the C programming language, evaluated on a NeXTstation, VentPlan computes steady state predictions in 0.05 seconds, and takes approximately 1 minute to fit a set of 10 clinical data points. VentSim computes steady-state solutions in approximately 2 seconds, and requires approximately 116 minutes to fit the same set of 10 data points. The greater than 100-fold increase in computation time for VentSim to fit a set of data, compared to the time required by VentPlan, is due to a 40-fold increase in the time per model evaluation, and a three- to four-fold increase in the number of model evaluations that are needed to fit the additional parameters of the more complex model.[11]

Table 4.3 Characteristic parameter values for the VentSim oxygenation model.[a]

Diagnosis	Q_T (dl/min)	f_s	f_{p1}	R_{A1} (cm H_2O min/dl)	R_{A2} (cm H_2O min/dl)	C_{A1} (dl/ cm H_2O)	C_{A2} (dl/ cm H_2O)
Normal	50	0.05	0.42	0.075	0.08	0.95	1.0
ARDS[b]	100	0.4	0.25	0.075	0.08	0.2	0.5
Pulmonary edema	25	0.20	0.35	0.075	0.08	0.95	1.0
Asthma	75	0.10	0.10	0.15	0.5	1.0	1.0
Pulmonary embolus	44	0.18	0.24	0.11	0.06	0.8	1.0

a. Q_T: cardiac output, in dl/min; f_s: shunt fraction; f_{p1}: first pulmonary-compartment fraction; R_{A1}, R_{A2}: resistance of first and second alveolar compartments, in cm H_2O min/dl; C_{A1}, C_{A2}: compliance of first and second alveolar compartments, in dl/cm H_2O. The metabolic rate ($\dot{V}_{O_2}$) was set to 2.5 dl/min in all cases.

b. ARDS: adult respiratory distress syndrome

4.4 Patient Simulations

The VentSim model is able to represent cardiopulmonary physiologic abnormalities that correspond to many common clinical conditions. Table 5.3 lists sets of typical parameter values for the VentSim-oxygenation model that correspond to diagnoses that are common among patients in the ICU. For example, the diagnosis *normal* corresponds to VentSim parameters that specify a low shunt fraction ($f_s = 0.05$), a nearly equal fraction of blood flow in each pulmonary perfusion compartment ($f_{p1} = 0.42, f_{p2} = 1 - f_s - f_{p1} = 0.43$), and nearly symmetric values for the resistances and compliances of the two alveolar compartments. For the diagnosis *asthma,* the cardiac output is increased (Q_T = 75 dl/min), the shunt

11. VentPlan fits four parameters to the data (Q_T, fs, $\dot{V}_{O_2}$, RQ), whereas VentSim fits nine parameters (Q_T, fs, f_{p1}, $\dot{V}_{O_2}$, RQ, R_{total}, R_{ratio}, RC_{ratio}, C_{total}). See the footnote to Table 5.1 for parameter abbreviations.

fraction is minimally increased ($f_s = 0.1$), and there is marked asymmetry of blood flow in the perfusion compartments and of airflow in the alveolar compartments.

Figure 4.8 shows the model predictions for the simulation of a normal patient and for the simulation of an asthma patient. For these simulations, the FIO_2 gradually increased, and, at the same time, the RR increased and the V_{Tset} decreased, so the minute ventilation (product of RR and V_{Tset}) remained constant. The figure shows that the arterial oxygen pressure is lower and the mean airway pressure is higher for the patient with severe asthma. The simulation also shows that, as the frequency of ventilation increases, the rate of increase of oxygen concentration falls. This reduced responsiveness to increases in inspired

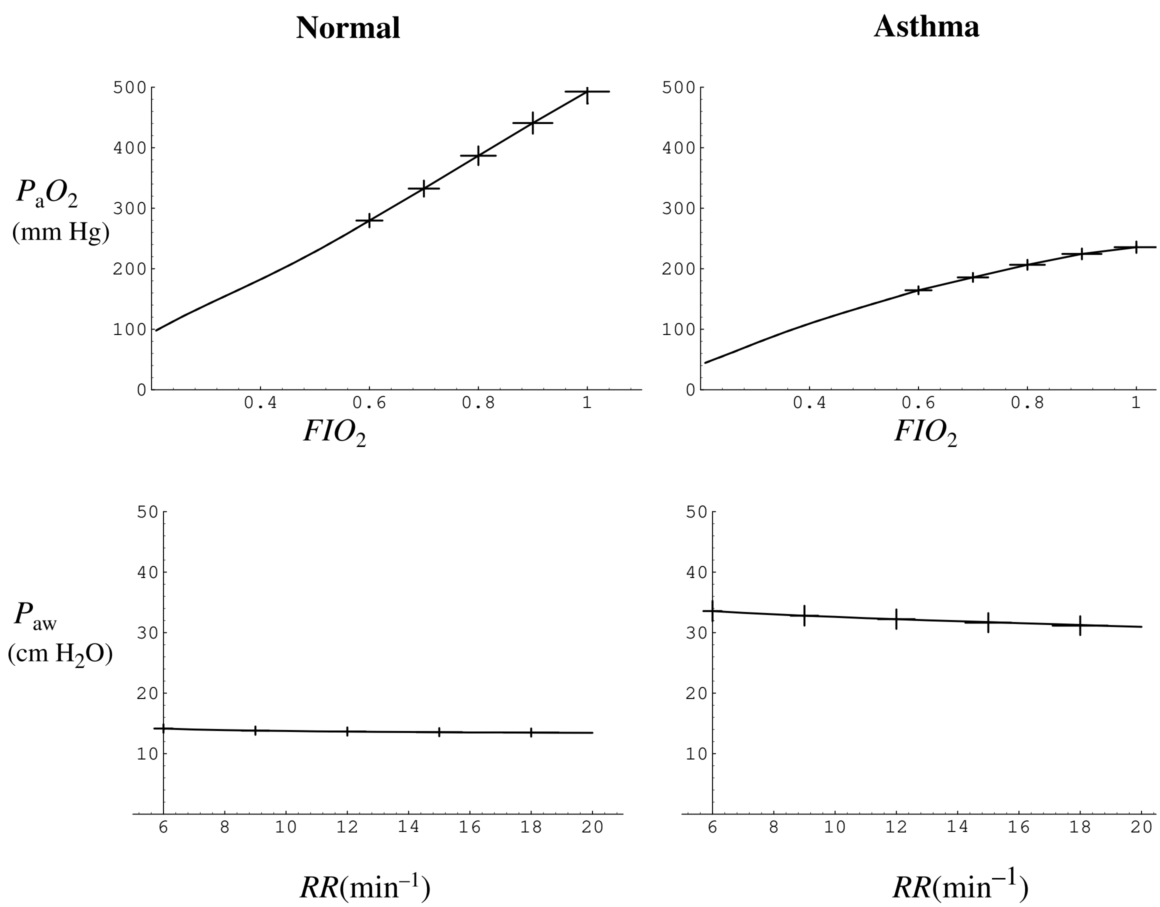

Figure 4.8 VentSim simulation for model parameters that correspond to the diagnosis *normal* (left-hand plots) and to the diagnosis *asthma* (right-hand plots), as specified in Table 4.3. Upper plots show the oxygenation results; lower plots show the mean airway pressures. Ventilator settings that correspond to the crosses are as follows:
$\{FIO_2, V_{Tset}, RR\}$=$\{0.6, 12, 6\}$, $\{0.7, 8, 9\}$, $\{0.8, 6, 12\}$, $\{0.9, 4.8, 15\}$, $\{1, 4, 18\}$.
Abbreviations: see footnote for Table 4.1.

oxygen is due to asymmetric V_A/Q distribution that occurs at higher frequencies of ventilation in this simulated patient who has asymmetric distribution of resistance in the airways.

4.5 Summary

Models of physiology enable us to perform experiments on simulations of physiologic systems. From such experiments, physiologists have derived many insights into the physiologic mechanisms that underlie homeostatic regulation. Physiologic models can help us to interpret the physiologic implications of observations of patients who have abnormal physiology, and can allow us to answer *what if?* questions regarding the response of patients to experimental manipulations.

The VentPlan program—a prototype VMA—demonstrates that a simplified physiologic model provides a basis for evaluating the effects, on patients in an ICU, of changes to the settings of a mechanical ventilator. The VentSim patient simulator demonstrates that a detailed physiologic model can represent a wider range of physiologic abnormalities than can the less detailed VentPlan model. Although the model in VentSim is too computationally complex for use at the inner loop of a VMA, this model forms a reference against which we can compare the performance of models that make simplifying assumptions to reduce their complexity.

Chapter 5

Model Selection Under Time Constraints

In this chapter, I consider the requirements of a real-time control application, as a background for the discussion of model selection under time constraints. I discuss a formalism for describing model-based control applications, and then derive the relative value of alternative models as a function of their time complexity and of the quality of their recommended control actions.

5.1 Computation-Resource Constraints

All computer-based applications must compute their results within the computation-resource constraints of computer memory, rate of computation, and time. In many applications, the computation-resource constraints allow the implementation of a solution method that provides an exact answer. In other applications, the computation task approaches or exceeds the available computation time that the computation machinery requires.

A model-based application computes a model prediction after some computation delay. The length of the computation delay depends on the maximum rate of computation and on the complexity of a model's solution method. For some applications, the tolerable computation delay is measured in hours to days, and the computation-resource constraints do not limit the solution method. For some problems, the finite resources that are available limit the complexity of the solution methods that an application can implement.

For example, Agogino and colleagues developed a model-based application, called IDES, that controlled a robotic manipulator arm for a milling machine. In IDES, a robust auxiliary controller implemented a detailed mathematical model that predicted the behavior of the manipulator arm. Under certain time-critical conditions, the auxiliary controller was unable to compute its recommended control actions within the time available. To maintain real-time response under all conditions, IDES also implemented an adaptive controller that was less complex and less accurate, but that computed a control recommendation in a shorter time. In the presence of unexpected destabilizing disturbances, IDES switched from the more accurate auxiliary controller to the less accurate adaptive controller, so that IDES always computed control inputs before the arm destabilized [Ramamurthi, 1993; Agogino, 1992].

In the setting of limited computer resources, the optimal solution method for a control problem reflects a tradeoff of the cost of computing a control action and the benefit of the computed action. IDES demonstrated that, under differing conditions, this tradeoff led to the selection of differing models for computing the control response. In the IDES system, the benefit of the precise control actions that were computed by the auxiliary controller were balanced by the cost of the computation delay imposed by the auxiliary controller's model of the robotic arm. Under time-critical conditions, the cost of the computation delay imposed by the more accurate auxiliary controller exceeded the additional benefit, and IDES preferred the less accurate adaptive controller.

As another example, consider the problem of the model needed by a VMA such as VentPlan (see Chapter 3 for a description of VentPlan). The VMA implements a physiologic model to explain a patient's clinical observations and to predict the effects of alternative ventilator settings. When a patient's condition is relatively stable, the VMA can devote a relatively longer computation time to solving a more detailed—and more accurate—physiologic model. On the other hand, if the patient's condition deteriorates suddenly, then the VMA must reassess which model and which model parameters most accurately predict the patient's response to alternative control settings. A highly accurate model will lead to an improved ventilator-setting recommendation, whereas a less accurate model will lead to a lower-valued recommendation. For patients with complex physiologic abnormalities, only detailed models that have a higher computation complexity will explain the abnormalities and will make accurate predictions. The value of an improved ventilator-setting recommendation from a highly detailed model is offset by the risk of an adverse event—such as the patient's death—during a longer computation interval.[1] The optimal model to select should reflect a balance of the cost of the computation delay and the expected value of the ventilator-setting recommendation.

In the remaining sections of this chapter, I present a formalism for describing model-based control problems, then I describe of the effects of computation-time delay on the value of the computer-recommended control settings.

5.2 Prediction Models for Model-Based Control

A model-based control program applies a model to compute the effects of alternative control settings, then searches for settings that maximize a value model. The control program compares alternative settings by comparing the value—as determined by a value model—of the predicted effects of the settings. For example, VentPlan is a model-based control program that evaluates the steady-state effects of alternative settings of a ventilator. VentPlan recommends ventilator settings that maximize the value of the predicted state of the patient.

5.2.1 A Mathematical Formulation for Model-Based Control

The vector of prediction variables at steady state is

$$\hat{Y}_i = M_i(X, \Theta_i)\,, \tag{5.1}$$

1. Patients with complex physiological abnormalities are not necessarily unstable (their risk of adverse events may be small), whereas patients with easily modeled physiologic abnormalities may be highly unstable. The tradeoff of model complexity and model accuracy is critical only for unstable patients with complex abnormalities that require highly detailed and computationally complex models for accurate predictions.

in which the subscript i refers to the ith model, X is the vector of control variables, and Θ_i is the vector of parameters of the ith-model.

The model-based control recommendation is

$$X_i = \arg\max_X \; u(X, M_i, \Theta_i) \;, \tag{5.2}$$

in which $u(X, M_i, \Theta_i)$ is a value function that provides a measure of the instantaneous value of the steady state that occurs with control settings X, as defined by model M_i and model parameters Θ_i. This value function assumes that the model predictions are accurate, and that the model parameters are known with certainty.

If the model is known, but the model parameters are uncertain, then the value function u may be replaced with the expected-value function

$$X_i = \arg\max_X \; \mathrm{E}\,[u(X, M_i, \Theta_i)\,] \,. \tag{5.3}$$

That is,

$$X_i = \arg\max_X \int_\Theta u(X, M_i(X, \theta), \theta)\; p(\theta) d\theta. \tag{5.4}$$

Here the integral represents integration over the multidimensional distribution of the parameter estimates Θ_i.

5.2.2 Assumptions of Model-Based Control

This formulation of a control problem makes several assumptions:

- There is a model available that provides predictions of alternative control settings with no computation delay.
- The value of a control setting is dependent on only the control setting and the true value of the current system parameters, Θ_s.
- The optimal control setting is the setting that maximizes the expected value of the control settings, given the current system parameters.

Dynamic models predict the trajectory that a system follows after a change in control variables (inputs). Physiologic systems (which are homeostatic) reach a steady state for any constant input, as long as the system parameters remain fixed. For homeostatic systems, the trajectory is less important than the final steady state, and so a control application may optimize the predicted steady state without explicitly considering the trajectory. This approximation simplifies the computation of the value of alternative control settings.

System parameters change over time, even though these parameters are modeled as system-specific constants. Changes in system parameters may occur gradually or suddenly, due to influences that are not modeled. This formulation of the control problem asserts that the optimal control settings for a system change as the system parameters change. A control application can compute the optimal settings by repeatedly updating the system parameters and recomputing the control settings.

The assumption that there is no computation delay for model predictions allows an application to compare alternative control settings using an instantaneous value function. In the VentPlan prototype, the physiologic model that predicts the effects of alternative settings of the ventilator causes a delay of less than 1 minute in the computation of recommended control settings [Rutledge, 1993a]. Because no alternative strategies that involve longer or shorter delays are available, VentPlan ignores this delay, and compares alternative control settings by considering the expected value of the predicted system state induced by each control setting, $E[\hat{u}_i(X)]$. The function $\hat{u}_i(X) = u(X, M_i, \Theta_i)$ is the value function that describes physicians' instantaneous preferences for setting the ventilator [Farr, 1991].

5.3 Control-Setting Strategies

If a control system has available multiple models that vary in their computation complexity, then the choice of alternative models leads to varying delays before the model-based recommendations are available, and leads to varying quality of control-setting recommendations. Each model computes a different strategy for setting the ventilator.

A **strategy**, s, is a sequence of control actions over time; $s(t)$ denotes the control setting that is in effect at time t. The strategy that results from the selection of model M_i is

$$s_i(t) = \begin{cases} X_o, & 0 \le t < t_i \\ X_i, & t_i \le t < t_{period} \end{cases}, \tag{5.5}$$

in which t_i is the computation delay (the time taken by model M_i to compute X_i), and t_{period} is the length of time after which the control settings may be adjusted again. That is, the strategy $s_i(t)$ has an initial interval of length t_i, during which the control settings remain at their previous settings (X_o), then an interval of length $t_{period} - t_i$, after a change to new control settings at time $t = t_i$. The new settings remain in effect at least until time $t = t_{period}$. Alternative models lead to different strategies because each model leads to a different recommendation after a different computation time.

5.4 Value of Control Settings Over Time

We defined $u(X) = u(X, M_s, \Theta_s)$ as the instantaneous value of a control setting. This function may reflect an objective value that is based on observable outcomes, or it may represent the preferences of experts for the setting of the control variables. For example, the authors of the VentPlan prototype assessed the value function by eliciting the subjective preferences of physicians for setting the controls of the ventilator for patients in varying clinical states. In an application to control assembly-line production, the value function might be an observable measure of productivity, such as the number of correctly assembled units per time period.

The instantaneous value function allows the comparison of alternative control settings, but does not allow the comparison of alternative strategies. Applications that compute multiple alternative control-setting strategies require a method to compare strategies that implement higher-value control settings after a longer computation delay with strategies that implement lower-value control settings after shorter computation delays.

One method to compute the value of a sequence of control settings over time is to evaluate the probability that the system will make a transition to a lower-valued state during an interval of interest. For example, a control application may attempt to avoid a given event or combination of events—*catastrophes*—by taking actions that minimize the cumulative probability that one of these events will occur. If an application minimizes the probability of adverse events, the probability that the system will remain in the current state or will move to a higher-valued state is maximal.

Ventilator settings are suboptimal if the settings cause an increased chance that an adverse event will occur, compared to some other ventilator setting. For example, ventilator settings that allow a patient to have a very low level of oxygen in the arterial blood are suboptimal, because low levels of oxygen increase the risk of ventricular fibrillation.[2] The risk returns to a lower—usually negligible—magnitude for alternative settings that increase the oxygen level to normal. Similarly, settings are suboptimal if they cause a high mean airway pressure, which increases the risk of developing a tension pneumothorax.[3] Optimal ventilator settings minimize the risk of adverse events.

A value function to assess alternative strategies can be based on the probability that an adverse event will occur during a time interval of interest. During any short interval, the risk per unit time of an adverse event is constant, and depends on the state of the system (as defined by Θ_s),[4] and the settings of the control variables. Let A represent the initial state and

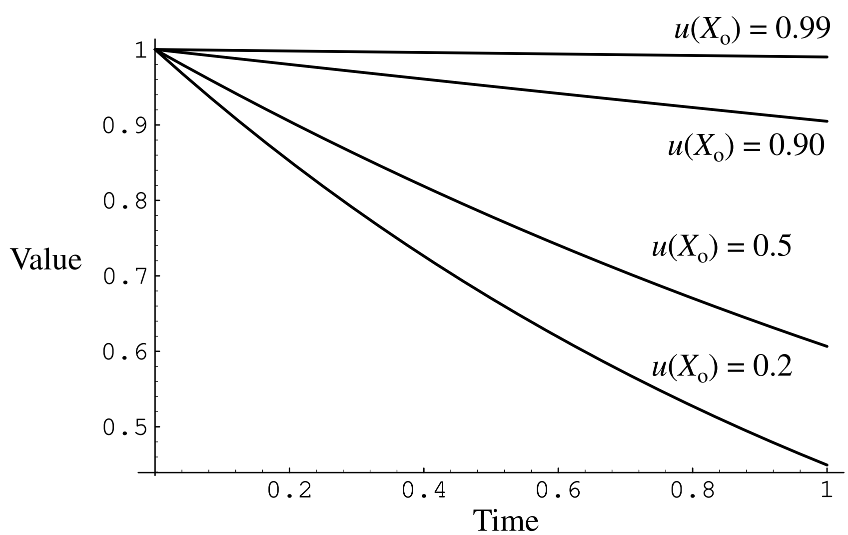

Figure 5.1 Value of a control setting over time. The value of a control setting X_o, is shown as a function of the length of time that the control setting is in effect, for various instantaneous values, $u(X_o)$. The value decreases exponentially with time. For a $u(X_o)$ that is close to 1, the decay in value with increasing time is approximately linear.

---

2. Ventricular fibrillation (VF) is a chaotic cardiac rhythm in which the heart's pumping action ceases.
3. A tension pneumothorax is a collapse of a lung that is accompanied by a buildup of air pressure around the lung, which leads to compromise of the blood flow to the heart.

all higher-valued states, and let $\bar{A}$ represent all lower-valued states of the system. The probability that the system will still be in one of the states in A after time t, $p(A, t)$, is a measure of the value of a strategy. That is, the value of the ith-model strategy for the interval $(0, t_{\text{period}})$ is

$$v(s_i) = p(A, t_{\text{period}}). \tag{5.6}$$

We can interpret the instantaneous value function, $u(X)$, as a measure of the probability per unit time that the system remains in one of the states in A, and then $k(X) = 1 - u(X)$ is the rate that a transition occurs from a state in A to a state in $\bar{A}$. Where convenient, we set our unit of time to length t_{period}, which calibrates $k(X)$ in units of probability per t_{period}.

The first derivative of $p(A, t)$ with respect to time is

$$p'(A, t) = k(X)p(A, t). \tag{5.7}$$

Integrating both sides with respect to time gives

$$p(A, t) = \int -k(X)p(A, t)dt. \tag{5.8}$$

The solution for $p(A, t)$, when $k(X)$ is a constant, is

$$p(A, t) = e^{-k(X)t} + C, \tag{5.9}$$

and setting the value of $p(A, t)$ to 1 at time 0 sets the constant term to 0. The value of the strategy that sets the control variables to X for time t is

$$v(s) = e^{-k(X)t}. \tag{5.10}$$

Figure 5.1 shows the relationship of the value of a strategy to the computation delay. This figure shows graphically that, as $u(X)$ approaches 1, the exponential decay in value is approximately linear. Figure 5.2 illustrates the value over time for strategies that have very short, intermediate, and very long computation delays.

---

4. For the purpose of assigning a value to the state of the system, Θ_s may include additional parameters that would not otherwise be needed in a model that predicts Y.

