TECHNISCHE UNIVERSITÄT WIEN

DISSERTATION

Diagnostic Monitoring of Clinical Time Series

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Technisch Naturwissenschaftliche Fakultät

von

Friedrich Steimann, Friedrichstraße 2/14, A-1010 Wien
Matrikelnummer 9226982
geboren am 5.März.1965 in Heidelberg

Wien, im

Kurzfassung

Mit der Verfügbarkeit einer ständig zunehmenden Anzahl technischer Geräte und Verfahren zur Patientenüberwachung stellt sich unweigerlich das Problem der Interpretation der gemessenen Werte. Klinische Zeitreihen, wie sie auf Intensivstationen und im Operationssaal, aber auch bei der Beobachtung ambulanter Patienten anfallen, bedürfen einer anspruchsvolleren Auswertung als die Momentaufnahme im Rahmen einer einmaligen Konsultation.

In der vorliegenden Arbeit wird das allgemeine Problem der diagnostischen Interpretation klinischer Zeitreihen in einen theoretischen Rahmen eingebettet, der die Synthese problemspezifischer, die Befunde selbst interpretierender Monitore erlaubt. Mit der praktischen Umsetzung dieses Rahmens können Systeme und Methoden unterschiedlichsten Ursprungs, darunter aus der künstlichen Intelligenz, der Signalverarbeitung und der Zeitreihenanalyse, zu einem heterogenen Monitor verbunden werden, der die Transformation automatisch und manuell erhobener Befunde in zeitlich stabile Abstraktionen oder Diagnosen durchführt.

Als Neuerungen werden Methoden zur Trenderkennung auf der Basis unscharfer Parameterverläufe und zur Verfolgung typischer Krankheitsverläufe mittels unscharfer Automaten vorgestellt. Beide Ansätze lösen jeweils ein zentrales Problem der klinischen Patientenüberwachung: Die Trenderkennung extrahiert solche Symptome, die allein in der zeitlichen Entwicklung bestimmter Parameter erkennbar werden, während die unscharfen Automaten die Beobachtungen im Kontext des bereits geschehenen, also des Krankheitsverlaufs interpretieren. Beide Ansätze bedienen sich der Theorie der unscharfen Mengen von Lotfi A. Zadeh und gestatten damit eine stufenlose Differenzierung zwischen nicht vorhandenen und vorhandenen Symptomen sowie die Darstellung des fließenden Übergangs von einem Krankheitsstadium zum nächsten.

Um die Praktikabilität des gewählten Ansatzes unter Beweis zu stellen, werden verschiedene Experimente durchgeführt. Aus dem Anwendungsgebiet der Intensivmedizin wird eine Anzahl kleinerer Monitore vorgestellt, die sich die Interpretation in kurzen Abständen automatisch erhobener Daten zur Aufgabe machen. Die Daten stammen von einer Patientin mit dem Atemnotsyndrom des Erwachsenen (ARDS) und von einer Reihe von Patienten, die von der künstlichen Beatmung entwöhnt werden sollen. Besondere Betonung liegt auf der Verwendung verschiedener Varianten der Trenderkennung und der Modellierung etwaiger Komplikationen durch endliche Automaten.

Die Experimente zeigen, daß mit Hilfe der vorgeschlagenen Monitore eine Reduktion großer Informationsmengen auf ein überschaubares Maß möglich ist. Durch die Berücksichtigung medizinischer Unschärfen zeichnen sich zudem sich anbahnende Veränderungen im Zustand des Patienten frühzeitig ab, so daß ein rechtzeitiges Eingreifen ermöglicht wird.