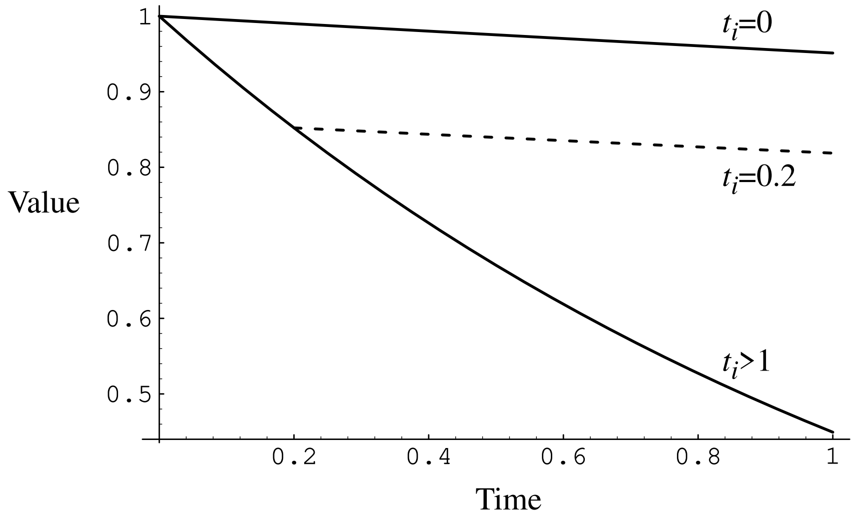

Figure 5.2 Value over time for strategies with a computation delay. After a computation delay, t_i, the control settings change from the initial values, X_o, to the values that are computed by the ith model, X_i. The dashed line shows the value over time for a computation delay of t_i=0.2. The upper solid line represents no computation delay (t_i=0); the lower solid line represents a long computation delay (t_i>1). The instantaneous values of the control settings are $u(X_o) = 0.2$ and $u(X_i) = 0.95$.

5.4.1 Linear-Value Assumption

The **linear-value assumption** asserts that changes in value with time are linear. That is, under this assumption, the value of a delayed control setting decreases linearly with increasing computation time. For control systems in which the objective is to avoid adverse events, the decrease in value with time is an exponential decay, as described by Equation 5.10. However, for small values of $k(X)\,t$, the linear function $1 - k(X)\,t$ is a close approximation of the exponential decay function $e^{-k(X)t}$. In other words, as long as the rate of adverse events is small, the linear value assumption holds. Table 5.1 compares e^{-k} to $1 - k$ for values of k from 10^{-1} to 10^{-5}. For values of k as large as 0.1, the exponential decay differs from the linear expression by less than 0.5 percent.

Table 5.1 Approximation of e^{-k} by $1-k$.

k	e^{-k}	$1-k$	$e^{-k} - (1-k)$	$(1-k)/e^{-k}$
0.1	0.905	0.9	0.005	0.995
0.001	0.9990005	0.999	5.0×10^{-7}	0.9999995
0.00001	0.99999000005	0.99999	5.0×10^{-11}	1.0

The linear-value assumption also applies in other control applications for which the decay of value over time is linear for all decay rates. For example, a model-based controller that maximizes the rate of a mechanized assembly process has an instantaneous value of a control setting that is proportional to the rate of production. In this case, the value of a strategy is linear in both time and instantaneous value, because the benefit of a control setting is the integral of the rate of production with respect to time.

When $u(X)$ changes during a time period of interest, the probability of A is the product of the probabilities of A for each subinterval during which $u(X)$ is constant. That is, we compute the value of a strategy as the product of the values of each interval during which the control settings are constant. For a strategy, s_i, that has a single change in control settings from X_o to X_i that occurs at time t_i,

$$v(s_i) = e^{-k(X_o)t_i} e^{-k(X_i)[t_{\text{period}} - t_i]}, \tag{5.11}$$

$$v(s_i) = e^{-k(X_o)t_i - k(X_i)[t_{\text{period}} - t_i]}. \tag{5.12}$$

5.4.2 Maximum Value of a Control Setting

The *maximal value of a control setting,* $v^{\max}(X_i)$, is the value that the control setting would have if there were no computation delay (that is, if $t_i = 0$). From Equation 5.10,

$$v^{\max}(X_i) = e^{-k(X_i)t_{\text{period}}}. \tag{5.13}$$

For values of $u(X_o)$ that approach 1, $k(X_o)$ approaches 0, and the linear-value assumption applies

$$v^{\max}(X_i) \cong 1 - k(X_i)t_{\text{period}}\,. \tag{5.14}$$

If we calibrate the $k(X_i)$ in units of time equal to t_{period}, then

$$v^{\max}(X_i) \cong u(X_i)\,. \tag{5.15}$$

The maximum change in value due to a control setting is

$$\Delta v^{\max}(X_i) \;=\; \Delta v^{\max}(X_i, X_o) \;=\; [u(X_i) - u(X_o)]\, t_{\text{period}}\,. \tag{5.16}$$

For a model M_i that computes a recommended control action X_i, the greatest possible benefit of the model is $b(M_i)$, which is the value of the model when the computation delay is negligible.

5.4.3 Cost of Computation Delay

A cost function provides a measure of the decrease in value of the strategy s_i that results from the need to compute the model-based optimal control setting X_i. The full cost of computing a model recommendation includes the direct monetary cost associated with the computing device and the cost resulting from the model-imposed delay for the computation of X_i. I assume that the monetary cost of the computing device is a sunk cost [Keeney, 1976]. I set the cost of the computation delay to the decrease in value that occurs due to the fact that an adverse event might occur during the delay. During the computation delay, the control settings remain at X_o, so the value of the strategy decays at a rate $k(X_o)$. The *cost of the computation delay* is the difference between the decay in value that occurs, during the computation interval from $t = 0$ to $t = t_i$, due to the settings X_o, and the decay in value, during the same interval, that would occur if the settings X_i were implemented at $t = 0$:

$$c(t_i, X_i) \;=\; e^{-k(X_o)t_i} - e^{-k(X_i)t_i}. \tag{5.17}$$

Figure 5.3 plots the cost of computation delay for various values of $k(X_i)$ and $k(X_o)$. This figure shows that the cost increases linearly from 0 to 0.05 for a wide range of values of $k(X_i)$ and $k(X_o)$. The maximum cost plotted in this figure corresponds to a 5-percent risk of

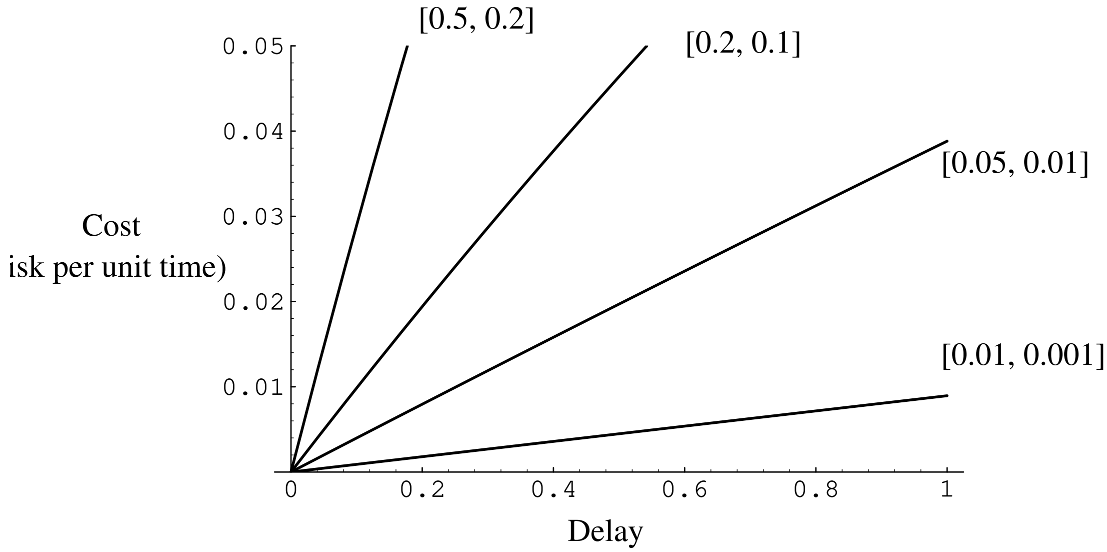

Figure 5.3 Cost of a computation delay. The cost of delay is the difference of two exponential decays, as defined in Equation 5.17, for a change from an initial decay rate $k(X_o)$, to a smaller decay rate $k(X_i)$ after the computation delay. Each plot is labeled with the pair of decay rates as $[k(X_o), k(X_i)]$. The delay is calibrated in units of t_{period}.

an adverse event during the time interval, which is a large risk to incur for a single interval between control-setting adjustments. That is, the linear-value assumption applies even in situations where the cumulative risk of adverse events is very high, because the risk per interval during which the control settings remain constant is not high enough for the exponential decay in value to deviate substantially from a linear decay.

By the linear-value assumption,

$$c(t_i, X_i) \cong [k(X_o) - k(X_i)]\, t_i = [u(X_i) - u(X_o)]\, t_i, \tag{5.18}$$

and, for a model M_i that computes recommendations X_i after time t_i,

$$c(M_i) = c(t_i, X_i). \tag{5.19}$$

The cost of delay, $c(t_i, X_i)$, increases linearly with the length of the delay, and also with the magnitude of increase in instantaneous value of the new control setting. If the current setting is close to optimal, the cost of a delay is small. On the other hand, the farther the current setting is from the optimal setting, the larger is the cost of a delay.

5.4.4 Value Of A Control Strategy

The value of a strategy is (from Equation 5.12)

$$v(s_i) = e^{-[k(X_o) - k(X_i)]t_i - k(X_i)t_{\text{period}}}, \tag{5.20}$$

which is closely approximated by

$$v(s_i) = 1 - k(X_i)t_{\text{period}} - [k(X_o) - k(X_i)]\,t_i. \tag{5.21}$$

For t_{period} equal to 1,

$$v(s_i) = u(X_i) - [u(X_i) - u(X_o)]\,t_i. \tag{5.22}$$

The first term in the right-hand side of Equation 5.22 is the maximal value of a control setting (Equation 5.15); the second term is the cost of the computation delay (Equation 5.18). Combining Equations 5.22, 5.15, and 5.18 gives:

$$v(s_i) = v^{\max}(X_i) - c(t_i, X_i). \tag{5.23}$$

The value of an alternative, model-based strategy is the maximum value of the model's computed control setting reduced by the cost of the computation time that the model requires to compute the setting.

A consequence of the linear-value assumption is that the value of a strategy is the integral over time of the instantaneous values of the control settings. That is, the value of a model selection is

$$v(s_i) = \int_0^{t_{\text{period}}} u(s_i(\tau), M_s, \Theta_s)\,d\tau, \tag{5.24}$$

assuming that we select model M_i at $t = 0$, and the model leads to the strategy s_i.

If the strategy s_i adjusts the control settings once during the interval $(0, t_{\text{period}})$, at time t_i, and if $t_i \le t_{\text{period}}$, then the value of the strategy is

$$v(s_i) = \int_0^{t_i} u(X_o)d\tau + \int_{t_i}^{t_{\text{period}}} u(X_i)d\tau \qquad (5.25)$$

For $u(X)$ constant, Equation 5.25 gives

$$v(s_i) = u(X_o)t_i + u(X_i)(t_{\text{period}} - t_i), \qquad (5.26)$$

which is equivalent to Equation 5.22.

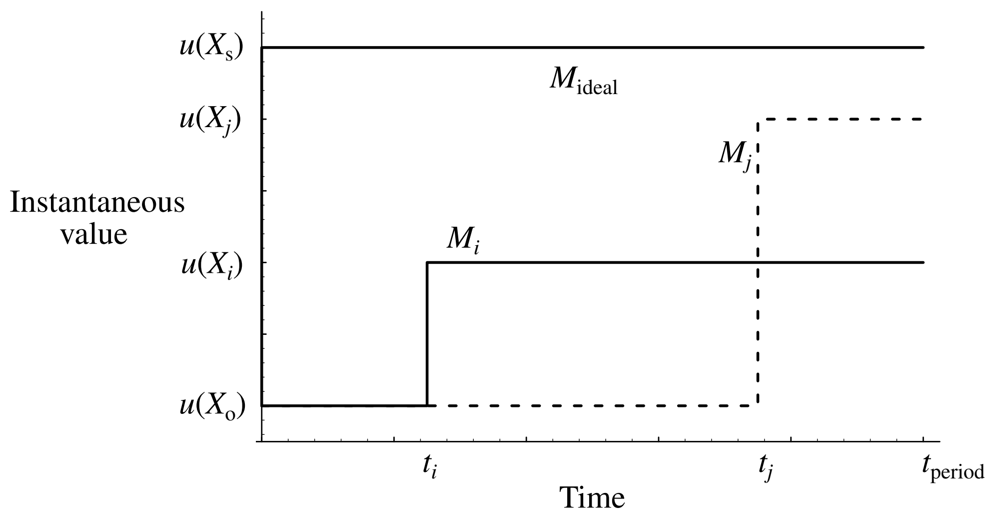

Figure 5.4 Instantaneous value of control settings versus time. The values of the strategies that are computed by M_i and M_j correspond to the areas under the plots of instantaneous value versus time. At $t = 0$, new observations indicate that the control settings X_o are not optimal. Model M_i requires time t_i to compute its recommended control settings X_i. A more complex model M_j requires a greater time, t_j, to compute control settings X_j. M_{ideal} is an hypothetical model that computes the optimal control setting, X_s, with no computation delay. $u(X)$ is the instantaneous value of the control setting X. t_{period} is the length of time after which the control settings will be reassessed.

Equation 5.24 suggests that a graphical representation of $v(s_i)$ is the area under the plot of instantaneous value versus time. Figure 5.4 shows the graph of instantaneous value versus time for strategies that result from the selection of three alternative models. M_{ideal} is a hypothetical model that computes the globally optimal control settings, X_s, with no computation delay. M_i and M_j are alternative models that require different computation times to compute their recommended control settings. The area under the plot of each strategy represents the value of the strategy for the interval (0, t_{period}).

Figure 5.5 shows how the value of a strategies changes as a function of the computation delay, for two control settings. The maximum value of each strategy, $v^{\max}(s)$, corresponds to the value that the strategy would have if there were no computation delay. As the computation delay increases, the controls remain at the original settings for a greater proportion of the interval, and the value of the strategy approaches the value of the strategy that makes no changes to the control settings. The points labeled M_i and M_j correspond to the strategies of Figure 5.4, for which $v(s_i) > v(s_j)$.

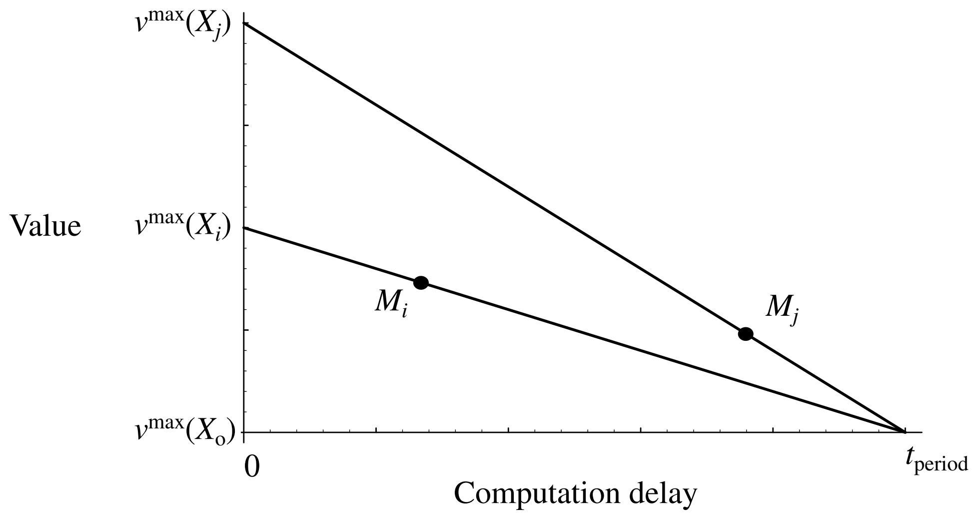

Figure 5.5 Value of a strategy versus computation delay. The plot shows the effect of increasing computation delay on the change in value of two control settings, X_i and X_j, for the interval $(0, t_{\text{period}})$. $v^{\max}(X)$ is the maximum value of the control setting X, which occurs when there is no computation delay. The points labeled M_i and M_j correspond to the computation delays of the models in Figure 5.4.

5.4.5 Comparison of Alternative Models

The *value of a model* is the value of the strategy that the model computes, $v(M_i) = v(s_i)$.

From inspection of Figure 5.4, and from Equation 5.22, the difference in value between any two alternative models is

$$\Delta v(M_i, M_j) = \Delta v(s_i, s_j) = v(s_i) - v(s_j), \tag{5.27}$$

$$\Delta v(M_i, M_j) = [u(X_i) - u(X_o)]\,(t_{\text{period}} - t_i) - [u(X_j) - u(X_o)]\,(t_{\text{period}} - t_j)\;. \tag{5.28}$$

Setting $\bar{t}_i = t_{\text{period}} - t_i$ and rearranging, we obtain

$$\Delta v(M_i, M_j) = u(X_o)(t_i - t_j) + u(X_i)\bar{t}_i - u(X_j)\bar{t}_j. \tag{5.29}$$

Now let us consider the strategy s_i that improves the current settings after time t_i, and the strategy s_{si} that, after the same time, sets the control settings to their optimal value X_s. The difference in value between these strategies is

$$\Delta v(s_{si}, s_i) = [u(X_s) - u(X_i)]\,\bar{t}_i. \tag{5.30}$$

If we believe that the optimal control settings reduce the risk of adverse events to zero, then $u(X_s) = 1$, and

$$\Delta v(s_{si}, s_i) = [1 - u(X_i)]\,\bar{t}_i = k(X_i)\bar{t}_i. \tag{5.31}$$

Similarly, the difference in value between the strategy that is computed by M_i and the strategy that is computed by the current model M_o is

$$\Delta v(M_i) = \Delta v(M_i, M_o) = [u(X_i) - u(X_o)]\,\bar{t}_i \tag{5.32}$$

Substituting back $\bar{t}_i = t_{\text{period}} - t_i$, and rearranging, we obtain

$$\Delta v(M_i) = [u(X_i) - u(X_o)]\,t_{\text{period}} - [u(X_i) - u(X_o)]\,t_i. \tag{5.33}$$

Finally, we note that the first term in this equation is the difference between the maximum values of the computed control setting and of the current control setting, and the second term is the cost of the model computation delay. If we define the benefit of a model selection to be

$$b(M_i) = v^{\max}(X_i) - v^{\max}(X_o), \tag{5.34}$$

then the change in value that occurs after selection of a model is the benefit of the model selection reduced by the cost of the model:

$$\Delta v(M_i) = b(M_i) - c(M_i). \tag{5.35}$$

With this analysis of the value of a model selection, we now consider the problem of how to select models for a real-time control application in which the time constraint limits the complexity of the optimal model.

5.5 The Optimal Model Under a Time Constraint

A model-based control application requires a model to predict the effects of control settings during a search for the optimal settings. If there is time to compute the base model[5] and all less complex models, an application can compare the value of the recommendations made by each model to find the optimal model M_{opt}:

$$\Delta v(M_{\text{opt}}) = \arg\max_i \ \Delta v(M_i) = \arg\max_i \ u(X_i | M_i)\bar{t}_i. \tag{5.36}$$

Equivalently,

$$\Delta v(M_{\text{opt}}) = \arg\min_i \ k(X_i | M_i)\bar{t}_i. \tag{5.37}$$

That is, to calculate the model that represents the optimal tradeoff of computation-time delay and value of recommended action, we must first compute the recommended actions of all models. The optimal model is the model that we would have selected initially, if we had known the computation delays and the relative values of model-based recommendations for all alternative models.

Under a time constraint, we might attempt to search among the models until we found a model with a high value. Unfortunately, to compute a value for a single model, we must apply the reference model, M_s, to predict the effects of X_i. If we were able to evaluate the reference model in real time, we would apply the reference model to compute the optimal settings, and would not need to search for a simplified, approximate model.

The definition of the optimal model in Equation 5.37 is not directly helpful if we wish to determine dynamically the optimal model for a real-time control application. However, we can apply the definition to compute the optimal model *after the fact* to assess the

5. As defined in Section 2.2, the base model is the most detailed and accurate model available— the base model is a reference against which we would like to compare other models.

performance of other methods for selecting models dynamically. The definition of the optimal model provides a standard against which to test an heuristic or approximate model-selection algorithm.

5.6 Approximate Dynamic Selection of Models

The optimal model leads to a change in value that is a tradeoff of the value of a higher-valued computed control setting and a cost of a model computation delay. A dynamic-selection algorithm that selects models for a real-time application is unable to compute this tradeoff exactly.

A dynamic-selection method may be able to approximate this tradeoff to select a model that is likely to be the optimal model. The change in value due to the selection of model M_i is

$$\Delta v(M_i) = b(M_i) - c(M_i). \tag{5.35}$$

This analysis suggests that a heuristic for dynamic model selection might approximate the optimal model by comparing a measure of the benefit of a model with a measure of the cost of the model.

5.6.1 Benefit of a Model

The difference in benefit between a simple model and a more complex model is due to the difference in the models' computed optimal control settings. As a model's prediction accuracy decreases, the inaccurate model's computed optimal control settings differ from the reference model's computed optimal control settings. That is, because both models attempt to optimize the same value function, $u(X)$, differences in the predictions of the effects of X are likely to lead to different values of X that maximize $u(X)$. If we believe that the reference model is accurate, then any model that computes different control settings must be inaccurate.

The relationship of accuracy and value of computed optimal control settings suggests that a measure of model accuracy also would be a measure of the maximum value of a model's computed optimal control settings.

5.6.1.1 Model Adequacy

I asserted in Section 2.7 that there is no such entity as a correct model, and that all models are inaccurate to some degree, in some experimental frame. Nevertheless, we can consider a *hypothetical* correct model, and ask questions of the form: "If a given model were correct, what is the probability that the observed data would occur?" For practical purpose, I define any model whose behavior in a specific experimental frame is indistinguishable from the system behavior to be a correct model in that experimental frame, and I refer to such a model as a correct model. However, what matters is not whether a model is correct, but whether a particular model under consideration is accurate enough to be useful.

The concept of *accurate enough to be useful* suggests a definition for **model adequacy**. I define the event that the ith model is adequate as the intersection of the events that all of the ith model's predictions are adequate:

$$M_i^A = \bigcap_j Y_{i_j}^A, \tag{5.38}$$

where $Y_{i_j}^A$ is the event that predictions for the jth variable are adequate. The ith model's predictions for the jth variable are adequate if the probability that the prediction errors are within the bounds for adequacy ($\pm\varepsilon_j\sigma_j$) is greater than a decision-threshold probability. That is,

$$Y_{i_j}^A = p(|\hat{y}_{i_j} - y_{\mathrm{obs}_j}| < \varepsilon_j\sigma_j) > 1 - \delta_A, \tag{5.39}$$

in which $\hat{y}_{i_j}$ is the ith model's prediction for the jth observation, y_{obs_j} is the jth observation, σ_j is the standard deviation of the observation error for the jth observation, ε_j is a scaling factor that specifies the relative accuracy that the jth prediction variable must satisfy for the model to be adequate, and $1 - \delta_A$ is the decision threshold. The decision threshold is less than 1 because the distribution of observation errors includes a nonzero probability that the magnitude of an error will exceed the $\pm\,\varepsilon_j\sigma_j$. bound for adequate prediction accuracy.

The relationship of ε_j and δ_A follows from the requirement that a correct model is an adequate model. From Equations 5.38 and 5.39, the probability that a correct model is adequate is

$$p(\mathrm{M}_i^{\mathrm{A}}|\mathrm{M}_i^{\mathrm{c}}) = p(\bigcap_j (\mathrm{Y}_{i_j}^{\mathrm{A}}|\mathrm{M}_i^{\mathrm{c}})). \tag{5.40}$$

To evaluate the probability that a correct model will be adequate, we must evaluate

$$p(|\hat{y}_{i_j} - y_{\mathrm{obs}_j}| < \varepsilon_j\sigma_j) = \int_{-\varepsilon_j}^{\varepsilon_j} p\left(\left(\frac{\hat{y}_{i_j} - y_{\mathrm{obs}_j}}{\sigma_j}\right) = x\right)dx. \tag{5.41}$$

Assuming that the observation errors are normally distributed, unbiased, and independent, we have

$$p(|\hat{y}_{i_j} - y_{\mathrm{obs}_j}| < \varepsilon_j\sigma_j) = \frac{1}{\sqrt{2\pi}}\int_{-\varepsilon_j}^{\varepsilon_j} e^{-\frac{x^2}{2}}dx. \tag{5.42}$$

Defining the error function, erf(z), as

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt, \tag{5.43}$$

gives

$$p(|\hat{y}_{i_j} - y_{\mathrm{obs}_j}| < \varepsilon_j\sigma_j) = \mathrm{erf}(\frac{\varepsilon_j}{\sqrt{2}}). \tag{5.44}$$

Substituting Equation 5.44 into Equation 5.39 gives

$$\mathrm{Y}_{i_j}^{\mathrm{A}} = p(\mathrm{erf}(\frac{\varepsilon_j}{\sqrt{2}})) > 1 - \delta_{\mathrm{A}}, \tag{5.45}$$

which is true for

$$\varepsilon_j \geq \sqrt{2}\mathrm{erf}^{-1}(1 - \delta_{\mathrm{A}}). \tag{5.46}$$

This restriction on the minimum value for ε_j guarantees that correct models are adequate, and that $p(\mathrm{M}_i^{\mathrm{A}} | \mathrm{M}_i^{\mathrm{c}}) = 1$. For example, if we set δ_{A} to 0.05, Equation 5.46 specifies $\varepsilon_j >$ 1.96, and all scaling factors for adequate prediction accuracy must be at least 1.96.

The observation errors reduce our ability to distinguish accurate models from inaccurate models, and set a lower limit on the size of prediction errors that we must allow for a model to be adequate.

5.6.1.2 A Measure of Benefit

The probability that a model is adequate is a computable measure of the accuracy of a model. The probability of adequacy provides a measure of how probable it is that a model under consideration would have generated the observations within the limits of adequate prediction accuracy. We can consider the natural logarithm of the posterior probability of model adequacy, given both the set of observations ***Y*** and the prior information ξ, to be a measure of the benefit of a model, $\hat{b}(M_i)$,

$$\hat{b}(M_i) = \log p(\mathrm{M}_i^{\mathrm{A}} | \boldsymbol{Y}, \xi). \tag{5.47}$$

Application of Bayes' rule to expand $p(\mathrm{M}_i^{\mathrm{A}} | \boldsymbol{Y}, \xi)$ gives

$$\hat{b}(M_i) = \log p(\mathrm{M}_i^{\mathrm{A}} | \xi) + \log p(\boldsymbol{Y} | \mathrm{M}_i^{\mathrm{A}}, \xi) + \log C, \tag{5.48}$$

in which C is a constant equal to $1/p(\boldsymbol{Y})$. The first term, $p(\mathrm{M}_i^{A} | \xi)$, is the probability that the model is adequate, given only the prior information ξ. The second term, $p(\boldsymbol{Y} | \mathrm{M}_i^{\mathrm{A}}, \xi)$, is the conditional probability that the errors would occur, given that the model is adequate, and under the assumption that the parameters of the model are the maximum-likelihood estimates for the observations ***Y***.

The prior information, ξ, represents all quantitative and nonquantitative data, other than ***Y***, that have any information about whether or not a model is likely to make accurate predictions. For example, for the decision of which model to select for a VMA, the information that a patient suffers from asthma has an important effect on which models are likely to provide adequate prediction accuracy, and which will be inadequate. I shall assume that all probabilities are assessed in light of the prior information, ξ, and will drop this term from my notation.

One approach to computing $p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{A}})$ is to note that the likelihood of the data, $p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{c}})$, is a lower bound on $p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{A}})$. That is,

$$p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{A}}) \geq p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{c}}). \tag{5.49}$$

This relationship is true because a correct model is, by definition, an adequate model. We can use the probability that a model is correct—that is, indistinguishable from the source of observations—as a conservative estimate of the probability that a model is adequate. As described in Section 2.7, we can assess $p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{c}})$ as a penalized likelihood from the value of R_i, the sum of squared residuals of the ith model, evaluated at the maximum likelihood estimates of the m_i model parameters. $p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{c}})$ is the probability that the observed value of R_i would occur. We may prefer to consider the probability that errors as large as those observed would occur when the model is, in fact, the source of the observations.

This related concept—the probability that a set of observations differs from the predictions by as much as does $\boldsymbol{Y}$—is an alternative measure of the probability of the data, given that the model is adequate. If the model is indistinguishable from the system—that is, if the data could have been generated by independent observations of the model—R_i is distributed as a χ^2 random variable with $N - m_i$ degrees of freedom. I replace $p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{c}})$ with the probability that R_i would be as large as the observed value, given that the model is adequate:

$$p(\chi^2 \leq R_i|\mathrm{M}_i^{\mathrm{c}}) = \int_0^{R_i} p(\chi^2 = z|\mathrm{M}_i^{\mathrm{c}})dz. \tag{5.50}$$

This probability is a measure of the goodness of fit of the model to the data, and model builders use this measure to assess the ability of a model to explain a set of observations. We can find values for this integral in standard statistical tables, or we can compute the values from the incomplete gamma function, $\Gamma(a, x)$ [Press, 1988; Ott, 1988]:

$$\Gamma(a, x) = \int_0^x e^{-t} t^{a-1} dt, \tag{5.51}$$

$$P(a, x) = 1 - \frac{\Gamma(a, x)}{\Gamma(a, \infty)}, \tag{5.52}$$

and

$$p(\chi^2 \le R_i | \mathrm{M}_i^{\mathrm{c}}, \xi) = P(\frac{\chi^2}{2}, \frac{m_i}{2}). \tag{5.53}$$

Replacing $p(\boldsymbol{Y}|\mathrm{M}_i^{\mathrm{A}}, \xi)$ in Equation 5.48 with the expression in Equation 5.53 gives an alternative metric for the benefit of the control settings that are computed by a model:

$$\hat{b}(M_i) = \log p(\mathrm{M}_i^{\mathrm{A}}) + \log p(\chi^2 \le R_i | \mathrm{M}_i^{\mathrm{c}}) + b_{\max}. \tag{5.54}$$

If the prior probability of the model is 1, and as the model-prediction errors approach 0 (that is, $p(\chi^2 \le R_i | \mathrm{M}_i^{\mathrm{c}})$ approaches 1), then $\hat{b}(M_i)$ approaches $b_{\max}$. For a correct model, as N increases, the expected value of R_i is N, and the expected value of $\log p(\chi^2 \le R_i | \mathrm{M}_i^{\mathrm{c}})$ is $\log(1/2)$, so the expected value of $\hat{b}(M_i)$ for a correct model is $b_{\max} + \log(1/2)$.

The scale for this metric of benefit is arbitrary, because the probability of accurate predictions is not a direct measure of the value of the improved control recommendations that occur with improved predictions. A more intuitive scale for our metric of benefit sets models that are not useful to have a benefit less than or equal to 0, and models that may be considered to have a benefit that is positive. We can adjust the scale of benefit by setting the value of the constant $b_{\max}$ in Equation 5.54. A value of $b_{\max} = \log(x)$ sets the expected value of a correct model to $\log(x/2)$, and sets the benefit of models that have a posterior probability greater than $1/x$ to a positive value. For example, setting $x = e^5$ gives $b_{\max} = 5$, and models with a posterior probability that is greater than $e^{-5} \cong 0.0067$ have a positive benefit.

5.6.2 Cost of a Model

The cost of a computation delay increases linearly with time, so we can measure the cost by assessing the value of the linear rate constant. The cost of a delay depends on the value (risk of adverse event) due to the current settings, and also on the change in value that would occur with improved settings. We can apply the value function to compute the current value from the current state of the system, but we are unable to assess the change in value that will occur with a more accurate model's recommendation before the model computes the recommendation.

That is, if a model under consideration is unable to compute control settings of higher value than the current settings, the cost of a delay is zero. The maximum cost of a delay occurs whenever a model under consideration computes settings with maximal value, and the model has the maximum benefit. The upper bound on the cost of a delay is

$$\hat{c}(M_i) = C_o k(X_o) t_i. \tag{5.55}$$

C_o reflects the time criticality of the cost function. The assessment of C_o is based on the risk that we are willing to accept per time interval to compute an accurate model, and on the benefit of the correct model. The assessment of C_o depends on the answer to the question, "For a given rate of adverse events, how much computation time should we expend to obtain accurate predictions?" Let this time be t_{max}, (and $t_{max} < t_{period}$). Setting the cost of t_{max} to the maximum benefit gives

$$C_o = \frac{b_{max}}{t_{max} k(X_o)}. \tag{5.56}$$

For example, we could assess the time that we are willing to delay implementing a new setting recommendation, t_{max}, in the presence of a risk per t_{period} of $k(X_o) = 0.01$. If we are willing to wait one-half of the time between setting adjustments ($t_{max} = 0.8\ t_{period}$) to find a model that has a b_{max} of 5, then $C_o = 625$.

5.6.3 A Metric for Dynamic Selection of Models

The estimates for cost and benefit of a model selection allow us to estimate the increase in value that an improved model may bring:

$$\Delta \hat{v}(M_i) = \hat{b}(M_i) - \hat{c}(M_i). \tag{5.57}$$

I call this estimate the *dynamic-selection-of-models metric*, which I refer to as the DSM, or the DSM metric. The DSM is the sum of measures of cost and of benefit of a model selection. The cost is a function of the expected time that the model will take to compute its recommendations, t_i. The benefit is the log prior probability that the model is adequate, plus the log probability that the fit of the model to the data would be as poor as observed if the model were correct. We can compute the DSM metric for the *i*th model as

$$\mathrm{DSM}_i = b_{\max} + \log p(\mathrm{M}_i^{\mathrm{A}}) + \log p(\chi^2 \le R_i | \mathrm{M}_i^{\mathrm{c}}) - C_{\mathrm{o}} k(X_{\mathrm{o}}) t_i, \tag{5.58}$$

which follows from substitution of Equations 5.54 and 5.55 into Equation 5.57.

We can compute an upper bound on DSM_i without fitting M_i to the observations, by assuming that the model fits the observations perfectly. In the case of a perfect fit, $R_i = 0$, and the DSM reduces to the $\mathrm{DSM}_i^{\mathrm{prior}}$:

$$\mathrm{DSM}_i^{\mathrm{prior}} = b_{\max} + \log p(\mathrm{M}_i^{\mathrm{A}}) - C_{\mathrm{o}} k(X_{\mathrm{o}}) t_i. \tag{5.59}$$

Adding $\log p(\chi^2 \le R_i | \mathrm{M}_i^{\mathrm{c}})$ to the prior DSM ($\mathrm{DSM}_i^{\mathrm{prior}}$) gives the *posterior DSM* (DSM_i). Because the logarithm of a probability is a negative number, the $\mathrm{DSM}_i^{\mathrm{prior}}$ sets an upper bound on the DSM_i for a model under consideration.

5.7 An Algorithm for Dynamic Selection of Models

The algorithm that I propose for the dynamic selection of models under time constraints is based on a search metric—the DSM—and on an organization of the models—the GoM (as described in Chapter 2).

5.7.1 Graph of Models

The algorithm assumes that the models are arranged as a GoM, in which a single, most detailed model—the base model—is at the top of the graph, and the least detailed, most simplified model is at the bottom. Directed arcs from each model are labeled with the additional simplifying assumptions that the adjacent models assert. The set of simplifying assumptions that each model asserts is the sum of the assumptions on any directed path from the base model to the model of interest. We saw the structure of a generic GoM in Figure 2.1, and, in Chapter 6, we shall instantiate that generic GoM for a set of models of cardiopulmonary physiology (see Figure 6.3).

5.7.2 Search Metric

The DSM is a search metric that defines the optimal model to select under a time constraint. The search algorithm cannot apply this metric exhaustively, however, because the computation time required to perform the model fitting that is needed to compute DSM_i accounts for the majority of the model-computation time t_i. The $\mathrm{DSM}_i^{\mathrm{prior}}$ suggests an initial model for testing—the model with the greatest $\mathrm{DSM}_i^{\mathrm{prior}}$. In addition, the $\mathrm{DSM}_i^{\mathrm{prior}}$ assists the search for models with a higher posterior DSM; the search algorithm can reject models for which the $\mathrm{DSM}_i^{\mathrm{prior}}$ is less than the current posterior DSM.

5.7.3 Search Algorithm

The algorithm has three steps, as shown in Figure 5.6:

1. *Initial model guess:* Search for the model that maximizes the prior search metric, $\mathrm{DSM}_i^{\mathrm{prior}}$.

2. *Model selection:* Search, within the time available, for the model that represents the best tradeoff of prediction accuracy and computation complexity.

   a. Fit the initial model guess to the observations to compute the model-selection criterion (DSM_i). If DSM_i is greater than zero, select the model and make a control-setting recommendation.

      i. If DSM_i is negative, search for an alternative model. Consider only models for which $\mathrm{DSM}_i^{\mathrm{prior}}$ is greater than the cost of the computation time expended so far, $C_o k(X_o)\, t_s$.

      ii. If the cost of the computation time expended exceeds $\mathrm{DSM}_i^{\mathrm{prior}}$ for all remaining models, then declare failure and go to step 3.

3. *Model refinement:* Continue the search to refine the model selection.

5.7.3.1 Step 1: Initial Model Guess

The first step—making an initial model guess—assumes that we have available a computationally inexpensive method to compute the prior search metric, $\mathrm{DSM}_i^{\mathrm{prior}}$, in a time much less than that required to compute DSM_i. In other words, the first step assumes that we have a method to compute the prior probability of model adequacy that takes far

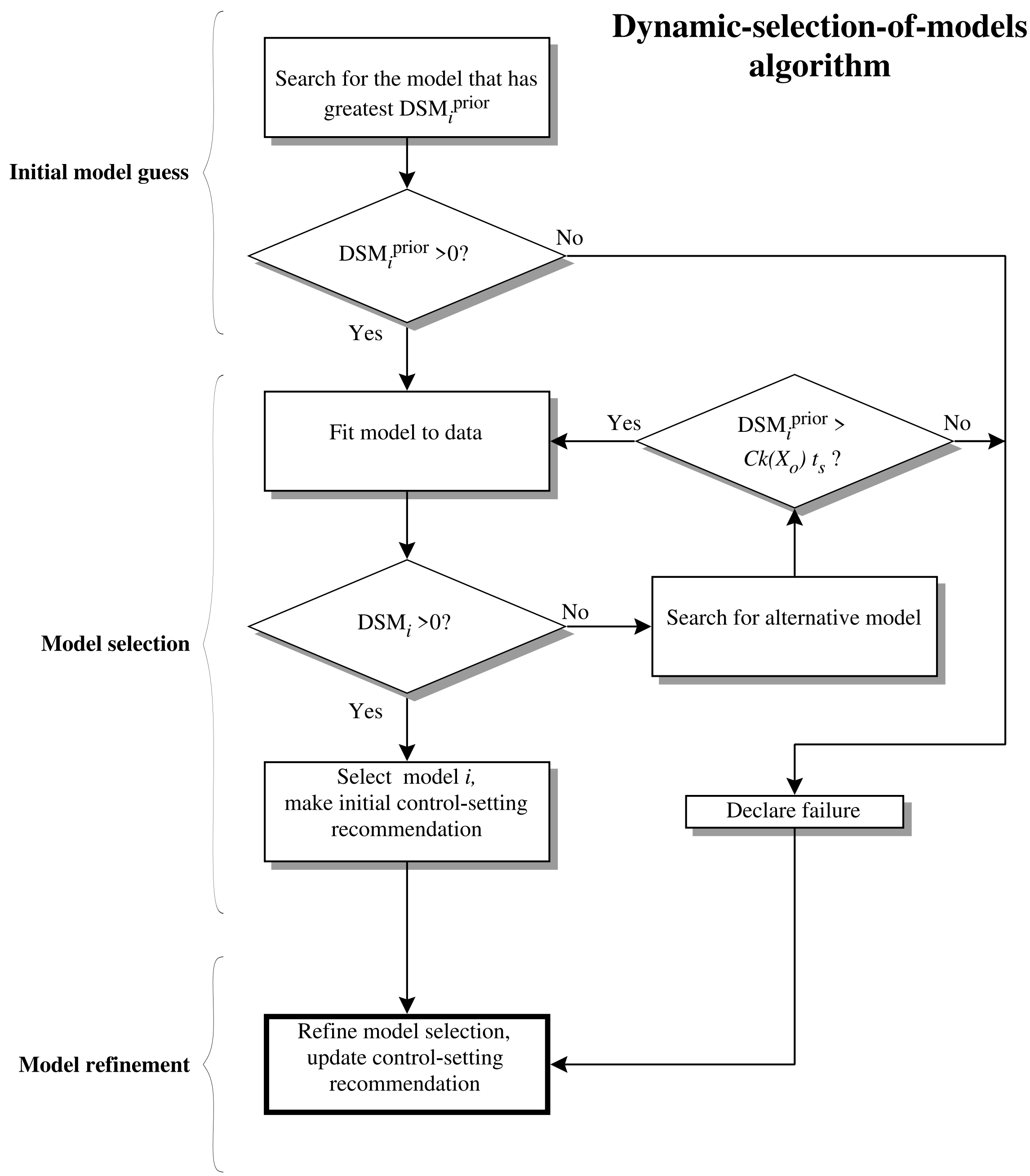

Figure 5.6 Dynamic-selection-of-models algorithm. Rectangles represent actions; the rectangle in heavy outline represents an action that terminates the algorithm. Figure 5.7 shows the final step (model refinement) in greater detail. DSM_i: dynamic-selection-of-models criterion (Equation 5.58); DSM_i^{prior}: prior estimate of DSM_i (Equation 5.59); $Ck(X_o)$: cost per unit of time; t_s: computation time expended during the model search.

less time than does the method to compute the best fit of a model to the data. If the total computation time to compute the DSM^{prior} of all models is a small fraction of the time available to search for an optimal model, then the search algorithm should perform an exhaustive search to find the model with the global maximum DSM^{prior}. Alternatively, if the total time to compute all DSM^{prior}s is large, the search algorithm can perform a local greedy search for the model with the greatest DSM^{prior} (starting the search at the least complex model).

5.7.3.2 Step 2: Model Selection

The model-selection step attempts to verify that the initial model guess has a DSM_i that is greater than 0. If the initial guess meets this criterion, then the algorithm applies this model for a control-setting recommendation, and then moves to the model-refinement step. If the initial model guess has a negative posterior DSM_i, then the algorithm performs a breadth-first search in the GoM for a model that has a positive DSM_i. The algorithm compares the DSM_i^{prior}s of alternative models to the cost of the time spent searching so far. As the computation time increases, this cost eventually exceeds the maximum possible benefit of any alternative model. If the algorithm fails to find a model for which the DSM_i is positive within the time available, the algorithm declares that it cannot find an acceptable model to select.

The maximum length of time that the algorithm has to find a model that has a positive DSM_i is the computation time of a model that has $p(M_i^A)$ of 1, R_i of 0, and DSM_i of 0. This maximum possible search time, t_{max}, is

$$t_{max} = \frac{b_{max}}{C_o k(X_o)}. \tag{5.60}$$

In practice, we set the maximum possible search time to a fraction of the time between changes in control settings, and then derive the value of the time constant, C_o, from that time (see Section 5.6.2).