Ein anderes Beispiel für die Notwendigkeit der diagnostischen Interpretation klinischer Zeitreihen ist die Überwachung ambulanter Patienten. Die Methode zur Trenderkennung wird daher zusätzlich bei der Reihenuntersuchung Schwangerer auf postkonzeptionelle Erstinfektion mit dem Parasiten *Toxoplasma gondii* erprobt. Dieses Einsatzgebiet unterscheidet sich vom intensivmedizinischen in erster Linie durch die mangelnde Verfügbarkeit dichter, das heißt in ausreichend kurzen Abständen gemessener Daten. Mit der fehlenden Information wächst die Unsicherheit in der Diagnose, die nun nicht nur die allgemeine individuelle Variabilität serologischer Verläufe, sondern auch die Unbestimmtheit eines konkret vorliegenden Verlaufs in Betracht ziehen muß.

Experimente, die mit den Untersuchungsergebnissen von 1000 Schwangeren durchgeführt worden sind, zeigen, daß die auf die automatische Trenderkennung gestützte Diagnose eine hohe Übereinstimmung (98,2%) mit der des Labors hat, wenn alle Daten zur Verfügung stehen.

Eine Diskussion des gewählten Ansatzes im Vergleich mit anderen Arbeiten sowie eine Bewertung der Kosten und Nutzen der Verwendung unscharfer Mengen schließen die Arbeit ab.

Disclaimer

Throughout this work, all anonymous persons are referred to using pronouns in the male gender. I generally do so to maintain readability, not to discriminate against females. It is understood that all positions could equally be filled by women.

Acknowledgements

I wish to thank my dissertation advisor, Professor Klaus-Peter Adlassnig, for his support and letting me do it my way. Dr. Michael Hayde and Professor Arnold Pollak from the University Children's Hospital in Vienna have greatly supported my work in the field of toxoplasmosis; thank you, too.

Special thanks go to Serdar Uckun, for pointing me in the right direction and giving me the opportunity to speak at the 1994 Stanford Spring Symposium of Artificial Intelligence in Medicine.

Table of Contents

Chapter 1

Introduction

In recent years, much effort has been spent on attempts to make clinical instrumentation more intelligent. Nevertheless, the capabilities of current bedside monitors are still limited to the recording and display of directly observed—or, by some fairly simple calculation, derived—numerical data. Alarms triggered on threshold crossings seem to be the only data abstraction feature added to date, but, considering the numerous reports on their high false alarm rates and accordingly low utilization, their behaviour can hardly be called intelligent.

Generally, monitoring is not tied to the bedside: outpatients suffering from chronic disease may also require regular surveillance, posing basically the same problems. Repeated consultation of one-shot diagnostic systems analysing only instantaneous findings, is not sensitive to the disease history and thus cannot be deemed sufficiently intelligent either.

What then would be expected from an intelligent monitor? Ideally, an intelligent monitor would track and analyse the patient's disease history, provide therapy advice and critique, or, even more ambitiously, implement closed-loop therapy control. However, therapy support, no matter at which level, invariably requires correct interpretation of findings as a prior step. Because this interpretation is in essence what is called diagnosis, the diagnostic monitoring of clinical time series must be viewed as a fundamental contribution towards more machine intelligence in medical care.

1.1 Problems of Diagnostic Monitoring

The often cited information overload in intensive care units or operation rooms is of a quality quite different from that in many technical environments: it is not the sheer number of monitored parameters that strains the clinical staff, but the often subtle containment of signs in the development of a single or some combination of variables. Diagnostic monitoring therefore requires a different quality from computers: not the fast and reliable "brute force" processing of

mass data, but the competent and sensitive tracing of significant developments in relatively few parameters.

The problem of diagnostic monitoring has been approached from many sides: time series analysis, signal processing, and artificial intelligence, to name a few, are disciplines that have dedicated much of their work to the solution of this and related problems.

However, any practical attempt to tackle the problem invariably reveals its interdisciplinary nature: methods of artificial intelligence are just as indispensable as techniques from signal processing and time series analysis. Whereas each discipline has contributed its share, as noted in [Dawant 94], not much work has been spent on their integration. Because no single method alone will solve the problem, the incompatibility, both conceptional and practical, of partial solutions from different disciplines is a persistent obstacle for the creation of integrated solutions as well as for comparative studies.