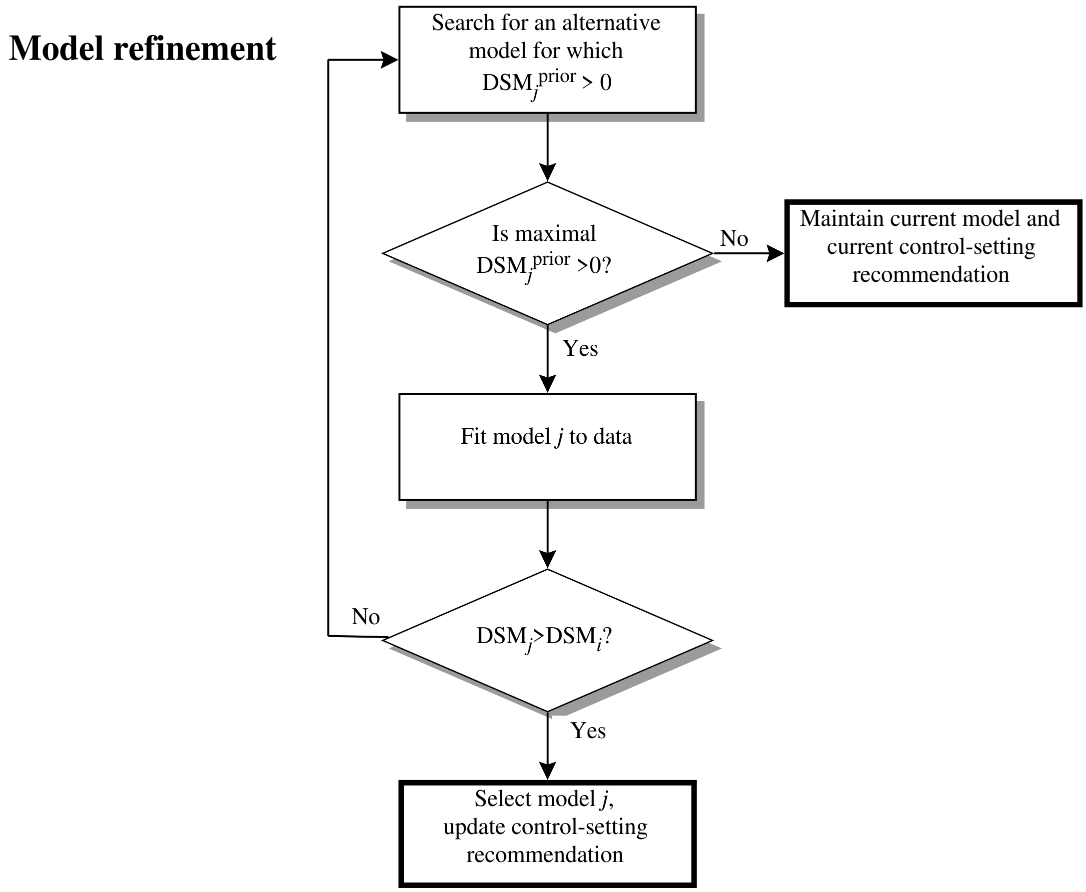

Figure 5.7 Model refinement in the dynamic-selection-of-models algorithm. Rectangles represent actions; rectangles with a heavy outline represent actions that terminate the algorithm. Diamonds represent decisions. DSM_i: dynamic-selection-of-models criterion (Equation 5.58); DSM_i^{prior}: prior estimate of DSM_i (Equation

5.7.3.3 Step 3: Model Refinement

After the algorithm selects a model and completes the model-based control-setting recommendation, or after the algorithm declares failure in the model-selection step, the algorithm moves to the model-refinement step. During the model-refinement step, the algorithm continues a local search for the model that has the maximum DSM_i, considering all models for which DSM^{prior} is positive.

The algorithm backtracks to consider all models for which the DSM_i^{prior} is positive, but that were not fitted to the data initially (because the DSMprior did not exceed the cost of the computation time expended at the moment the model was visited during the search). The algorithm terminates after evaluation of all models in the GoM that have a DSM_i^{prior} that is greater than 0.

5.7.3.4 Response to New Observations

When new observations of model variables become available, the algorithm repeats the model selection (step 2) using the current model as the initial model guess. If the current model has a positive DSM, then the algorithm retains this model selection and generates an updated control-setting recommendation. The algorithm then proceeds to model refinement (step 3).

5.7.3.5 Algorithm Characteristics

The model-selection algorithm depends critically on the assessment of prior probabilities of model adequacy. A low prior probability of adequacy for a model causes the algorithm to reject that model, even if the model explains all observations exactly. Rejection of models with a low prior probability of adequacy is the correct action if the knowledge on which the prior probability assessment relies is accurate, because the model may not explain new observations. If the prior probability is inappropriately low, then the algorithm may reject adequate models.

Similarly, if the prior probability of adequacy is inappropriately high for one or more models, then the algorithm may spend valuable computation time discovering that such models do not explain the observations. If many models that have incorrectly high DSM_i^{prior}s lie between the initial model guess and any model that is acceptable (that has a positive DSM), then the algorithm may fail to find an acceptable model within the available time. In the case of failure due to the time constraint, the algorithm moves to the model refinement step and continues to search for a model that has a positive DSM.

Whenever the prior probabilities of model adequacy are well defined, the dynamic-selection-of-models algorithm verifies that a model that we expect to be accurate is at least adequate.

5.8 Related Research

None of the prior research on AI and on statistical methods of model selection addressed the issue of the tradeoff of cost and benefit of computation (see the discussion in Sections 2.6 and 2.7). We discussed earlier, in Section 5.1, the manner in which the IDES system considered this tradeoff indirectly within a model-switching controller.

Guardian is a project that aims to develop a "comprehensive intelligent agent, having broad range of capabilities, to cooperate on the ICU team" [Hayes-Roth, 1992]. The Guardian system consists of a series of modules that are controlled within a blackboard architecture. This architecture supports multiple alternative modes of reasoning; Each module implements a different reasoning mechanism; these mechanisms include causal probabilistic models, qualitative models, simple quantitative models, and empiric situation–action rules. Guardian addresses the real-time constraint of decision making by establishing deadlines, and by attempting to compute a satisficing action before the deadline occurs. The blackboard architecture is opportunistic: As soon as any module has computed a satisficing action, Guardian implements that action. If no module has computed an action when the deadline occurs, Guardian executes the best action available at that time.

Before initiating the computation, Guardian makes no effort to evaluate the length of time that each module will take to compute a response, and thus does not reapportion the available computation resources to hasten the computation of an action from any one module. This approach makes sense if the high-level reasoner is unable to combine the computation resources of several modules to reduce the computation delay of the module or modules that are most likely to succeed before the deadline. In other words, additional high-level reasoning about the probability that a given module will compute a response before the deadline could avoid wasting computation resources during the critical period before the deadline and before any satisficing action is found [Hayes-Roth, 1994].

One of the modules in Guardian, called ReAct, implements an action-based hierarchy to respond to unanticipated critical events that may occur during a lengthy computation of earlier observations. Once a critical-event monitor identifies a critical event, the ReAct module guarantees that some response will be available before the real-time deadline [Ash, 1993]. The Guardian system implements multiple facets of behavior that an intelligent agent should have. Although Guardian is not designed to be a practical ICU monitoring tool, its authors expect that it will "... perform and coordinate a range of intelligent reasoning tasks of use in ICU monitoring" [Hayes-Roth, 1992].

Horvitz considered a problem that is similar to that of dynamic selection of models under time constraints—he addressed the problem of how to decide between computation and action for high-stakes, time-constrained decisions. He described a *model of rational action* that was based on decision-theoretic metalevel reasoning. These techniques guided an automated assessment of the value of additional computation, in comparison with the cost of additional delay in taking an action [Horvitz, 1989; Horvitz, 1990].

In Horvitz's formalism, the decision maker understood the structure of a decision problem, but the best action to take depended on an exact solution of the probabilities in a decision model. In certain time-constrained problems, the exact solution might not be computable within a short enough time to be useful, however, so Horvitz applied an approximate, anytime algorithm to solve the probabilities in the decision model. This algorithm had the property that, at any point in the computation, the solution was a set of bounds on the probabilities of the decision model. After halting the computation to inspect the results, a metalevel reasoner could decide to perform additional computation to provide a progressively more refined solution.

The benefit of additional computation was an increase in the expected value of the action that was recommended (recommended actions were based on the probabilities in the decision model). The cost of any additional computation was the risk that an adverse event (such as death) would occur during the additional computation delay. Horvitz's demonstrated his metalevel reasoning techniques in a program called Protos. In Protos, a criterion called the *expected value of computation* (EVC) guided an automated decision to stop computing and to recommend an action based on the approximate solution. Horvitz studied the behavior of Protos on hypothetical decision problems in medicine [Horvitz, 1989].

There is a strong similarity between the metalevel reasoning in Horvitz's model of rational action and the metalevel reasoning of the dynamic-selection-of-models approach. Both approaches implement approximate criteria (the EVC and the DSM) to direct reasoning about the tradeoff of cost and benefit of additional computation. In Protos, the metalevel actions are a series of decisions to perform additional increments of computation or to stop and to make a recommendation for a discrete action. The dynamic-selection-of-models approach does not assume that continuous refinement of the decision model is possible. Each alternative model represents a different computation cost and benefit (expected value of the corresponding model-based decision). Whereas the EVC assumes that additional computation will always refine the bounds on the probabilities in the decision model (and thus improve the decision), the dynamic-selection-of-models approach recognizes that certain computationally complex models may not provide any additional benefit.

This research and that of Horvitz describe the need for rational metalevel reasoning about the value of computation in model-based time-critical decision making. My approach (the dynamic-selection-of-models algorithm) makes discrete decisions regarding how much computation time to expend to compute a set of real-valued decisions (control settings); whereas Horvitz's approach (the model of rational action) makes continuous decisions regarding how much computation time to expend to compute a discrete-valued decision.

5.9 Summary

In this chapter, we have considered the application of model-based control methods to monitor and adjust the settings of a control system in the presence of computation-resource constraints. For automated methods that apply model predictions to compute optimal control settings, accurate models lead to better control settings; that is, more accurate models have a higher value than do less accurate models.

The value of a model results from an improvement in the system state that occurs after the model-based recommendation for the control settings are implemented. Under some conditions, computation-resource constraints limit the use of highly detailed prediction models. For time-critical decisions, the cost of computation delay may cause us to prefer less complex models that we expect to provide less accurate predictions of a system's behavior. The optimal model to select balances the cost of model-induced delay and the expected increase in value that the model's recommendation provides.

The cost of a delay is a linear function of time whenever the risk per unit of time that a system will make a transition to a less desirable state is constant and small. To compute the benefit of a model, we must incur the cost of the computation delay to fit the model to the observations and to measure how well the model explains the observations.

An algorithm for dynamic selection of models applies prior knowledge of the expected accuracy of each model (under the observed conditions) to make an initial estimate of the benefit of each model. The starting point for search for the optimal model is the model that has the highest net value (benefit minus cost). The algorithm fits the initial model to the data, then considers the possible increase in benefit and the expected cost of alternative models (1) to select the current model, (2) to fit an alternative model to the data, or (3) to declare that no model that can be computed within the time constraint has adequate prediction accuracy.

In Chapter 6, we shall discuss the implementation of the dynamic-selection-of-models algorithm to select, from a set of models of cardiopulmonary physiology, a patient-specific model that is suitable for use by a real-time VMA.

Chapter 6

Konan: Model Selection for a Ventilator-Management Advisor

The VentPlan prototype demonstrates that a VMA can apply a simplified quantitative prediction model of cardiopulmonary physiology to interpret ICU-patient observations, to monitor the ventilator treatments, and to make reasonable recommendations for adjustments to the ventilator (see Chapter 3). VentSim demonstrates that an expanded model of physiology represents a range of pathophysiologic states wider than that of the simplified VentPlan model; however, the computation complexity of VentSim is too great for the latter to be useful in a real-time ICU application.

In this chapter, I describe **Konan**[1], a program that applies the dynamic-selection-of-models algorithm that I described in Chapter 5 to search for patient-specific models that are suitable for use by a VMA. Konan implements a set of physiologic models that vary in computation complexity from that of VentPlan to that of VentSim. Konan searches for a model that

1. The name Konan is derived from the Greek *parakonan* (to sharpen), which is the root of the English word *paragon* (a model of excellence or perfection) [Mish, 1988]. The name has nothing to do with Arnold.

maximizes the tradeoff of model benefit (which derives from the model's prediction accuracy), and model cost (which is due to the model-induced computation delay).

In Sections 6.1 through 6.4, I describe the set of physiologic models from which Konan makes model selections, then the algorithm that Konan applies. Finally, in Section 6.5, I illustrate the behavior of Konan on data sets that correspond to common ICU-patient physiologic abnormalities.

6.1 From VentPlan to VentSim: Intermediate Models

Both the VentSim and the VentPlan models have components that represent the circulation, the airways, and the ventilator. VentSim represents all three components at a higher level of detail higher than that of VentPlan. As a result, the computation complexity of the VentSim model would impose a much greater computation-time delay, compared to VentPlan, before the treatment recommendations were available. If the VentSim model were implemented in a VMA, the model would impose a delay of greater than 100 times the delay that VentPlan's model causes.

VentPlan predicts the effects of changes in ventilator settings for certain ICU patients, but we do not expect that VentPlan would make accurate predictions for patients whose abnormality cannot be expressed by the structure of the VentPlan model. For example, VentPlan cannot represent the asymmetry of airway resistance that occurs in patients with unilateral partial obstruction of a mainstem bronchus.[2] The VentSim model can represent these patients more accurately, because the more complex model includes a multicompartment representation of the airways. In addition to the expanded airways component, the VentSim model includes a higher level of detail in the circulation component, which is not required to represent an airway abnormality. If we simplify the VentSim model by reducing the complexity of the circulation component, the resulting model is less computationally complex than is the VentSim model, but is nevertheless able to represent the airway abnormality. This *intermediate model* remains accurate for the predictions of airway pressures, of airflows, and of the distribution of airflows in the lung compartments.

2. A unilateral partial obstruction of a mainstem bronchus may occur due to inhalation of a foreign object, such as a peanut, or due to a growth in the lining of the bronchus (an endobronchial tumor).

For a patient with a specific physiologic abnormality, an intermediate model may make accurate predictions after a computation delay shorter than that caused by the VentSim model. We can build a set of intermediate models by simplifying each of the three major VentSim components, and then composing models that include varying combinations of the more and less detailed versions of each component. In Sections 6.1.1 through 6.1.3, I describe the simplifying assumptions that I applied to create the less detailed versions of each component of the VentSim model.

6.1.1 Simplified Circulation Component

The simplified airways component eliminates one of the two pulmonary-perfusion compartments of the detailed circulation component, and thus models gas exchange as a process that occurs between the airways component and a single pulmonary-circulation compartment (p1). This representation of gas exchange corresponds to the simplifying assumption that the distribution of $\dot{V}_A / \dot{Q}$ within regions of the lung is uniform. Both the detailed and the simplified circulation components also include a shunt compartment (s) and a tissue compartment (t), as shown in Figure 4.1.

6.1.2 Simplified Airways Component

The simplified airways component includes a single, series deadspace compartment (ds), and a single alveolar compartment (A1), and thus models ventilation as a uniform process throughout the portions of the lung that participate in gas exchange. This representation of ventilation corresponds to the simplifying assumption that the distribution of resistance and compliance within the perfused regions of the lungs is symmetric. The structure of this component is the same as that of VentPlan, as shown in Figure 4.1.

6.1.3 Simplified Ventilator Component

The simplified ventilator component computes the airway pressures, airflows and tidal volumes from the airway model directly, by assuming that the ventilation is sinusoidal, and that the airways respond in a manner similar to the behavior of an analogous electrical circuit. Figure 6.4 shows an electrical circuit that corresponds to the VentSim airways component. Each compartment of the airways is represented by a series resistor and capacitor. In the electrical circuit, electromotive force[3] is analogous to pressure, whereas current is analogous to airflow. The capacitance (C) of a capacitor is analogous to the compliance (C) of the airways component, because the accumulation of electrical charge (q) in proportion to electromotive force (E) is similar to storage of volume (V) in proportion to airway pressure (P). For the airways (lung) model, $V = C \times P$; for the electrical circuit, $q = C \times E$. Similarly, the flow of charge—the current—in the electrical circuit is analogous to the flow of air in the airways component.

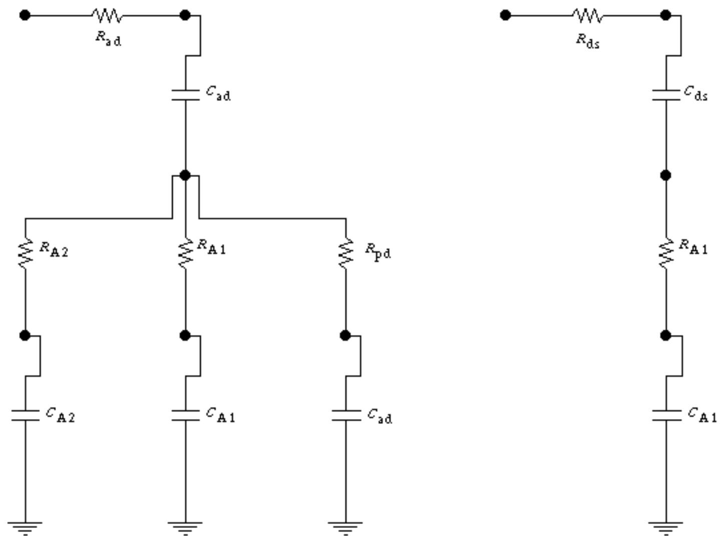

Figure 6.1 RC-circuit analogs of the airways components. The circuit on the left is analogous to the detailed airways component (as implemented in VentSim). The circuit on the right is analogous to the simplified airways component, which is similar to the airways model in VentPlan. Each compartment of the airways component is analogous to a resistor and capacitor that are connected in a series circuit. Abbreviations: *R*, resistance; *C*, capacitance; ad, anatomic deadspace; pd, physiologic deadspace; ds, total deadspace; A1, A2, alveolar airway compartments.

This analogy to electrical-circuit theory allows the simplified ventilator component to apply alternating-current circuit theory to compute the effects of changes in frequency of ventilation on distribution of ventilation [Sears, 1964]. For example, under the assumption of sinusoidal driving pressures, the minute ventilation of the first alveolar compartment in the detailed airway component is

3. Electromotive force (emf) is the electric potential, which is measured in Volts.

$$\dot{V}_{A1} = (V_{Tset} RR) \frac{k}{\sqrt{(R_{ad} + R_{A1}) \left(\frac{C_{A1}}{RR}\right)^2}}. \tag{6.1}$$

The variable names in Equation 6.1 are described in the legend to Table 5.1.

6.2 A Set of Physiologic Models

There are eight models that have one of two possible versions for each of the three model components, circulation, airways, and ventilator: {CAV, CaV, cAV, caV, CAv, Cav, cAv, and cav}. The upper- and lower-case letters in each model name indicate the level of detail of representation of the corresponding component. A capital letter indicates that the model incorporates the more detailed version; a lower-case letter indicates that the model incorporates the less detailed version. Thus, CAV (or M_{CAV}) is the model with the detailed version of all three components—the VentSim model; cav (or M_{cav}) is the model with the simplified version of all three components—the VentPlan model. The remaining models are intermediate in computation complexity and in size.

The set provides alternative models that vary in the expected accuracy and in the computation delay that they would cause if they were implemented in a VMA; it thus allows us to investigate the effects of model structure on model accuracy and on computation complexity.

Table 6.1 Model complexity for the set of physiologic models.

	CAV	cAV	CaV	caV	CAv	Cav	cAv	cav
Relative time per evaluation[a]	40	36	27	21	7.2	5.2	1.1	1
Number of fitted parameters	8	7	6	5	8	6	7	5

a. Times for a single model evaluation are reported by *getrusage*( ) under NextSTEP-3.2, running on a NeXTstation (Motorola 68040 processor with a clock rate of 25 MHz). Times are normalized to the time for a single evaluation of cav (0.05 seconds).

6.2.1 Computation Complexity of Intermediate Models

A VMA that incorporates a model that is more complex than the VentPlan model would make treatment recommendations after a longer computation delay. The amount of the increase in computation that occurs with when a model-based control application implements a more complex model depends on two factors: the computation time for a single model evaluation, and the number of model evaluations that the application performs to compute the model-based control action.

Table 6.1 and Figure 6.2 show the computation times of each physiologic model, relative to the computation time of the least complex model, cav. The numbers in this table are the relative times that the models require to predict the steady-state effects of a change in the control settings of the ventilator, when all patient-specific model parameters are specified. A single evaluation of the most complex model, CAV, requires 40 times as much computation time than does an evaluation of the least complex model, cav.

The models in this set vary in their computation complexity by a factor of 40 (from 0.05 to 1.96 seconds per model evaluation, for cav and CAV, respectively). To improve the model-solution times, we can find the steady-state solutions of all models by separating the solution of the slow time-constant components (the airways and ventilator) from the solution of the circulation component. The solution methods that each model allows are shown in Table 6.2.

The models that have a detailed ventilator component (– – V)[4] solve for the steady state of the ventilator and airway components by numeric simulation, and models that have the less detailed ventilator component (– – v) solve the ventilator and airway components analytically. That is, closed-form solutions for the ventilator–airway interaction reduce the computation complexity for the solutions of the models that include the simplified ventilator component.

Models that incorporate the less detailed circulatory component (c – –) find the solution to the steady-state circulation equations by a numeric method that searches for the roots of the steady-state equations. For oxygen transport, the steady state occurs when the oxygen uptake is equal to the oxygen consumption ($O_2uptake - O_2consumption = 0$). The models (c – –) search for a value of the oxygen concentration of the mixed venous blood that

4. The dashes in the model notation indicate any of the components in the corresponding position of the model name. That is, – – V represents all models that include the component V (models CAV, cAV, CaV, caV).

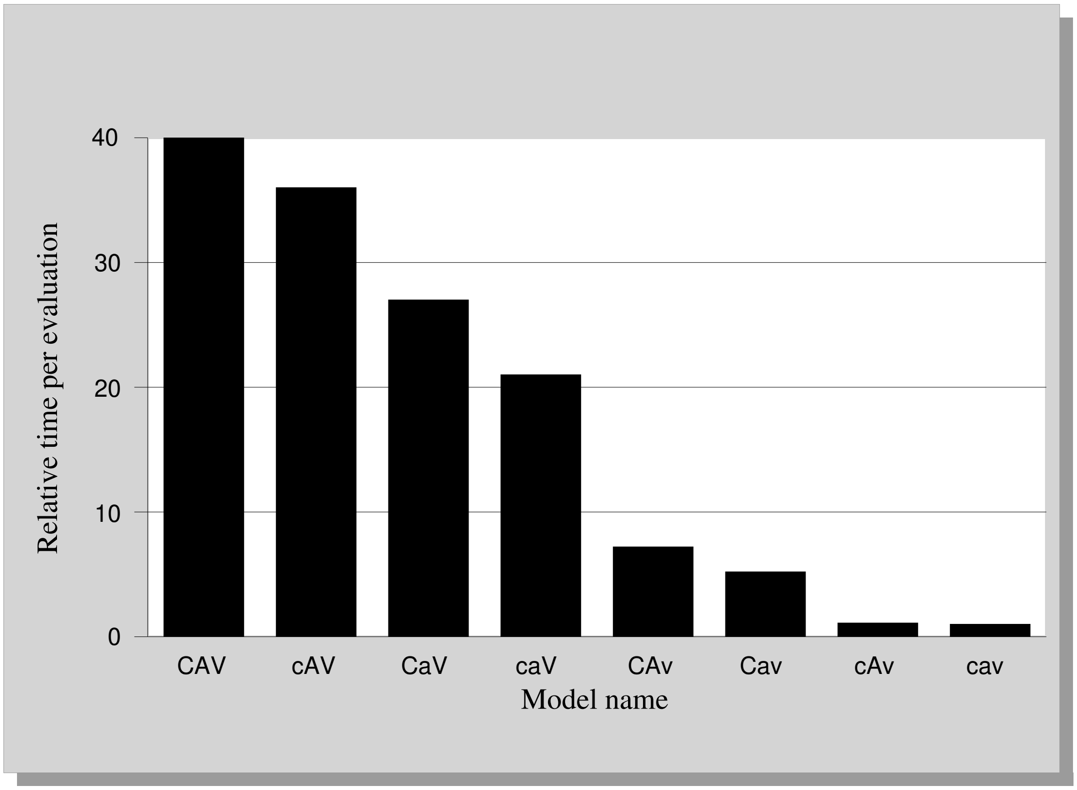

Figure 6.2 Model complexity for the Konan models. The time for a model to compute the effects of a change in ventilator settings when the model parameters are given is shown relative to the time required by the least complex model, cav. On a NeXT workstation, (Motorola 68040 processor/ 25 MHz), the time for each evaluation varies from 0.05 seconds for cav to 1.96 seconds for CAV. The model names show the level of detail for each of three model components; capital letters indicate greater detail, and lower-case letters indicate lesser detail. c, C: circulation; a, A: airway; v, V: ventilator. See also Table 6.2.

satisfies this equation for the single pulmonary-circulation compartment that receives ventilation.

The models that have the more detailed circulatory component (C – –) are more difficult to solve. At steady state, the sum of the oxygen uptakes in each of two compartments must be equal to the total oxygen consumption, but the proportion of the oxygen uptake (*f*) that occurs in each compartment is unknown. To find the steady-state solution for the oxygen model, these models nest a search for the oxygen concentrations that would occur for a specific value of *f* inside a search for the value of *f*. That is, the nested search finds the value of *f* that causes the solution for oxygen concentration in both ventilated pulmonary-circulation compartments to predict the same mixed venous oxygen concentration. The need to perform this nested search accounts for the increase in computation complexity of the models that have the more detailed circulation component.

Table 6.2 Solution methods for models in the Konan GoM.

Models	Ventilator / airway components	Circulation component
CAV , CaV	numeric simulation	nested search
cAV , caV	numeric simulation	search
CAv , Cav	analytic solution	nested search
cAv , cav	analytic solution	search

Because the more complex models have a greater number of fitted parameters (see Table 6.1), the parameter-estimation procedure may require a greater number of model evaluations to find the patient-specific parameter values than it requires for the less complex models. The relative increases in computation complexity that are shown in Table 6.1 are lower limits on the relative increases in model-based computation delay that the more complex models would cause if they were implemented in a VMA[5].

6.3 Konan

Konan is a program that I designed and implemented to investigate the application of the dynamic-selection-of-models algorithm (see Section 4.7) to the problem of model selection for a VMA. Konan organizes the set of cardiopulmonary physiologic models as a GoM; Konan applies the dynamic-selection-of-models algorithm to search the GoM for patient-specific models that would be suitable for use by a VMA.

The ideal model for a VMA is the model that results in a strategy that has maximal value. A model-based control application cannot compute the value of a strategy in time to be useful, because the value of a strategy depends on the results of computing the base model. However, a control application can approximate the value of a strategy from measures of the model benefit and the model cost. Konan estimates the benefit from the probability that a model has *adequate* prediction accuracy, and it estimates the cost of a model from the

5. For the Levenberg-Marquardt fitting algorithm, I estimate the number of model evaluations per iteration of the algorithm as the number of data points plus twice the number of fitted model parameters.

computation delay that the model causes (see Section 4.6). Konan searches the GoM for a model that maximizes the tradeoff of model benefit and model cost.

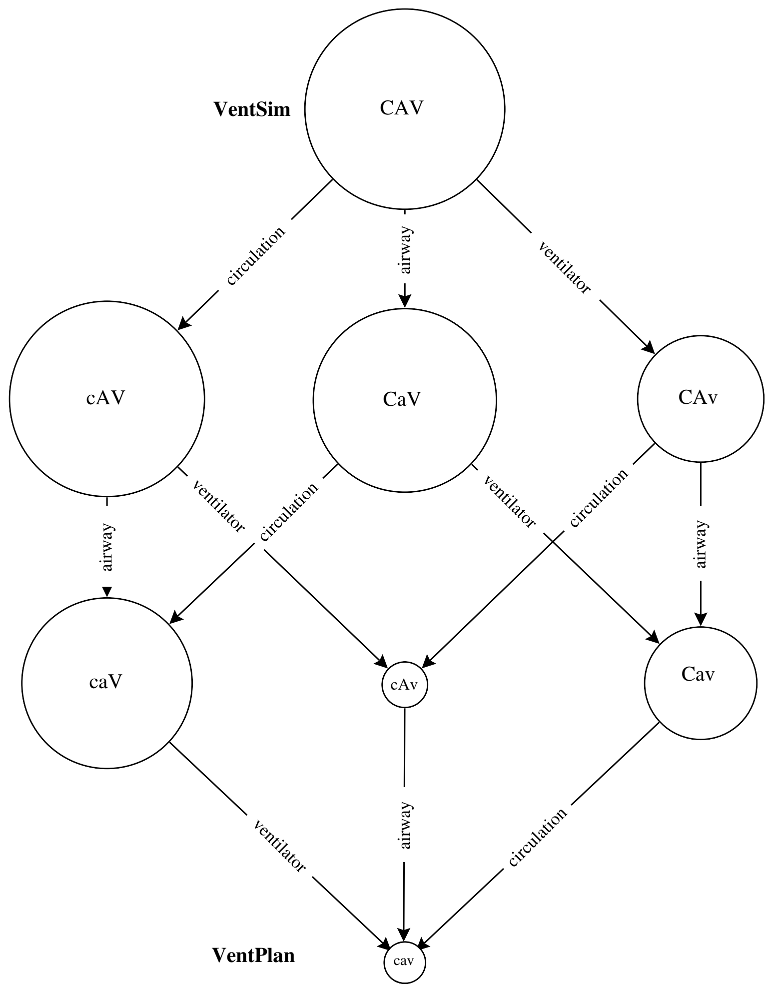

Figure 6.3 The Konan GoM. Nodes represent models that have circulation, airway, and ventilator components. The nodes are labeled with the names of the corresponding models, which have varying combinations of either expanded components (which are indicated by the upper-case letters C, A, and V) or simplified components (which are indicated by the lower-case letters c, a, and v). The arcs are labeled with the names of the simplifying assumptions that distinguish adjacent models. The diameter of each node is shown in proportion to the logarithm of the computation time of the corresponding model.

6.4 The Konan Graph of Models

The set of intermediate models in Section 6.2 includes models that assert all possible combinations of three simplifying assumptions. Konan organizes this set of models as a graph, as shown in Figure 6.3. In the Konan GoM, nodes correspond to the models, and adjacent models differ in a single assumption. Each arc between pairs of nodes is labeled with the simplifying assumption that distinguishes the corresponding pair of models. The GoM provides an organization of the models that makes explicit the assumptions that separate alternative models.

6.5 Dynamic Selection of Models in the Konan GoM

Konan implements the dynamic-selection-of-models algorithm to search for optimal models within the GoM, as described in Section 5.7 and in Figure 5.6. In this section, I describe how Konan implements each step of the algorithm.

The dynamic-selection-of-models algorithm searches for the model that maximizes the DSM search metric. The algorithm has three steps: initial model guess, model selection, and model refinement.

6.5.1 Initial Model Guess

- 1. *Search for the model that maximizes the prior search metric, DSM_i^{prior}.*

The initial model guess is crucial to the success of the heuristic search for an optimal model. The smaller the number of modeling assumptions that distinguish the initial model guess from the optimal model, the shorter the minimum path in the GoM from the initial selection to the optimal selection, and the shorter the computation time that Konan will expend to select a model. Under a time constraint, if the minimum path in the GoM from the initial model guess to an acceptable model is too long, then the algorithm will not discover the optimal model before the time constraint causes the search algorithm to halt.

The initial model to select is the model that has the highest estimate of prior value, $\Delta\hat{v}_{\text{prior}}(M_j)$. The estimate of prior value is based on the prior probability that a model is adequate and on the cost of the expected computation delay:

$$\Delta\hat{v}_{\text{prior}}(M_j) = b_{\max} + \log p(\mathrm{M}_j^{\mathrm{A}}) - C_{\mathrm{o}} k(X_{\mathrm{o}}) t_j. \tag{5.54}$$

In this equation, $b_{\max}$ is a constant, and $C_{\mathrm{o}}\, k(X_{\mathrm{o}})$ is constant for all models, given X_{o}, assuming that the patient state does not change during the interval of interest.

6.5.1.1 Model-Induced Computation Delay

In a model-based control application, the delay that is caused by the model depends on the computation time that the model requires per evaluation, and on the total number of evaluations that the model performs during parameter estimation and control-setting optimization.

The parameter-estimation procedure must search a continuous multidimensional hyperspace of dimension equal to the number of fitted parameters. By contrast, the control-setting optimization procedure searches the discrete space of alternative control settings, and the optimal setting is likely to be near the starting point for the search (the current setting). That is, the number of evaluations that are required for parameter estimation usually is much larger than the number of evaluations that are required for control-setting optimization.

Konan fits a model to the data by performing parameter estimation with the Levenberg-Marquardt method [Press, 1988], modified according to the method of Sheiner and colleagues [Sheiner, 1982]. The starting point for the fitting procedure is the vector of modes of the parameter prior distributions, which has length equal to the number of fitted parameters, p. The fitting procedure iteratively refines the parameter estimates to minimize a loss function—the weighted sum of squared model-prediction errors, plus the weighted sum of the differences between the parameter estimates and the modes of the parameter prior distributions. The loss function, evaluated at the parameter estimates, is a measure of the goodness of fit of the model to the set of observations and to the prior distributions. The shape of the loss function is a nonlinear p-dimensional surface that varies for each model and for each set of observations.

The exact number of model evaluations that the parameter-estimation procedure will require is fixed for each iteration of the procedure, but the number of iterations is difficult to compute in advance, because the models are nonlinear, and because the p-dimensional surface of the loss function is unpredictable. An application can set a maximum number of model evaluations, however, by setting a maximum number of iterations for the parameter-estimation procedure. This limit on the number of iterations for the parameter-estimation procedure sets an upper bound on the model-induced computation delay, but removes the guarantee that the parameter estimates will correspond to the maximum-likelihood estimates.

Konan cannot compute, in advance, the number of iterations that the search procedures will take during the estimation of system-specific parameters and during the search for recommended control settings. Konan sets the model-induced computation delay to the delay that corresponds to the maximum number of iterations of the search procedures. This maximum possible delay leads to an estimate of the cost of the model-induced delay that is an upper limit of the cost.

6.5.1.2 Prior Probability of Model Adequacy

The prior probability of model adequacy represents prior information regarding which models are likely to predict the behavior of the system from which the observations are taken. The models in the GoM are hierarchical; therefore, the models are not mutually exclusive, and the sum of the prior probabilities exceeds 1.

Several methods for computing $p(M_i^{\mathrm{A}})$ are possible:

1. Assume the prior probability of adequacy is fixed for each model. From an expert in physiology and in modeling, assess the $p(M_i^{\mathrm{A}})$ for each model, for the population of ICU patients who are treated with a ventilator.

2. Index the prior probabilities by the clinical diagnosis. That is, assess the $p(M_i^{\mathrm{A}})$ for each model, for the population of ICU patients who are treated with a ventilator and who have a specific clinical diagnosis.

3. Apply a rule-based expert system that evaluates multiple features of a patient's clinical state to compute $p(M_i^{\mathrm{A}})$. For example, a rule-based expert system might consider such features as the clinical diagnosis, the presence of diffuse infiltrates on a chest X-ray film, the value of the cardiac output, and the current setting of the FIO_2. The rule-based approach is analogous to the constraint-satisfaction methods implemented previously for model selection [Penberthy, 1987; Addanki, 1991; Weld, 1992; Falkenhainer, 1991; Nayak, 1992].

4. Apply a belief-network model to compute $p(M_i^{\mathrm{A}})$ from features of a patient's clinical state.

Konan implements methods 1 and 2. Konan applies a fixed distribution for the $p(M_i^{\mathrm{A}})$ that applies to the population of all ICU patients who are treated with a ventilator, and other distributions for the $p(M_i^{\mathrm{A}})$ that apply to subsets of the ICU population (subsets are defined by a clinical diagnosis). I combined knowledge of the pathophysiologic abnormalities that occur in various disease states with knowledge of the structure of the alternative models in the GoM, to assess subjective prior distributions for the probabilities of model adequacy. Table 6.3 shows these subjective estimates of the prior distributions on model adequacy for several clinical diagnoses.

Table 6.3 Distributions for the prior probability of adequacy for models in the Konan GoM.

Diagnosis[a]	$p(M^{\mathrm{A}}_{\mathrm{CAV}})$	$p(M^{\mathrm{A}}_{\mathrm{cAV}})$	$p(M^{\mathrm{A}}_{\mathrm{CaV}})$	$p(M^{\mathrm{A}}_{\mathrm{caV}})$	$p(M^{\mathrm{A}}_{\mathrm{CAv}})$	$p(M^{\mathrm{A}}_{\mathrm{Cav}})$	$p(M^{\mathrm{A}}_{\mathrm{cAv}})$	$p(M^{\mathrm{A}}_{\mathrm{cav}})$
Normal	1	1	1	1	1	1	1	1
Pulmonary edema	1	0.3	0.75	0.3	0.95	0.5	0.3	0.3
Asthma (severe)	1	0.75	0.7	0.25	0.9	0.5	0.6	0.2
Pulmonary embolus	0.9	0.8	0.25	0.1	0.8	0.2	0.15	0.1

a. The model parameters that correspond to the diagnoses are shown in Table4.3. ARDS: adult respiratory distress syndrome

6.5.2 Model Selection

- 2. *Model Selection: Search, within the time available, for the model that represents the best tradeoff of prediction accuracy and computation complexity.*

Konan fits the initial model guess, M_i, to the observations to compute DSM_i, as defined in Equation 5.58. If DSM_i is positive, then Konan selects the model, makes an initial control-setting recommendation, and then performs model refinement (step 3).

If DSM_i is negative, then the model has an unacceptable fit to the data, and Konan initiates a search in the GoM for an alternative model. Konan performs a search in two phases. In the first phase, Konan ranks candidate models according to their $\mathrm{DSM}_i^{\mathrm{prior}}$, then considers the models for which the $\mathrm{DSM}_i^{\mathrm{prior}}$ is greater than the cost of the computation time expended so far. In the second phase, Konan fits each candidate model in turn, until Konan finds a model that has a positive DSM_i, or until no additional models have. a DSM_i^{prior} that is greater than the current cost of time. The cost of time increases after each model fitting.

Konan considers all models that retract one assumption (these models are one edge up in the GoM) before considering models that retract more than one assumption. That is, Konan performs a breadth-first search of the GoM.

6.5.3 Model refinement

• 3. *Model refinement: Continue the search to refine the model selection.*

The initial model selection is the first model for which DSM_i is greater than 0. This model may not represent the best possible tradeoff of prediction accuracy and computation complexity, because the computation-time constraint prevents a full search of the GoM during the model-selection step. In addition, the models incur an ever-increasing cost-of-time penalty, which causes the algorithm to favor less complex and less accurate models as the search proceeds. Once Konan makes the initial model-based recommendations, Konan resumes the search of the GoM for the model that has the highest DSM_i.

Konan backtracks to reconsider all models that were not fitted to the data during the initial search.[6] Konan continues an exhaustive breadth-first search, fitting models for which the DSM^{prior} is positive, looking for the model whose DSM is maximal.

Figure 5.7 shows a diagram of the model-refinement step. The search continues until all models are examined, or until new observations warrant reevaluation of the current model and reassessment of the current recommendations for the control settings.

6.5.4 Analysis of Konan Algorithm

To test the dynamic-selection-of-models algorithm, I examined the performance of the algorithm on simulated-patient data. I generated the data as noisy observations of VentSim simulations, for which the VentSim parameters were set to values that correspond to patients with various physiologic abnormalities (the parameter values are described in Table 4.3). For each simulated patient, VentSim predicted the steady-state effect of five ventilator settings. The ventilator settings had varying FIO_2 (from 0.6 – 1.0), and varying

6. Some models that have a $DSM^{prior} > 0$ were not evaluated during the initial search, because the cost of computation time that was expended at the moment they were considered was greater than the DSM^{prior}.

RR (from 6 to 18). The V_{Tset} varied to maintain a constant minute ventilation of 72 dl/min.[7] The data set for each diagnosis contained observations of P_aO_2 and of mean P_{aw} at each ventilator setting. Figure 4.7 shows the predictions for the normal-patient and asthma-patient simulations.

To simulate noisy observations of the data, I added random errors to the VentSim predictions. The error terms, ε, were unbiased, normally distributed random variables, as computed by the formula

$$\varepsilon = \mu + \sigma\left(\sqrt{-2\log(u_1)}\right)\cos(2\pi u_2), \tag{6.2}$$

in which σ is the standard deviation, μ is the mean (bias) of the error term, and the variables u_1 and u_2 are independent uniform random numbers[8] between 0 and 1 [Ross, 1984]. For the experiments reported here, μ was 0, and σ was 2 percent.

6.5.4.1 Fitting of Models to the Simulated-Patient Data

The results of fitting the models in the GoM to the simulated patient data for the diagnoses *normal*, *pulmonary edema*, *pulmonary embolus*, and *asthma* are shown in Figure 6.4. This figure displays the information required to assess, after the fact, the model that would have been optimal to select at $t = 0$ (the optimal model is the model that represents the best tradeoff of cost of computation and of benefit). The diagonal line in each graph is the cost-of-time line. The linear constant for the cost of time, $C_o k(X_o)$, sets the cost of a delay to b_{max} for a computation time of 16 minutes.[9] As discussed in Chapter 4, the measure of benefit is

$$\hat{b}(M_i) = \log p(\mathrm{M}_i^{\mathrm{A}}) + \log p(\chi^2 \le R_i | \mathrm{M}_i^{\mathrm{c}}) + b_{max}. \tag{4.54}$$

To compute the benefits that are shown in Figure 6.4, I set the prior probabilities of all models ($p(\mathrm{M}_i^{\mathrm{A}})$) to 1, so that differences in benefit among the models are due entirely to the sum-of-squared residuals of the fits.

---

7. Ventilator settings were: {FIO_2, V_{Tset}, RR}={0.6,12,6}, {0.7,8,9}, {0.8,6,12}, {0.9,4.8,15}, {1,4,18}.
8. I applied the Mathematica function Random[] to generate these uniform random numbers.
9. As described in Section 5.6.2, $b_{max} = 5$, $C_o = 625$, $k(X_o) = 0.01$, $t_{period} = 20$ minutes.

The plots shows that there is wide variation in benefit and in computation times for the models in this GoM, and that models that represent the best combination of computation cost and benefit for one simulated-patient data set are not optimal for other data sets.

Models that have a positive DSM—models that have a benefit greater than their cost—appear as points above the cost-of-time line. Models that are in the upper left corner of each graph represent the most favorable tradeoff of accuracy and cost. For the normal data and the pulmonary-edema data, all models fit the data well (all models have a positive benefit). For the asthma data, only the models Cav, CAv, CaV and CAV fit the data well. For the pulmonary-embolus case, only the most complex models (CAv and CAV) fit the data well enough to have a positive benefit.