The following presents a collection of fundamental problems any successful diagnostic monitoring system will have to take into account.

1.1.1 The Crucial Role of Time

The nouns “development”, “sequence” and “time series” used above in connection with findings all have one thing in common: they suggest the significance of time. Time is inherent in the very notion of monitoring. Without regard to it, a monitoring system misses out one (if not the most) important aspect of clinical information.

While this fact, although very obvious, has long been ignored by work in the field of deductive (medical) systems [McDermott 82], other disciplines such as signal processing have made it their very basis right from the beginning. Despite a recent tendency to adopt time-related issues in all branches concerned with clinical monitoring, there is still not one, agreed-upon way of doing it. Rather, representation and processing of temporal information depends on the ontological constraints of the respective solution.

Adequate treatment of time imposes several nontrivial problems: trend detection and gradual uncertainty arising from sparse sampling, to mention but two, are time-related issues that are ignored by all atemporal diagnostic systems.

1.1.2 The Necessity of Trend Detection

It is generally agreed that trends play an outstanding role in clinical decision making [Blom 85]. Whereas an instantaneous observation requires consideration of idiosynchrasy and transience, a trend is usually of unconditional diagnostic value. In particular, unlike an instantaneous finding, a trend often can

- distinguish natural fluctuations from symptomatic developments,
- indicate forthcoming perturbations,
- differentiate competing diagnoses, and
- indicate the patient’s response to a therapy.

Trends can be observed on various levels of abstraction. At one end, a trend manifests itself in a characteristic change in one or a number of directly observed variables. At the other, the tendency of the patient to proceed in one of a number of possible disease histories may also be referred to as a trend. As will be seen, respective trend detection methods differ significantly with the level on which they are employed.

1.1.3 The Problem of Uncertainty

> *"Medical data are disturbingly "soft;" they are defined, collected, and interpreted with a degree of variability and inaccuracy which falls far short of the standards which engineers expect from most data."* [Komaroff 79]

In diagnostic monitoring, uncertainty arises in different ways: imprecision as to the uncertainty associated with measurement, vagueness as to the uncertainty inherent in the meaning of semantic concepts such as normality and abnormality and the reasoning based thereon, indeterminacy as to the uncertainty in the occurrence of an event, and ignorance as to the uncertainty caused by missing values.

The types or kinds of uncertainty are not to be confused with the formal models available to deal with them: probability theory for example can equally be used to model imprecision in measurement (normally distributed noise), vagueness in relations (a consequence is probable, yet not certain) and indeterminacy of a future event (chance).

Adequate treatment of the full scale of uncertainty in diagnostic monitoring poses various problems. In the case of finite problem spaces, uncertainty may be covered by treating all theoretically possible cases separately, for example by maintaining a number of competing diagnostic hypotheses. For the more general, infinite problem space, however, uncertainty can only be dealt with by employing models that allow the encompassing of possibly infinite sets of possibilities with a finite number of expressions, as do probability and possibility theory.

In either case, taking uncertainty into account invariably increases the complexity of the monitoring problem, particularly if different kinds of uncertainty are treated as orthogonal (i.e. mutually independent) properties of information. Further processing of uncertain information encoded in the form of alternative hypotheses usually causes a combinatorial explosion, while processing of possibility or probability distributions frequently leads to a progressive loss in certainty, reflected in a widening and flattening of the distributions.

1.2 Fuzzy Set Theory

Since its advent in 1965, fuzzy set theory and its offspring (fuzzy logic, fuzzy control, etc.) have been the subject of numerous text books and filled thousands of pages in periodicals. The utilization of fuzziness in this work, however, is very basic in that it relies solely on the fundamentals of fuzzy set theory as described in the following.

1.2.1 Fundamentals

Fuzzy set theory comes with a basic notion of fuzzy sets and fuzzy set operations. The definitions naturally extend to fuzzy relations, and with the extension principle a scheme allowing the fuzzification of ordinary functions and operations is provided.