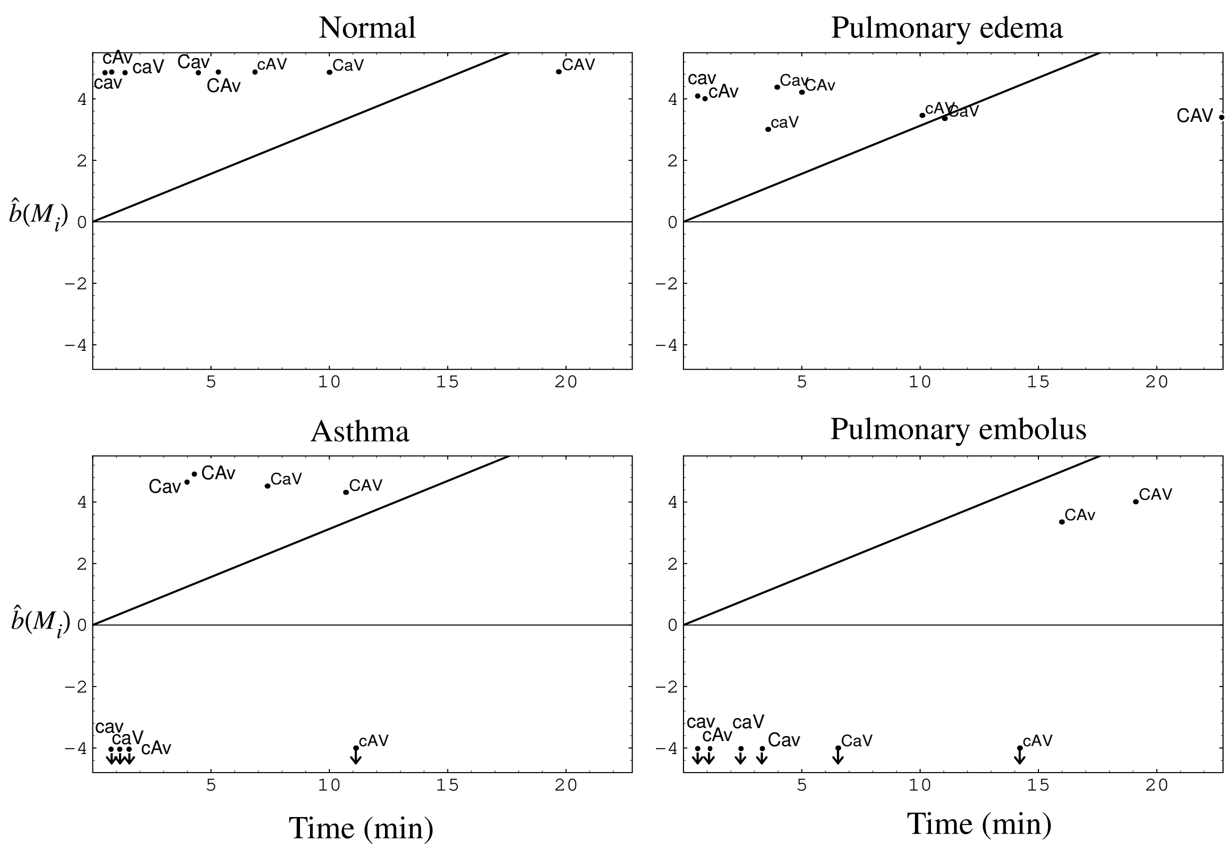

Figure 6.4 Model benefit versus model computation time. The benefit of each model in the GoM is plotted against the computation time that each model required to fit sample data sets, for data sets that correspond to each of four diagnoses. The diagonal line shows the cost of computation time. VentSim generated the data sets corresponding to each diagnosis using the parameter values that are listed in Table 4.3. $b_{max} = 5$, $C_o k(X_o) = 6.25$. The benefit is computed with the prior probability of model adequacy, $p(M_i^A)$, set to 1. Model names are defined in Figure 6.3. Models that are plotted with downward-pointing arrows have a benefit of less than –4.

Table 6.4 Dynamic-selection-of-models test results for normal data.

	CAV	cAV	CaV	caV	CAv	Cav	cAv	cav
$\mathrm{DSM}^{\mathrm{prior}}$	–4.90	–0.1	0.02	3.4	0.3	1.9	4.0	4.5
t_i (minutes)	19.7	6.8	10.0	1.4	5.3	4.5	0.8	0.6
R_i	5.1	4.8	5.3	5.0	4.6	5.1	4.8	5.0
DSM metric	–1.3	2.8	1.7	4.5	3.2	3.5	4.7	4.7

Table 6.5 Dynamic-selection-of-models test results for pulmonary-edema data.

	CAV	cAV	CaV	caV	CAv	Cav	cAv	cav
$\mathrm{DSM}^{\mathrm{prior}}$	–4.9	–1.3	–0.2	2.2	0.2	1.2	2.8	3.3
t_i (minutes)	28.0	10.1	11.0	3.6	5.0	3.9	0.9	0.6
R_i	13.2	13.2	13.6	14.9	9.8	8.9	10.7	10.4
DSM metric	–5.3	–0.9	–0.4	0.7	2.6	2.4	2.5	2.7

Table 6.6 Dynamic-selection-of-models test results for asthma data.

	CAV	cAV	CaV	caV	CAv	Cav	cAv	cav
$\mathrm{DSM}^{\mathrm{prior}}$	–4.9	–0.4	–0.3	2.0	0.21	1.2	**3.5**	2.9
t_i (minutes)	10.7	11.1	7.4	1.5	4.3	3.9	1.2	0.8
R_i	9.3	42.4	8.1	55.5	4.2	7.1	48.3	50.2
DSM metric	1.0	–10.7	1.9	–14.4	3.5	2.7	–10.3	–12.1

Table 6.7 Dynamic-selection-of-models test results for pulmonary-embolus data.

	CAV	cAV	CaV	caV	CAv	Cav	cAv	cav
$\mathrm{DSM}^{\mathrm{prior}}$	–5.0	–0.4	–1.4	1.1	0.1	0.3	2.14	2.2
t_i (minutes)	19.1	11.2	6.5	2.4	16.0	4.4	1.1	0.7
R_i	10.8	154.6	768.7	163.0	13.6	891.7	158.2	159.0
DSM metric	–8.2	–62.5	–360.7	–62.7	–1.6	–420.7	–60.1	–60.3

Details of the evaluation of each model on the four simulated-patient cases are shown in Tables 6.4 through 6.7. In this table, the $\mathrm{DSM}_i^{\mathrm{prior}}$ are based on the prior probabilities of model adequacy (as shown in Table 6.3), and on the expected computation times for the models (as shown in Table 6.8). The expected computation times are estimates of the maximum computation times that the models may require to fit the observations. These estimates assume that the parameter-estimation algorithm will require the maximum of 15 iterations. In practice, the Levenberg-Marquardt parameter-estimation algorithm usually finds parameter estimates after fewer than the maximum number of iterations—which explains why the computation times are, in all but one case, less than the expected computation times in Table 6.8.[10]

Table 6.8 Expected computation times for the set of physiologic models.

	CAV	cAV	CaV	caV	CAv	Cav	cAv	cav
Time to fit 10 data points (minutes)[a]	31.7	16.5	15.9	5.1	15.0	10.0	3.1	1.6

a. I estimate the number of evaluations required by a model per iteration of the Levenberg-Marquardt fitting algorithm as number of data points × twice the number of fitted parameters, and I assume a worst-case of 15 iterations of the fitting algorithm. See also discussion in Section 6.2.

6.5.4.2 Selection of Models

I applied the dynamic-selection-of-models algorithm to the four simulated-patient data sets to demonstrate the performance of the algorithm. Table 6.9 summarizes the results of the algorithm's performance.

For the normal and the pulmonary-edema data, cav was the initial model selection, because cav had the maximum $\mathrm{DSM}^{\mathrm{prior}}$ and had a positive DSM. For the normal-patient data, Konan fit five models during model refinement (which required 21.3 minutes of

10. Termination criteria for nonlinear parameter-estimation algorithms are something of a black art. See [Press, 1988] for a discussion. In Konan, the termination criteria for the Levenberg-Marquardt algorithm are (1) after 15 iterations, (2) after a minimum of two iterations, if R_i is less than one half the expected R_i for a correct model, (3) after at least seven iterations, if R_i improves by a minimal amount (less than 0.1%), (4) if there is no change in R_i for seven consecutive iterations.

computation) to determine that cav had the most favorable DSM. Similarly, for the pulmonary-edema data, Konan fit four models during model refinement (which required 13.4 minutes) to determine that cav had the most favorable DSM. For these two cases, the model refinement did not change the initial model selection. Although other models fit the data with a lower sum-of-squared residuals than did cav, no alternative model offered a better tradeoff of benefit and computation cost than did the least complex model, cav.

Table 6.9 Summary of performance of the Konan algorithm.

Simulated-patient diagnosis	Models fitted (initial)	Time to initial selection (minutes)	First model selected	Models fitted (refinement)	Time for model refinement (minutes)	Final model selection
Normal	cav	0.6	cav	cAv, Cav, caV, CAv, CaV	21.3	cav
Pulmonary edema	cav	0.6	cav	cAv, caV, Cav, CAv	13.4	cav
Asthma	cAv, caV, Cav	6.6	Cav	CAv, cav	5.1	CAv
Pulmonary embolus	cav, cAv, caV	4.2	none	Cav, CAv	20.4	none

For the asthma data, the model that had the highest DSM^{prior} (cAv) had a poor fit to the data, and thus had a negative DSM. Konan fit a total of three models (which required 6.6 minutes), to find a model that had a positive DSM (Cav). During model refinement, Konan fitted two additional models (which required 5.1 minutes) to find the model that had the most favorable DSM (CAv, see Table 6.9).

For the pulmonary embolus data, Konan was unable to make an initial model selection. Konan required 4.2 minutes to fit the initial model guess and one additional model. After fitting two models, the cost of the computation time exceeded the DSM^{prior} of all remaining models, and so Konan declared that it was unable to make an initial model selection due to running out of time during the search. During model refinement, Konan considered two remaining models for which $DSM^{prior} > 0$. Konan required an additional 20.4 minutes to fit these two models, but neither model had a positive DSM. Konan then declared that the most favorable model was CAv, but that CAv had a negative DSM. Although CAv had a reasonable fit to these data, the cost of CAv's computation-time delay was so high that CAv was rejected.[11] For these data, if the time-criticality factor, $C_o k(X_o)$, were smaller, then the cost of CAv's computation-time delay would be lower, and $\mathrm{DSM}_{\mathrm{CAv}}$ would be positive.[12]

For high values of the time-criticality factor, complex models—such as CAv and CAV—have a negative DSM, even when they fit the data well, because the cost of their computation time is so high. For example, in all cases shown in Figure 6.4, model CAV has a negative DSM, despite having a positive benefit (that is, low sum-of-squared residuals for the fits). Under severe time constraints, the rejection of this accurate but complex model is appropriate, because CAV would not allow a model-based controller to recommend actions within the time available.

The computation times for each model vary from data set to data set. These variations in computation time are due to two factors: (1) the varying number of iterations of the parameter-estimation procedure that the models require for different data sets, and (2) the varying number of iterations of the numeric simulation of the ventilator in models that have the detailed ventilator component. The lack of predictability of the computation-time requirement of fitting each model to the data means that, for certain models, the DSM, which is based on the observed computation time, may be greater than the $\mathrm{DSM}^{\mathrm{prior}}$, which is based on the expected computation time. For example, for the normal data set, model CAv has a $\mathrm{DSM}^{\mathrm{prior}}_{\mathrm{CAv}}$ of 0.32, whereas $\mathrm{DSM}_{\mathrm{CAv}}$ is 3.24 (see Table 6.4). The small positive value for $\mathrm{DSM}^{\mathrm{prior}}_{\mathrm{CAv}}$ indicates that the expected computation time has a cost that is comparable to the maximum benefit.[13] The relatively higher value of $\mathrm{DSM}_{\mathrm{CAv}}$ indicates that model CAv has a larger benefit than cost. In this case, the complex model found a good fit to the data after a computation time that was shorter than expected.[14]

11. $R_{\mathrm{CAv}} = 13.6$, $p(\chi^2 \leq R_{\mathrm{CAv}} \mid \mathrm{M}^{\mathrm{c}}_{\mathrm{CAv}}) = 0.19$).
12. For the pulmonary embolus data set, $\mathrm{DSM}_{\mathrm{CVa}}$ is positive for values of $C_o k(X_o) < 3.9$.
13. Expected computation time 15.0 minutes (actual 5.3 minutes), expected cost 4.6 (actual cost.
14. $\chi^2 = 4.63$, $t_i = 5.34$ minutes.

6.6 Discussion

The results of applying the Konan algorithm to simulated-patient data demonstrate that the algorithm is able to select models according to the DSM criterion, and that the algorithm performs a time-sensitive search for an initial model, followed by a more intensive search for the globally optimal model (as defined by the posterior DSM). In each case tested, the algorithm found the model that had a maximal DSM.

Konan attempts to find the optimal model to select under a time constraint, but makes an initial model selection as soon as it finds a model that satisfices—Konan selects the first model for which the DSM is positive. The number of models that Konan examines before it finds a model that satisfices depends on the starting point for the search within the GoM. Konan combines a knowledge-based method that provides an assessment of the prior probability of model adequacy for the alternative models with a statistical method that checks the ability of a proposed model to explain the observations. Konan performs time-constrained validation of models that are suggested by knowledge-based model-selection methods.

6.6.1 Estimate of a Model's Benefit

The estimate of a model's benefit that Konan implements, $\hat{b}$, is a measure of the log posterior probability that a model could have generated the known quantitative observations. This measure is an heuristic estimate of how accurately a model will make predictions; this estimate is a surrogate measure for the benefit of a model (the improvement in value of the control settings that a more accurate model computes).

In the absence of time constraints, the most complex model in the GoM is the model that has the highest benefit, because all other models are less accurate approximations of the most complex model. For the estimate of benefit of the most complex model to be highest, however, the model must have parameter values that correspond to the values of the system for which the model makes predictions. Unfortunately, the nondeterministic, nonlinear parameter-estimation procedure may not find the set of parameter values that cause the model to match most closely the observations. For example, in Table 6.8, compare the sum-of-squared residuals (R_i values) for Cav ($R_{\mathrm{Cav}} = 8.9$) and for CAV ($R_{\mathrm{CAV}} = 13.2$). Cav fit the observations more accurately than did CAV, even though CAV was the model that generated the observations. The parameter-estimation procedure was not able to find exactly the same values for CAV's parameters that generated the observations.

6.6.2 Estimate of a Model's-Computation Complexity

The time delay that the fitting of a model to a specific data set will cause is nondeterministic, and so the DSM^{prior} is uncertain—and the initial model guess may not be the optimal model, even if the initial guess fits the observations closely. There are at least two reasons why the model-induced time delay is uncertain.

The first reason is that the number of iterations that the parameter-estimation algorithm performs is uncertain. The exact number of iterations that the Levenberg-Marquardt parameter-estimation algorithm requires depends in a complex manner on the model, the observations, the starting values of the parameters, and the exact form of the algorithm's termination criteria (see also Section 6.5.4.1). The second reason is that the time per model evaluation may be uncertain. That is, the time that a model takes to simulate the steady-state effect of a single set of control settings and parameter values may vary. For example, in the Konan GoM, models that have the detailed ventilator component run a numeric simulation of the interaction of a ventilator with the airway component, which takes a variable number of cycles of the ventilator. The length of computation that this takes varies according to the values for the resistance and compliance of each compartment of the airways model component.[15]

Although the estimates of model-computation complexity are uncertain, Konan sets the estimates of the model computation times to worst-case values—to the values that correspond to the maximum number of iterations of the parameter-estimation procedure during model fitting. This conservative estimate of the possible computation delay prevents the model-search algorithm from becoming stuck in a lengthy computation that exceeds the time constraint for model selection. Unfortunately, the conservative estimate of computation delay may also cause the globally optimal model to be overlooked during the initial time-constrained search for a model that satisfices.

---

15. The ventilator-airways simulation continues until the cyclic change in airway volumes reaches a steady state; the number of cycles of the ventilator increases as the resistance increases and as the compliance of the airways increases, and as the longest RC-time constant increases.

6.7 Summary

Konan demonstrates the heuristic metalevel reasoning strategy described in Chapter 5, to select, from a set of alternative physiologic models, a model that represents the best tradeoff of quality of recommended ventilator control settings and cost of model computation. The set of models is organized as a graph (GoM), in which the alternative models vary from the least complex at the bottom (VentPlan model) to the most complex at the top (VentSim model). Intermediate models have varying combinations of VentPlan and VentSim components. Edges in the GoM are directed from the more complex models to the less complex models. Each edge is labeled with the simplifying assumption that allows the less complex model to incorporate a simplified version of one model component.

Konan implements the dynamic-selection-of-models algorithm; this algorithm performs a breadth-first search of the GoM, using a prior search metric (DSM^{prior}) to define an initial model guess, and then using the posterior search metric (DSM) to guide the search for a satisficing model (a model for which the estimate of the model benefit is greater than the cost of the observed computation time). After finding a satisficing model within the initial time constraint, Konan extends the search to identify the optimal model (the model that has the maximal DSM) within the GoM.

The DSM^{prior} is a criterion that is based on subjective disease-specific prior distributions and on estimates of the computation time for each model, whereas the DSM is based on the quality of the fit of the model to the observations, the observed computation time, and the prior distribution on model adequacy.

Konan demonstrates that the ability of the dynamic-selection-of-models algorithm to make an initial model selection is limited if the prior distributions on model adequacy suggest a starting model that is far from any satisficing model in the GoM. In this case, Konan may not find a satisficing model within the initial time constraint, but will find the optimal model during the model-refinement step. In the next chapter, we shall discuss future research to develop improved methods of computing the prior distributions on model adequacy.

Protos implemented a metalevel reasoning method to select the amount of a continuously variable computation resource to refine a decision-theoretic model that estimated the probability of alternative discrete system states [Horvitz, 1990]. In contrast, Konan implements a metalevel reasoning method to select among a set of models that predict a set of real-valued system variables. The results of both methods suggest that, in high-stakes, time-constrained decision environments, rational decisions for action should include a consideration of the tradeoff of the benefit of improved action that may result from additional deliberation, and the cost of that deliberation.

Chapter 7

Conclusions and Significance

In this dissertation, I have explored approaches to the automation of high-stakes decision making under time constraints, by considering the problem of how to compute appropriate adjustments for the settings of a mechanical ventilator, within the real-time constraints of ICU-patient care. In this chapter, I shall review the highlights of the dissertation, then discuss the significance of this work. Finally, I shall consider directions for future research.

7.1 Summary of Work Presented

In Chapter 3, we saw that the VentPlan prototype implemented a simplified, computationally efficient numeric model of cardiopulmonary physiology as the basis of a ventilator-monitoring and treatment-advice program. VentPlan predicted the effect of changes in ventilator settings on postoperative patients surprisingly well, but the simplified model in VentPlan was an appropriate match for these patients, who were selected for elective surgical procedures, and so did not have complex heart or lung abnormalities.

For some patients with complex physiologic abnormalities, simplified models do not provide accurate predictions, and only more detailed—and thus more complex—models are accurate. In Chapter 4, I developed a more detailed and more accurate model, called VentSim, that describes the interaction of the mechanical ventilator with patients who suffer from a wide range of abnormalities of cardiopulmonary physiology. Models, such as VentSim, that have greater complexity also have longer computation times, and cause a VMA to compute treatment recommendations after a longer computation delay. Models that make treatment recommendations after a long delay are not as valuable as models that make treatment recommendations immediately, because adverse events may occur during the computation delay. In other words, ICU-patient monitoring applications, such as a VMA, are *time-critical* applications that require models to make accurate predictions in the shortest possible computation time.

In Chapter 5, I discussed a formalism for describing time-constrained model-based control problems (such as the problem of controlling the settings of a ventilator in response to changes in a patient's physiology), and the assumptions that this formulation makes. Within the context of a model-based controller, I developed an analysis of the benefit of a model as a function of the state of the system (patient), the current control settings (ventilator treatment), and the model-based recommended control settings. In this formalism, the cost is a linearly increasing function of time, and the net value of a model selection is the benefit less the cost. I then noted that it is impossible to compute, in advance, the value of a model selection, because this value depends on the model-based control settings, and on the effect that those control settings will have.

To compute the value of a model selection, we must have information that invalidates any need for a model selection! Because we cannot predict the benefit of a model, I use an approximate measure for benefit—a measure that is based on the expected prediction accuracy of a model. Accurate models will have the highest benefit, so I use a measure of model-prediction accuracy as our measure of model benefit. I combine measures of prediction accuracy and measures of cost to create two metrics that guide a heuristic dynamic-selection-of-models search algorithm. This algorithm applies the prior search metric (the $\mathrm{DSM}^{\mathrm{prior}}$) to make initial model selections, then confirms a selection by fitting the model to the data to compute the posterior search metric (DSM).

In Chapter 6, I discussed an application of the dynamic-selection-of-models algorithm to the problem of model selection for a VMA. The implementation, called Konan, is based on a set of models that vary in their level of detail and their computation complexity, from the VentPlan model, which is too simplified for use in patients with severe or complex abnormalities, to the VentSim model, which is too computation intensive for use in the real-time ICU environment. The models in the set are composed from all combinations of lesser- and greater-detailed versions of three model components.

Konan organizes the set of models as a GoM, in which adjacent models differ by a single simplifying assumption, and applies the dynamic-selection-of-models algorithm to search for the model that maximizes the DSM metric. Konan begins a local search at the model that has the maximal DSM^{prior}, then selects the first model that satisfices. A model is satisficing if it has a benefit that is greater than 0 (which implies an adequate prediction accuracy) and a cost that does not exceed the benefit. After making an initial model-based recommendation for the control settings, Konan refines the model selection by searching for the model that has the maximal DSM. The Konan implementation demonstrates my hypothesis that an automated model-selection method can reason with a graph of hierarchically organized, structural models to find minimally complex models suitable for use in a control application, as I postulated in Chapter 1.

7.2 Contributions

The research described in this dissertation makes contributions in the fields of critical-care medicine, AI, statistics, and medical informatics.

7.2.1 Critical-Care Medicine

This dissertation demonstrates new methods for the application of physiologic models to support critical-care medicine. The implementation of VentPlan demonstrates the success of an ICU decision-support architecture that combines quantitative physiology models, semiquantitative belief-network models, and decision-theoretic multiattribute-utility models. In addition, VentPlan shows that (1) prior information about a patient's probable physiologic abnormalities is helpful to make a simulation model patient specific, (2) a patient-specific simulation model can provide *what-if?* simulations to the critical-care team, (3) a physiologic model provides a useful framework within which to interpret monitored physiologic variables, and (4) the combination of a preference model for treatments and a patient-specific simulation model allows for the automated generation of treatment advice for ICU patients who receive respiratory support.

The application of detailed simulation models to assist the care of patients in the ICU is hampered by two major factors. First, the many parameters of detailed physiologic models are underdetermined. That is, the quantitative observations that are available for patients in the ICU do not constrain the values of all parameters. VentPlan's architecture addresses this limitation by combining prior distributions on probable model parameters with the quantitative observations to compute patient-specific model parameters. The second factor is the requirement for real-time decision making for ICU patients. The dynamic-selection-of-models algorithm provides a method to implement more complex physiologic models that meet this real-time constraint. Rather than considering the time-constraint as an absolute time limit, the dynamic-selection-of-models method evaluates the tradeoff of the level of expected model-prediction accuracy and the magnitude of the computation-resource expense of each model.

7.2.2 Artificial Intelligence

Previous work by researchers in AI on automated modeling has focused on engineering models of physical systems. My work extends previous methods of automated modeling in several ways.

The traditional AI approach to model selection is based primarily on prior knowledge of the system that is to be modeled. My work demonstrates a method to combine prior information about model accuracy with the information in quantitative observations to select models that take into account all available information about model-prediction accuracy.

Konan selects appropriate models of a complex system (the ICU patient) for use in a high-stakes, time-critical control application, and thereby shows that automated selection of a model that balances computation-resource constraints and model-prediction accuracy is possible.

The time constraint on decision making is compelling in the domain of ICU care, yet *some* form of computation-resource constraint is present in all computer-based decision-support methods. The dynamic-selection-of-models algorithm, and the DSM metric, provide a rational basis on which to assess the tradeoff of computation-resource constraints and model-prediction accuracy in high-stakes, time-critical control applications.

Whereas prior work on automated selection of models developed methods to select qualitative models of minimally complex physical systems, Konan applies the GoM organization to a set of computation-intensive differential-equation models of a highly complex, nonlinear "real-world" biological system.

Previously, AI programs attempted to automate model selection by searching for models that met *absolute*, or categorical, constraints on model accuracy and on computation complexity. The dynamic-selection-of-models algorithm, and the DSM, provide a method to assess the balance of computation-resource expense and model accuracy as a continuous tradeoff. The DSM metric should be useful whenever a computation-resource constraint limits the complexity and the prediction accuracy of a model that is used in a model-based control application.

7.2.3 Statistics

The traditional statistical view of model selection is based purely on model performance (model-prediction accuracy). The problem of model selection for a VMA points out that prior information regarding which models are likely to make accurate predictions may provide important distinctions among models.Whenever the observations of a system underdetermine the parameters of system models, the prior information about the system allows simpler and less accurate models to be rejected, even before quantitative observations prove, in a statistical sense, that a more complex model is required to predict the system's behavior. The DSM metric demonstrates the usefulness of combining an assessment of the prior probability of model accuracy with a measure of goodness of fit, as a basis for assessing the prediction accuracy of alternative models.

7.2.4 Medical Informatics

Medical informatics is a broadly interdisciplinary field. The need to develop interdisciplinary approaches to solving difficult problems is the reason that medical informatics exists as a separate field of study. In this dissertation, we have seen that

VentPlan brings together model-based control theory with quantitative physiologic model, a belief-network model, and a decision-theoretic model, to create a real-time model-based controller of a complex nonlinear system. Similarly, Konan builds on techniques from AI, statistics, decision theory, simulation modeling, and control theory to solve part of the modeling problem for an automated ICU program that performs model-based monitoring. Both VentPlan and Konan demonstrate the value of applying interdisciplinary approaches to solve difficult problems.

7.3 Future Research

In this section, I shall discuss possible directions for future research to develop improved methods of model selection under computation-resource constraints. Then, I shall consider how researchers could apply the results of this dissertation to create programs to assist clinicians to care for ICU patients.

7.3.1 Automated Model Selection

The limitations of the dynamic-selection-of-models method suggest directions for future research.

7.3.1.1 Parameter-Estimation Procedure

The parameter-estimation procedure is a multidimensional nonlinear optimization that is not guaranteed to find the globally optimal values for the set of fitted parameters. For example, in Table 6.6, compare the fit of the detailed model, M_{CAV}, with the fit of the less-detailed model, M_{CAv}, to the simulated asthma-patient data. These data were generated by the M_{CAV} model, so the R_i should be the lowest for this model. However, the parameter-estimation procedure found a local minimum for the set of fitted parameters, and, although the fitted parameters corresponded to an acceptable fit of the model to the data, the parameter-estimation procedure did not find the optimal set of parameter values that corresponded to the parameters that generated the data.

The performance of the parameter-estimation procedure depends—among several factors—on the starting point for the search. In VentPlan and in Konan, the set of prior parameter distributions varies according to the diagnosis. If the diagnosis is in error, then the starting point for the parameter estimation may be far from the globally optimal values, and the search may fail to find an acceptable fit of the model to the data. One possible approach to alleviating this problem is to consider multiple starting points for the search, if the initial parameter estimation fails to find acceptable parameter values. For example, the starting point for the search could be the modes of the prior parameter distributions that correspond to alternative diagnoses.

7.3.1.2 Prior Probability of Model Adequacy

As I discussed in Section 6.5, the search for an initial model to select depends strongly on the starting point for the search in the GoM. The starting point for the search is based on the prior probability of model adequacy for the set of models. In the implementation of Konan, these prior distributions were based on a table lookup for each diagnosis. A more general method to compute prior distributions on model adequacy would include an explicit model of how knowledge of a system affects the prior probability of adequacy for each model.

For example, Figure 7.1 shows a proposed structure for a belief-network model to compute the distribution of prior probabilities for the Konan GoM. This belief network takes as inputs the clinical assessment of the physiologic diagnosis and computes a prior probability of model adequacy for the set of alternative models.

7.3.1.3 Metalevel Control of Computation

The dynamic-selection-of-models algorithm is a heuristic metalevel strategy to control the computation resources that a VMA applies to solve the problem of time-constrained data interpretation and optimization of control inputs. The algorithm does not consider the possibility that further search will consume additional computation resources but lead to no improvement in model accuracy. For example, the delay caused by the model-selection algorithm leads to a less desirable outcome whenever the algorithm fails to find an acceptable model.

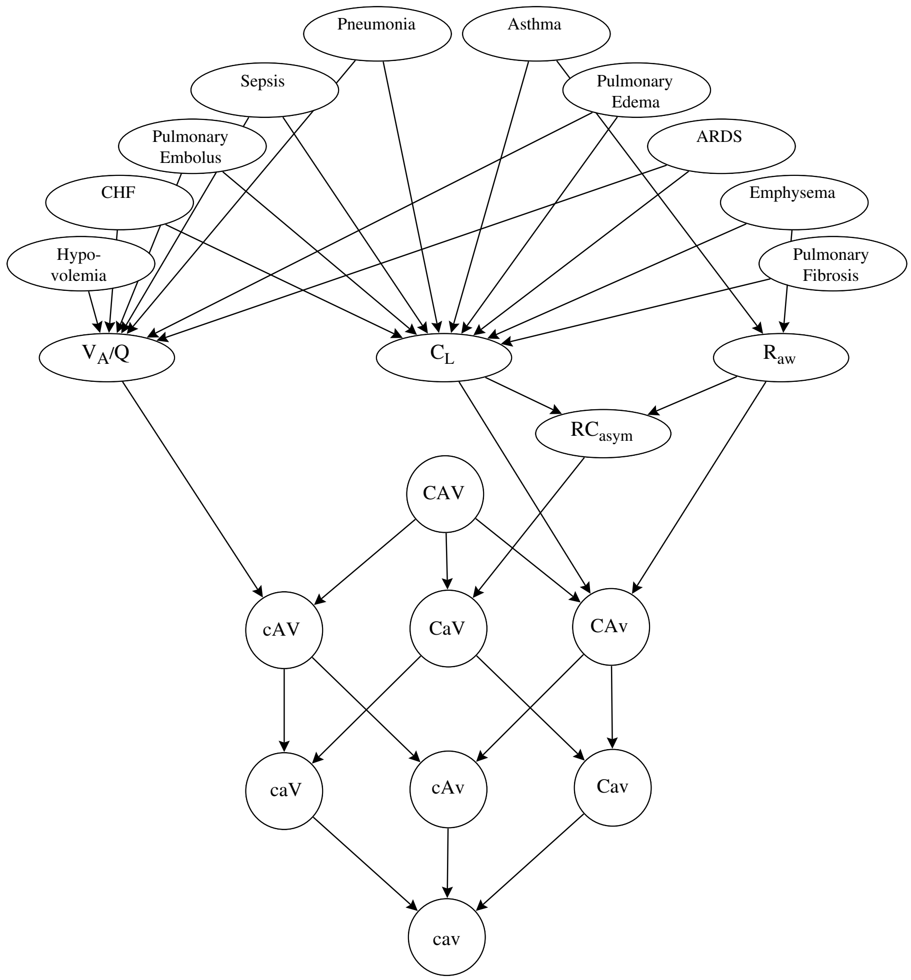

Figure 7.1 Proposed design for a belief network that computes the prior probability of adequacy for models in the Konan GoM. The eight circular nodes at the bottom of the network are TRUE–FALSE nodes that correspond to the models in the Konan GoM. The probability that a model node is TRUE is the probability that the corresponding model in the GoM is adequate. The top-level nodes are TRUE–FALSE nodes that correspond to diagnoses; these nodes are used as inputs to the network when one or more diagnoses are known. Middle nodes are multivalued nodes that correspond to parameters of the physiologic state. CHF: congestive heart failure; ARDS: adult respiratory distress syndrome; V_A/Q: ratio of lung ventilation to perfusion; R_{aw}: mean airway resistance; C_L: mean static lung compliance; RC_{asym}: degree of asymmetry of the resistance-time constants of the lung. Labels for the model nodes correspond to the models in Figure 6.3.

We might wish to supervise the metalevel strategy by implementing a method to analyze each decision to evaluate a model, before committing the computation resources to the evaluation. For example, we could implement the fitting approximation method to ensure that the differences in model structure that are present in an alternative model are likely to improve the fit of the model to the data [Weld, 1992].

We could develop higher and higher levels of supervision of the metalevel strategy, but this process of “thinking about thinking,” or *analytic regress*, rapidly reaches the point of no additional benefit. Previous researchers considered the problem of analytic regress to decide when to stop reformulating the structure of an inference model [Breese, 1990].

7.3.2 VentSim

The VentSim simulator, and the VentSim interface for *what-if?* simulations, may be useful to clinicians who care for ICU patients. In the current prototype, the clinician must set the physiologic parameters of the model to values that correspond to a patient, before VentSim simulates the effect of alternative control settings.

A simple improvement to this interface would lead to a patient simulator that would be of greater direct use. An extension to VentSim would apply the Konan parameter-estimation procedure to assess automatically the model parameters for a specific patient. The clinician would be able to simulate the effects of alternative control settings of the ventilator that were based on patient-specific parameter values. The interface also would allow the clinician to modify the model parameters to investigate the effect of alternative interpretations of the patient’s physiology on the model predictions.

Additional improvements to the VentSim model are possible. For example, we could expand VentSim to include a model of the effect of hypoxic pulmonary vasoconstriction, and to include a more realistic (nonlinear) pulmonary-compliance relationship.

7.3.3 Ventilator-Management Advisor

VentPlan is a prototype ventilator-management assistant (VMA). A next-generation VMA would incorporate the models of the Konan GoM and the dynamic-selection-of-models algorithm, to create a VMA that modified its model in response to the time criticality of a patient’s treatment decision. In Chapter 1, I showed an example of what an interface to such a VMA might look like.

Given a working VMA, we could compare the relative performance of model-based control systems and protocol-based approaches. For example, the protocol for ventilator care of ARDS patients provides instructions for adjustments to the ventilator. Differences between the VMA recommendations and the protocol recommendations either would reveal flaws in the decision model that the VMA implements, or would indicate inconsistencies in the definition of the protocol.

The architecture of a VMA guarantees that the VMA-based treatment recommendations are consistent with the predictions of the physiologic model and with the preferences of the decision model. If we adjust the decision model to reflect the preferences that are expressed in a protocol, then the VMA provides a method to verify the consistency of the protocol. That is, we can compare the recommendations of a protocol with the recommendations of the VMA, for a wide variety of simulated or observed patient behavior. This ability to test a protocol would be helpful to ensure that a protocol leads to intended actions under a wider variety of extreme, or infrequently encountered, physiologic circumstances.

7.4 Conclusions

I developed a method to implement mathematical models of complex systems within an automated, model-based control program to demonstrate that my hypothesis is valid: *An automated model-selection method can reason with a graph of hierarchically organized, structural models to find minimally complex models suitable for use in a control application*. I assumed that the complexity of highly detailed models that might be considered would exceed the computation-time available for computing model predictions. I developed the dynamic-selection-of-models method for use in model-based control applications, in which the consequences of model-generated control actions provided a framework for evaluating the adequacy of the model selections. The method also should be useful in other modeling tasks for which a goal is to find a minimally complex model.

Appendix

Abbreviations and Symbols

ad: anatomic-deadspace compartment
AI: artificial intelligence
AIC: Akaike information criterion
ARDS: adult respiratory distress syndrome
ARMA: autoregressive moving-average
A1, A2: alveolar-ventilation compartments
$b(M_i)$: benefit of ith model
BIC: Bayesian information criterion
CAV, CAv, CaV, cAV, Cav, cAv, caV, cav: Model names for the Konan GoM. The letters correspond to model components: c,C: circulation; a,A: airway; v,V: ventilator. Capital letters indicate that a model includes a more detailed version of the corresponding component.
cm, cms: centimeters
$c(M_i)$: cost of ith model
CMV: constant mandated volume
C: compliance (of an airway compartment)
C_{O_2}: concentration of oxygen
C_{CO_2}: concentration of carbon dioxide
CO_2: carbon dioxide
dl: deciliters
ds: deadspace

DSM: dynamic-selection-of-models
DSM_i: DSM metric for the *i*th model
DSM_i^{prior} :prior DSM metric for the *i*th model
EVC: expected value of computation
f: fraction
FIO_2: fraction of inspired oxygen
GoM: graph of models
Hb: hemoglobin
Hg: mercury
IC: information criterion
ICU: intensive-care unit
KIC: Kashyap information criterion
L_i^{max} : maximum likelihood
MAP: maximum a posteriori probability
M_i: *i*th model
M_o: base model
MBIC: modified Bayesian information criterion
MDL: minimum description-length criterion
MIGET: multiple inert-gas-elimination technique
mm Hg: static pressure, in millimeters of mercury
N: sample size
O_2: oxygen
ODE: ordinary differential equation
P_{aw}: ventilator pressure at the mouth
pd: physiologic-deadspace compartment
PEEP: positive end-expiratory pressure
PLC: penalized likelihood criterion
$p(M_i^c)$: probability that the *i*th-model is correct
$p(M_i^A)$: probability that the *i*th-model is adequate
P_{O_2}: partial pressure of oxygen
P_{CO_2}: partial pressure of carbon dioxide
p1, p2: pulmonary-perfusion compartments
$\dot{Q}$: blood flow (perfusion)
$\dot{Q}_T$: total blood flow (cardiac output)
$\dot{Q}_s$: shunt
Θ_i: vector of *i*th-model parameters
R: resistance
R_i: sum of squared residuals for fit of *i*th model
RR: respiratory rate (setting of ventilator in CMV or IMV mode)
RQ: respiratory quotient
STP: standard temperature and pressure

$\dot{V}_A$: ventilation
$\dot{V}_A / \dot{Q}$: ventilation to perfusion ratio
V_{ds}: deadspace volume
VMA: ventilator-management advisor
VF: ventricular fibrillation
$\dot{V}_{O_2}$: metabolic rate (oxygen consumption)
VPnet: VentPlan belief network
V_T: tidal volume
V_{Tset}: tidal volume, as set on the ventilator controls
X_o: vector of ventilator-control settings
y: model prediction
Y: vector of model predictions

Bibliography

[Addanki, 1989] Addanki, S., Cremonini, R. and Penberthy, J.S. Reasoning about assumptions in graphs of models. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, pp. 1432–1438, Palo Alto, CA: Morgan Kaufman, 1989.

[Addanki, 1991] Addanki, S., Cremonini, R. and Penberthy, J.S. Graphs of models. *Artificial Intelligence*. **51**:145–178, 1991.

[Agogino, 1992] Agogino, A.M., Nour-Omid, O., Imaino, W. and Wang, S.S. Decision-analytic methodology for cost-benefit evaluation of diagnostic testers. *IIE Transactions*. **24**:39–54, 1992.

[Akaike, 1973] Akaike, H. Information theory and an extension of the maximum likelihood principle. In Petrox, B.N. and Caski, F. (ed), *Second International Symposium on Information Theory.* Budapest: Akademiai Kiado, p. 267, 1973.

[Akaike, 1979] Akaike, H. A Bayesian extension of the minimum AIC procedure of autoregressive model fitting. *Biometrika*. **66**:237–242, 1979.

[Andersen, 1989] Andersen, S.K., Olesen, K.G., Jensen, F.V. and Jensen, F. HUGIN – a shell for building Bayesian belief universes for expert systems. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, pp. 1080–1085, Palo Alto, CA: Morgan Kaufman, 1989.

[Ash, 1993] Ash, D., Gold, G., Seiver, A. and Hayes-Roth, B. Guaranteeing real-time response with limited resources. *Artificial Intelligence in Medicine*. **5**:49–66, 1993.

[Ashby, 1963] Ashby, W.R. *An Introduction to Cybernetics.* New York, NY: John Wiley and Sons, 1963.

[Atkinson, 1980] Atkinson, A.C. A note on the generalized information criterion for choice of a model. *Biometrika*. **67**:413–418, 1980.

[Bai, 1992] Bai, J., Ying, K. and Jaron, D. Cardiovascular responses to external counterpulsation: A computer simulation. *Medical and Biological Engineering and Computing*. **30**:317–323, 1992.

[Ballester, 1990] Ballester, E., Roca, J., Ramis, L., Wagner, P.D. and Rodriguez–Roisin, R. Pulmonary gas exchange in severe chronic asthma: Response to 100% oxygen and salbutamol. *American Review of Respiratory Disease*. **141**:558–562, 1990.

[Barnea, 1992] Barnea, O., Smith, B.T., Dubin, S., Moore, T.W. and Jaron, D. Optimal controller for intra-aortic balloon pumping. *IEEE Transactions on Biomedical Engineering*. **39**:629–634, 1992.

[Barnea, 1993] Barnea, O. and Sheffer, N. A computer model for analysis of fluid resuscitation. *Computers in Biology and Medicine*. **23**:443–454, 1993.

[Barosi, 1985] Barosi, G., Cazzola, M., Berzuini, C., Quaglini, S. and Stefanelli, M. Classification of anaemia on the basis of ferrokinetic parameters. *British Journal of Haematology*. **61**:357–370, 1985.

[Beinlich, 1989] Beinlich, I.A., Suermondt, H.J., Chavez, R.M. and Cooper, G.F. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *Second European Conference on Artificial Intelligence in Medicine*, London, UK, pp. 247–256, Berlin, West Germany: Springer-Verlag, 1989.

[Bhansali, 1977] Bhansali, R.J. and Downham, D.Y. Some properties of the order of an autoregressive model selected by a generalization of Akaike's FPE criterion. *Biometrika*. **64**:547–551, 1977.

[Box, 1985] Box, G.E. *The Collected Works of G.E.P. Box.* Belmont, CA: Wadsworth Advanced Books and Software, 1985.

[Boyle, 1991] Boyle, J. Ventilatory control (Ventrol) simulation for education. *American Journal of Physiology*. **261**:25–29, 1991.

[Breese, 1990] Breese, J.S. and Horvitz, E.J. Ideal Reformulation of Belief Networks. *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI-90),* Cambridge, MA, pp. 64–72. Association for Uncertainty in Artificial Intelligence, 1990.

[Burton, 1984] Burton, G.G. and Hodgkin, J.E. (ed) *Respiratory Care, a Guide to Clinical Practice.* Philadelphia, PA: J.B. Lippincot, 1984.

[Castaing, 1989] Castaing, Y. and Manier, G. Hemodynamic disturbances and VA/Q matching in hypoxemic cirrhotic patients. *Chest*. **96**:1064–1069, 1989.

[Cellier, 1991] Cellier, F.E. *Continuous System Modeling.* New York, NY: Springer-Verlag, 1991.

[Coleman, 1983] Coleman, T.G. and Randall, J.E. HUMAN – A comprehensive physiological model. *The Physiologist*. **26**:15–21, 1983.

[Coleman, 1990] Coleman, W.P., Siegel, J.H., Giovannini, I., de Gaetano, A., Goodarzi, S., Tacchino, R.M. and Sganga, G. Probability and the patient state space. *International Journal of Clinical Monitoring and Computing*. **7**:201–215, 1990.

[Dagum, 1994] Dagum, P., Galper, A. and Seiver, A. Automating ARDS management: A dynamical systems approach. *Workshop notes of the AAAI Spring Symposium: AI in Medicine*, Stanford, CA, pp. 19–23, 1994.

[Dantzker, 1987] Dantzker, D.R. Ventilation-perfusion inequality in lung disease. *Chest*. **91**:749–54, 1987.

[Dickinson, 1977] Dickinson, C.J. *A computer model of human respiration.* Baltimore: University Park Press, 1977.

[Draper, 1993] Draper, D. Assessment and propagation of model uncertainty. *Fourth International Workshop on AI and Statistics*, Fort Lauderdale, FL, Society for AI and Statistics, 1993.

[East, 1992] East, T.D., Bohm, S.H., Wallace, C.J., Clemmer, T.P., Weaver, L.K., Orme, J., Jr. and Morris, A.H. A successful computerized protocol for clinical management of pressure control inverse ratio ventilation in ARDS patients. *Chest*. **101**:697-710, 1992.

[Fagan, 1980] Fagan, L.M. *VM: Representing Time-Dependent Relations in a Medical Setting*. Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, CA, 1980.

[Fagan, 1984] Fagan, L.M., Kunz, J.C., Feigenbaum, E.A. and Osborn, J.J. Extensions to the rule-based formalism for a monitoring task. In Buchanan, B. and Shortliffe, E.H. (ed), *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading: Addison-Wesley, pp. 397–423, 1984.

[Falkenhainer, 1991] Falkenhainer, B. and Forbus, K.D. Compositional modeling: finding the right model for the job. *Artificial Intelligence*. **51**:95–143, 1991.

[Farr, 1989] Farr, B. and Fagan, L.M. Decision-theoretic evaluation of therapy plans. *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-89)*, Washington, DC, pp. 188–192, Washington, DC: IEEE press, 1989.

[Farr, 1991] Farr, B.R. *Assessment of Preferences Through Simulated Decision Scenarios.* Ph.D. thesis, Section on Medical Informatics, Stanford University, Stanford, CA, 1991.

[Forbus, 1984] Forbus, K.D. Qualitative process theory. *Artificial Intelligence*. **24**:85–168, 1984.

[Franzone, 1982] Franzone, P.C., Paganuzzi, A. and Stefanelli, M. A mathematical model of iron metabolism. *Journal of Mathematical Biology*. **15**:173–201, 1982.