Fuzzy Sets

A *fuzzy subset* of a set X is a set of pairs

$$\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) | x \in X\},$$

where

$$\mu_{\widetilde{A}} : X \rightarrow [0, 1]$$

is the *membership function* of $\widetilde{A}$ that assigns to each $x \in X$ the *degree* or *grade* to which x belongs to $\widetilde{A}$. $[0, 1]$ is called the *valuation set* of $\mu_{\widetilde{A}}$. Other definitions of fuzzy sets are also possible; the one used here is the one introduced by Zadeh [Zadeh 65] and is still the most common. Note that I prefer not to use Zadeh's notation (employing sum and integral signs),

because it is likely to be confused with the analytical notation required in the context of trend detection later in this work.

Clearly, the membership function is a generalization of the characteristic function of ordinary sets, which has a binary valuation set $\{0, 1\}$. Fuzzy sets are therefore regarded as generalizations of ordinary sets. To distinguish fuzzy sets from ordinary ones, I add a tilde above their labels.

Fuzzy sets are related to ordinary sets via so-called α-*cuts*. In short, the α-cut of a fuzzy set $\widetilde{A}$ for a given $\alpha \in \,]0, 1]$ is defined as

$$\widetilde{A}_\alpha = \{x \in X | \mu_{\widetilde{A}}(x) \geq \alpha\},$$

which is equivalent to defining the characteristic function

$$\mu_{\widetilde{A}_\alpha}(x) = \begin{cases} 1 & \text{if } \mu_{\widetilde{A}}(x) \geq \alpha \\ 0 & \text{else} \end{cases} .$$

Conversely, the membership function of $\widetilde{A}$ can be expressed in terms of the characteristic function of its α-cuts as in

$$\mu_{\widetilde{A}}(x) = \sup_{\alpha \in \,]0,1]} \min(\alpha, \mu_{\widetilde{A}_\alpha}(x)) ,$$

i.e., as the least upper bound α so that $\widetilde{A}_\alpha$ includes x [Dubois 80].

The *support* of a fuzzy set $\widetilde{A}$ is the ordinary set

$$\{x | \mu_{\widetilde{A}}(x) > 0\}|.$$

Note that

$$\lim_{\alpha \to 0} \widetilde{A}_\alpha = \{x | \mu_{\widetilde{A}}(x) > 0\}|.$$

A fuzzy set is *convex* if and only if its support and all of its α-cuts are convex. The *height* of a fuzzy set is its membership function's least upper bound

$$\text{hgt}(\widetilde{A}) = \sup_{x \in X} \mu_{\widetilde{A}}(x).$$

A fuzzy set is normalized if its height equals 1. A *fuzzy number* is a convex, normalized fuzzy subset of $\Re$, the real line. (The condition that there may be only one $x \in \Re$ for which the degree of membership is 1 is not required in this definition.) Note that graphical depictions of fuzzy numbers are usually hump-shaped, i.e., they have no local minima other than those at their borders.

$\widetilde{\wp}(X)$ denotes the set of fuzzy subsets (the fuzzy power set) of X. I denote the set of fuzzy numbers, i.e., the set of convex, normalized fuzzy subsets of $\Re$, with $\widetilde{\Re}$.

Fuzzy Set-Theoretic Operations

Analogous to ordinary set theory, fuzzy set theory provides set-theoretic operations such as intersection, union, and complement.

The complement of a fuzzy set $\widetilde{A}$ is defined as

$$\neg\widetilde{A} = \{(x, \mu) | \mu = 1 - \mu_{\widetilde{A}}(x)\} .$$

The union and intersection of two fuzzy set $\widetilde{A}$ and $\widetilde{B}$ are defined by

$$\widetilde{A} \cup \widetilde{B} = \{(x, \mu) | \mu = \max(\mu_{\widetilde{A}}(x), \mu_{\widetilde{B}}(x))\}$$

and

$$\widetilde{A} \cap \widetilde{B} = \{(x, \mu) | \mu = \min(\mu_{\widetilde{A}}(x), \mu_{\widetilde{B}}(x))\} ,$$

respectively. Other definitions of these and further set operations have also been proposed, but the ones described here are required in the proofs of Section 3.7.