[Gaar, 1991] Gaar, K., Jr. Model to facilitate studying oxygen and carbon dioxide transport. *American Journal of Physiology*. **260**:6–9, 1991.

[Gaba, 1988] Gaba, D.M. and DeAnda, A. A comprehensive anesthesia simulation environment: re-creating the operating room for research and training. *Anesthesiology*. **69**:387–394, 1988.

[Gaines, 1979] Gaines, B. General systems research: Quo vadis. *General Systems: Yearbook of the Society for the Advancement of General Systems Theory*. **24**:1–9, 1979.

[Gardner, 1990] Gardner, R.M. and Shabot, M.M. Computerized ICU data management: Pitfalls and promises. *International Journal of Clinical Monitoring and Computing*. **7**:99–105, 1990.

[Gardner, 1991] Gardner, R.M., Hawley, W.L., East, T.D., Oniki, T.A. and Young, H.F. Real-time data acquisition: Experience with the Medical Information Bus (MIB). *Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-91)*, Washington, DC, pp. 813–817, New York, NY: McGraw Hill, 1991.

[Gillies, 1993] Gillies, B.S., Posner, K., Freund, P. and Cheney, F. Failure rate of pulse oximetry in the postanesthesia care unit. *Journal of Clinical Monitoring*. **9**:326–329, 1993.

[Gravenstein, 1983] Gravenstein, J.S. and Newbower, R.S. *An Integrated Approach to Monitoring.* Woburn, MA: Butterworth's, 1983.

[Gruber, 1993] Gruber, T. Model formulation as a problem solving task: Computer-assisted engineering modeling. *International Journal of Intelligent Systems*. **8**:105–127, 1993.

[Guyton, 1983] Guyton, A.C., Coleman, T.G. and Granger, H.J. Circulation: Overall regulation. *Annual Reviews of Physiology*. **34**:13–46, 1983.

[Guyton, 1984] Guyton, A.C., Coleman, T.G., Jr., R.D.M. and Hall, J.E. Some problems and solutions for modeling overall cardiovascular regulation. *Mathematical Biosciences*. **72**:141–155, 1984.

[Guyton, 1991] Guyton, A.C. Blood pressure control – special role of the kidneys and body fluids. *Science*. **252**:1813–1816, 1991.

[Hannan, 1981] Hannan, E.J. Estimating the dimension of a model. *Journal of Multivariate Analysis*. **11**:459–473, 1981.

[Hayes-Roth, 1992] Hayes-Roth, B., Washington, R., Ash, D., Hewett, R., Collinot, A., Vina, A. and Seiver, A. Guardian: a prototype intelligent agent for intensive care-monitoring. *Artificial Intelligence in Medicine*. **4**:165–85, 1992.

[Hayes-Roth, 1994] Hayes-Roth, B. and Collinot, A. A satisficing cycle for real-time reasoning in intelligent agents. *Expert Systems with Applications*. **7**:31–42, 1994.

[Heckerman, 1989] Heckerman, D.E., Horvitz, E.J. and Nathwani, B.N. Update on the Pathfinder project. *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-89)*, Washington, DC, pp. 203–207, Washington, DC: IEEE press, 1989.

[Heffels, 1990] Heffels, J.J.M. Patient simulator for anesthesia training: A mechanical lung model and a physiological software model. Technical report EUT 90-E-235, Eindhoven University, Jan., 1990.

[Helal, 1994] Helal, M.A., Watts, K.C. and Marble, A.E. Hydrodynamic simulation of arterial networks which include compliant and rigid bypass grafts. *Journal of Biomechanics*. **27**:277–287, 1994.

[Henderson, 1991] Henderson, S., Crapo, R.O., Wallace, C.J., East, T.D., Morris, A.H. and Gardner, R.M. Performance of computerized protocols for the management of arterial oxygenation in an intensive care unit. *Int J Clin Monit Comput*. **8**:271-80, 1991.

[Hill, 1973] Hill, E.P., Power, G.G. and Longo, L.D. Mathematical simulation of pulmonary O2 and CO2 exchange. *American Journal of Physiology*. **224**:904–917, 1973.

[Hollan, 1984] Hollan, J.D., Hutchins, E.L. and Weitzman, L. STEAMER: An interactive inspectable simulation-based training system. *AI Magazine*. 15–27, 1984.

[Horvitz, 1989] Horvitz, E.J., Heckerman, D.E., and Cooper, D.F. Reflection and action under scarce resources: Theoretical principles and empirical study. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, Detroit, MI, pp. 1121–1127, Palo Alto, CA: Morgan Kaufman, 1989.

[Horvitz, 1990] Horvitz, E.J. Computation and Action Under Bounded Resources. Ph.D. thesis, Section on Medical Informatics, Stanford University, 1990.

[Kapitan, 1987] Kapitan, K.S. and Wagner, P.D. Linear programming analysis of VA/Q distributions: average distribution. *Journal of Applied Physiology*. **62**:1356–1362, 1987.

[Kashyap, 1982] Kashyap, R.L. Optimal choice of AR and MA parts in autoregressive moving average models. *IEEE Trans PAMI*. **4**:99–104, 1982.

[Kass, 1993] Kass, R.E. and Raftery, A.E. Bayes factors and model uncertainty. Technical Report 571, Department of Statistics, Carnegie Mellon University, March, 1993.

[Kaufman, 1987] Kaufman, R.D., Patterson, R.W. and Lee, A.S. Derivation of VA/Q distribution from blood-gas tensions. *British Journal of Anaesthesia*. **59**:1599–1609, 1987.

[Keeney, 1976] Keeney, R.L. and Raiffa, H. *Decisions with multiple objectives: Preferences and value tradeoffs.* New York, NY: John Wiley and Sons, 1976.

[Kirby, 1984] Kirby, R.A., Smith, R.A., and Desautels, D.A. Mechanical ventilation. In Burton, G.G. and Hodgkin, J.E. (ed), *Respiratory Care, a Guide to Clinical Practice*. Philadelphia: J B Lippincot Co., pp. 556–647, 1984.

[Kowalik, 1988] Kowalik (ed) *Coupling Symbolic and Numerical Computing in Expert Systems.* The Netherlands: Elsevier Science Publishers, B.V., 1988.

[Kuipers, 1984] Kuipers, B. and Kassirer, J.P. Causal reasoning in medicine: Analysis of a protocol. *Cognitive Science*. **8**:363–385, 1984.

[Kuipers, 1985] Kuipers, B.J. The limits of qualitative simulation. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pp. 128–136, Palo Alto, CA: Morgan Kaufman, 1985.

[Kuipers, 1986] Kuipers, B. Qualitative simulation. *Artificial Intelligence*. **29**:289–338, 1986.

[Langlotz, 1985] Langlotz, C., Fagan, L., Tu, S., Sikic, B. and Shortliffe, E. A therapy planning architecture that combines decision theory and artificial intelligence techniques. *Computers and Biomedical Research*. **20**:279–303, 1985.

[Lauritzen, 1988] Lauritzen, S.L. and Spiegelhalter, D.J. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*. **50**:157–224, 1988.

[Lee, 1987] Lee, A.S., Patterson, R.W. and Kaufman, R.D. Relationships among ventilation-perfusion distribution, multiple inert gas methodology and metabolic blood-gas tensions. *British Journal of Anaesthesia*. **59**:1579–1598, 1987.

[Lefevre, 1988] Lefevre, J., Roucou, D., Lambert, K. and Maes, B. Simulation and computer-aided instructions in cardiovascular physiology. *Conference on Computers in Cardiology*, Leuven, Belgium, pp. 597, Washington, DC: IEEE Press, 1988.

[Li, 1993] Li, C.W. and Cheng, H.D. A nonlinear fluid model for pulmonary blood circulation. *Journal of Biomechanics*. **26**:653–664, 1993.

[Lim, 1990] Lim, L.L. A statistical model of the VA/Q distribution. *Journal of Applied Physiology*. **69**:281–292, 1990.

[Meyer, 1994] Meyer, K.W., Buglia, J.J. and Desai, P.N. Lifetimes of lunar satellite orbits. Technical Report NASA TP-3394, NASA Langley Research Center, March, 1994.

[Miller, 1985] Miller, P.L. Goal-directed critiquing by computer: Ventilator management. *Computers and Biomedical Research*. **18**:422–438, 1985.

[Mish, 1988] Mish, F.C. (ed) Webster's Ninth New Collegiate Dictionary, NeXT Digital Edition. Springfield, MA: Merriam–Webster, 1988.

[Moore, 1992] Moore, J., Tabarrok, B. and Haddow, J.B. Mathematical modelling of +Gz acceleration-induced stresses in the human ventricular myocardium. *Physiologist*. **35**:S169–172, 1992.

[Morris, 1994] Morris, A.H., Wallace, C.J., Menlove, R.L., Clemmer, T.P., Orme, J., Jr. Randomized clinical trial of pressure-controlled inverse ratio ventilation and extracorporeal CO2 removal for adult respiratory distress syndrome. *American Journal of Respiratory and Critical Care Medicine*. **149**:295–305, 1994.

[Murthy, 1987] Murthy, S.S. and Addanki, S. PROMPT: An innovative design tool. *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pp. 637–642, Menlo Park, CA: AAAI press, 1987.

[Nayak, 1992] Nayak, P.P., Joskowicz, L. and Addanki, S. Automated model selection using context-dependent behaviors. *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Jose, CA, pp. 710–716, Menlo Park, CA: AAAI press, 1992.

[Neelankavil, 1987] Neelankavil, F. *Computer simulation and modeling*. Chichester: John Wiley and sons, 1987.

[NIH, 1989] NIH Office of Medical Applications of Research. Modeling in biomedical research: An assessment of current and potential approaches: Applications to studies in cardiovascular/pulmonary function and diabetes. NIH Technology Assessment Statement Online (4):19, 1989.

[Oniki, 1993] Oniki, T.A. and Gardner, R.M. Computerized detection of arterial oxygen desaturations in an intensive care unit. *Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC-93)*, Washington, DC, pp. 356–360, New York, NY: McGraw Hill, 1993.

[Ott, 1988] Ott, L. *An Introduction to Statistical Methods.* Boston, MA: PWS-Kent Publishing Co., 1988.

[Patil, 1981] Patil, R. Causal representation of patient illness for electrolyte and acid-base diagnosis. Technical Report MIT/LCS/TR-267, Laboratory for Computer Science, Massachusetts Institute of Technology, October, 1981.

[Penberthy, 1987] Penberthy, S. *Incremental Analysis in the Graph of Models.* S.M. thesis, Department of Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1987.

[Petrini, 1986] Petrini, M.F. Distribution of ventilation and perfusion: A teaching model. *Computers in Biology and Medicine*. **16**:431–444, 1986.

[Press, 1988] Press, W.H., Flannery, B.P., Teulokolsky, S.A. and Vetterling, W.T. *Numerical Recipes in C: The Art of Scientific Programming.* Cambridge, MA: Cambridge University Press, 1988.

[Purcell, 1963] Purcell, E.M. *Electricity and Magnetism.* New York, NY: McGraw Hill, 1963.

[Ramamurthi, 1993] Ramamurthi, K. and Agogino, A.M. Real-time expert system for fault-tolerant supervisory control. *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control*. **115**:219–227, 1993.

[Riley, 1949] Riley, R.L. and Cournand, A. "Ideal" alveolar air and the analysis of ventilation-perfusion relationships in the lungs. *Journal of Applied Physiology*. **1**:825–847, 1949.

[Ringsted, 1989] Ringsted, C.V., Eliasen, K., Andersen, J.B., Heslet, L. and Qvist, J. Ventilation-perfusion distributions and central hemodynamics in chronic obstructive pulmonary disease. Effects of terbutaline administration. *Chest*. **96**:976–983, 1989.

[Rissanen, 1978] Rissanen, J. Modeling by shortest data description. *Automatica*. **14**:465–471, 1978.

[Roca, 1988] Roca, J., Ramis, L., Rodriguez, R.R., Ballester, E., Montserrat, J.M. and Wagner, P.D. Serial relationships between ventilation-perfusion inequality and spirometry in acute severe asthma requiring hospitalization. *American Review of Respiratory Disease*. **137**:1055–1061, 1988.

[Roisin, 1990] Roisin, R.R. Gas exchange abnormalities in asthma. *Lung*. **Supplement**:599–605, 1990.

[Ross, 1984] Ross, S.M. *A First Course in Probability*. New York, NY: Macmillan, 1984.

[Rudowski, 1989] Rudowski, R., Frostell, C. and Gill, H. A knowledge-based support system for mechanical ventilation of the lungs. The KUSIVAR concept and prototype. *Computer Methods and Programs in Biomedicine*. **30**:59–70, 1989.

[Rudowski, 1990] Rudowski, R., Skreta, L., Baehrendtz, S., Bokliden, A. and Matell, G. Computer simulation of a patient end tidal CO2 controller system. *Computer Methods and Programs in Biomedicine*. **31**:33–42, 1990.

[Rutledge, 1989] Rutledge, G.W., Thomsen, G., Beinlich, I., Farr, B., Sheiner, L. and Fagan, L. Combining qualitative and quantitative computation in a ventilator therapy planner. *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-89)*, Washington, DC, pp. 315–319, Washington, DC: IEEE press, 1989.

[Rutledge, 1990] Rutledge, G.W., Andersen, S.K., Polaschek, J.X. and Fagan, L.M. A belief-network model for interpretation of ICU data. *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-90)*, Washington, DC, pp. 785–789, Los Alamitos, CA: IEEE press, 1990.

[Rutledge, 1991] Rutledge, G.W., Thomsen, G., Farr, B., Tovar, M., Sheiner, L. and Fagan, L. VentPlan: An intelligent ventilator management advisor. *Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-91)*, Washington, DC, pp. 869–871, New York, NY: McGraw-Hill, 1991.

[Rutledge, 1993a] Rutledge, G.W., Thomsen, G., Farr, B., Beinlich, I., Tovar, M.A., Sheiner, L. and Fagan, L.M. The design and implementation of a ventilator-management advisor. *Artificial Intelligence in Medicine*. **5**:67–82, 1993.

[Rutledge, 1993b] Rutledge, G.W. Dynamic selection of models for a ventilator-management advisor. *Proceedings of the Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC-93)*, Washington, DC, pp. 344–350, 1993.

[Rutledge, 1994a] Rutledge, G.W. and Shachter, R.D. A method for the dynamic selection of models under time constraints. In Cheeseman, P. and Oldford, R.W. (ed), *Selecting Models from Data: Artificial Intelligence and Statistics IV.* New York, NY: Springer-Verlag, pp. 79–88, 1994.

[Rutledge, 1994b] Rutledge, G.W. VentSim: A simulation model of cardiopulmonary physiology. *Journal of the American Medical Informatics Association*. **Symposium supplement**:878–883, 1994.

[Sacks, 1988] Sacks, E.P. *Automatic Qualitative Analysis of Ordinary Differential Equations Using Piecewise Linear Approximations.* Ph.D. thesis, Massachusetts Institute of Technology, 1988.

[Sacks, 1991] Sacks, E.P. Automatic analysis of one-parameter planar ordinary differential equations by intelligent numerical simulation. *Artificial Intelligence*. **48**:27–56, 1991.

[Schreiner, 1992] Schreiner, W. Computer simulation of the coronary circulation. *Simulation*. **59**:15–23, 1992.

[Schwarz, 1978] Schwarz, G. Estimating the dimension of a model. *Annals of Statistics*. **6**:461–464, 1978.

[Schwid, 1987] Schwid, H.A. A flight simulator for general anesthesia training. *Computers and Biomedical Research*. **20**:64–75, 1987.

[Sclove, 1994] Sclove, S.L. Small sample and large sample statistical model selection criteria. In Cheeseman, P. and Oldford, R.W. (ed), *Selecting Models from Data: AI and Statistics IV.* New York, NY: Springer-Verlag, 1994.

[Sears, 1964] Sears, F.W. and Zemansky, M.W. *University Physics.* Reading, MA: Addison-Wesley, 1964.

[Shabot, 1989] Shabot, M.M., LoBue, M., Leyerle, B.J. and Dubin, S.B. Inferencing strategies for automated ALERTS on critically abnormal laboratory and blood gas data. *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care* (SCAMC-89), Washington, DC, pp. 54-57, Washington, DC: IEEE press, 1989.

[Shabot, 1990] Shabot, M.M., LoBue, M., Leyerle, B.J. and Dubin, S.B. Decision support alerts for clinical laboratory and blood gas data. *International Journal of Clinical Monitoring and Computing*. **7**:27-31, 1990.

[Sheiner, 1982] Sheiner, L.B. and Beal, S.L. Bayesian individualization of pharmacokinetics: Simple implementation and comparison with non-Bayesian methods. *Journal of Pharmaceutical Sciences*. **71**:1344–1348, 1982.

[Shibata, 1976] Shibata, R. Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika*. **63**:117–126, 1976.

[Shortliffe, 1979] Shortliffe, E.H., Buchanan, B.G. and Feigenbaum, E.A. Knowledge engineering for medical decision making: A review of computer-based clinical decision aids. *Proceedings of the IEEE*. **67**:1207–1224, 1979.

[Simon, 1990] Simon, H.A. Prediction and prescription in systems modeling. *Operations Research*. **38**:7–14, 1990.

[Sittig, 1988] Sittig, D.F. *COMPAS: A Computerized Patient Advice System to Direct Ventilatory Care.* Ph.D. thesis, Dept. Medical Informatics, University of Utah, 1988.

[Sittig, 1989] Sittig, D.F., Pace, N.L., Gardner, R.M., Beck, E. and Morris, A.H. Implementation of a computerized patient advice system using the HELP clinical information system. *Computers and Biomedical Research*. **22**:474–487, 1989.

[Snyder, 1969] Snyder, M.F. and Rideout, V.C. Computer simulation studies of the venous circulation. *IEEE Transactions on Biomedical Engineering*. **16**:325–334, 1969.

[Stewart, 1988] Stewart, D.J., Barnett, C.C., Nelson, T.A. and Amlaner, C.J. A simulated white-tail deer population. *1988 Summer Computer Simulation Conference*, Seattle, WA, pp. 460–466, SCS: San Diego, CA, 1988.

[Summers, 1992] Summers, R.L. and Montani, J.P. Computer model of cardiac diastolic dynamics. *Proceedings of Computers in Cardiology Conference*, Durham, NC, pp. 583–585, Los Alamitos, CA: IEEE press, 1992.

[Sun, 1992] Sun, Y. and Chiaramida, S. Simulation of hemodynamics and regulatory mechanisms in the cardiovascular system based on a nonlinear and time-varying model. *Simulation*. **59**:28–36, 1992.

[Thomsen, 1989] Thomsen, G. and Sheiner, L. SIMV: An application of mathematical modeling in ventilator management. *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-89)*, Washington, DC, pp. 320–324, Washington, DC: IEEE press, 1989.

[Tomlinson, 1993] Tomlinson, S.P., Lo, J. and Tilley, D.G. Time transient gas exchange in the respiratory system. *IEEE Engineering in Medicine and Biology Magazine*. **12**:64–70, 1993.

[Torres, 1989] Torres, A., Reyes, A., Roca, J., Wagner, P.D. and Rodriguez, R.R. Ventilation-perfusion mismatching in chronic obstructive pulmonary disease during ventilator weaning. *American Review of Respiratory Disease*. **140**:1246–1250, 1989.

[Tovar, 1991] Tovar, M.A., Rutledge, G.W., Lenert, L.A. and Fagan, L.M. The design of a user-interface for a ventilator-management advisor. *Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care (SCAMC-91)*, Washington, DC, pp. 828–832, New York, NY: McGraw-Hill, 1991.

[Uckun, 1993] Uckun, S., Dawant, B.M. and Lindstrom, D.P. Model-based diagnosis in intensive care monitoring: the YAQ approach. *Artificial Intelligence in Medicine*. **5**:31–48, 1993.

[Wagner, 1987] Wagner, P.D., Hedenstierna, G. and Bylin, G. Ventilation-perfusion inequality in chronic asthma. *American Review of Respiratory Disease*. **136**:605–612, 1987.

[Wagner, 1974] Wagner, P.D., Saltzman, H.A. and West, J.B. Measurement of continuous distributions of ventilation-perfusion ratios: Theory. *Journal of Applied Physiology*. **36**:588–599, 1974.

[Weibel, 1989] Weibel, E.W. Lung morphometry and models in respiratory physiology. In Chiang, H.K. and Paiva, M. (ed), *Respiratory Physiology: An Analytical Approach.* New York, NY: Marcel Dekker, pp. 1–56, 1989.

[Weld, 1988] Weld, D.S. Comparative analysis. *Artificial Intelligence*. **36**:333–373, 1988.

[Weld, 1992] Weld, D.S. Reasoning about model accuracy. *Artificial Intelligence*. **56**:255–300, 1992.

[West, 1980] West, J.B. and Wagner, P.D. Predicted gas exchange on the summit of Mt. Everest. *Respiration Physiology*. **42**:1–16, 1980.

[West, 1989] West, J.B. *Respiratory Physiology*. Baltimore, MD: Williams and Wilkins, 1979.

[Widman, 1989] Widman, L.E. Expert system reasoning about dynamic systems by semi-quantitative simulation. *Computer Methods and Programs in Biomedicine*. **29**:95–113, 1989.

[Wiggs, 1990] Wiggs, B.R., Moreno, R., Hogg, J.C., Hilliam, C. and Pare, P.D. A model of the mechanics of airway narrowing. *Journal of Applied Physiology*. **69**:849–860, 1990.

[Young, 1995] Young, W.H., Gardner, R.M., East, T.D., Elliott, C.G. and Turner, K. Computerized ventilator charting: Artifact rejection and data reduction. *in press*. 1995.

[Yu, 1979] Yu, V.L., Buchanan, B.G., Shortliffe, E.H., Wraith, S.M., Davis, R., Scott, A.C. and Cohen, S.N. Evaluating the performance of a computer-based consultant. *Computer Programs in Biomedicine*. **9**:95–102, 1979.

[Zeigler, 1976] Zeigler, B.P. *Theory of Modelling and Simulation.* New York, NY: John Wiley and Sons, 1976.

[Zeigler, 1984] Zeigler, B.P. *Multifaceted Modeling and Discrete Event Simulation.* London: Academic Press, 1984.