Fuzzy Relations

An *n*-ary *fuzzy relation* $\widetilde{R}$ is a fuzzy subset of the Cartesian product $X_1 \times \ldots \times X_n$ of *n* ordinary sets $X_1, \ldots, X_n$. It is characterized by a membership function $\mu_{\widetilde{R}} : X_1 \times \ldots \times X_n \to [0,1]$ such that

$$\widetilde{R} = \{((x_1, \ldots, x_n), \mu_{\widetilde{R}}(x_1, \ldots, x_n)) | x_1 \in X_1, \ldots, x_n \in X_n\} .$$

Again, I refrain from using Zadeh's notation because it is misleading in the context of this work.

The Extension Principle

The extension principle allows the extension of functions with ordinary set domain and codomain to fuzzy set domain and codomain. Given an *n*-ary function

$$f : X_1 \times \ldots \times X_n \to Y,$$

it is extended to its fuzzified version

$$\widetilde{f} : \widetilde{\wp}(X_1) \times \ldots \times \widetilde{\wp}(X_n) \to \widetilde{\wp}(Y)$$

by

$$\mu_{\widetilde{f}(\widetilde{x}_1,\ldots,\widetilde{x}_n)}(y) = \begin{cases} 0 & \text{if } \neg\exists x_1, \ldots, x_n : y = f(x_1, \ldots, x_n) \\ \sup_{x_1,\ldots,x_n : y=f(x_1,\ldots,x_n)} \min(\mu_{\widetilde{x}_1}(x_1), \ldots, \mu_{\widetilde{x}_n}(x_n)) & \text{else} \end{cases} \tag{1.1}$$

If the extension principle is applied to real algebra, it yields a fuzzy algebra operating on fuzzy numbers [Dubois 80]. Other examples for the application of the extension principle will be found in Sections 3.8 and 3.9.

1.2.2 Interpretation of Fuzzy Sets

The adequate interpretation of fuzzy sets has caused much debate. Whenever in doubt as to what a fuzzy set means, I find it useful to recall that fuzzy sets are generalizations of ordinary sets; so whatever one assumes a fuzzy set to mean, it should be the same as what an ordinary set stands for, only fuzzier.

Sets are often employed as abstractions of collections of objects with identical properties. For example, the set of real numbers $\Re$ and the set of patients *P* represent collections of objects that are similar in that they can be calculated with or be the subject of monitoring. Now if a set abstracts from objects, then so does a fuzzy set, the only difference being that objects with gradual membership are only partially compatible (or incompatible) with the abstraction.

In this spirit, fuzzy numbers are frequently employed to model linguistic abstractions mathematically. For example, *low*, *normal* and *high* are frequently defined as fuzzy numbers modelling one's understanding of a measurement reading being low, normal or high. The degree of membership of a measurement *x* in each of the fuzzy numbers is then a measure of compatibility of *x* with the semantic concept the number represents [Adlassnig 88]. Usually, the supports of such fuzzy numbers overlap so that *x* can be member of more than one fuzzy number. The expression

$$\{(low, \mu_{low}(x)), (normal, \mu_{normal}(x)), (high, \mu_{high}(x))\} ,$$

which is called a *level* 2 *fuzzy set* (a fuzzy set of fuzzy sets) [Dubois 80], then comprises the compatibility of a measurement *x* with all abstractions.

Let x be a variable and $\widetilde{A}$ be a fuzzy set of some base set X. One may then wonder what an expression of the form

$$x = \widetilde{A}, \tag{1.2}$$

denoting that the value of x is $\widetilde{A}$, means. If $\widetilde{A}$ were a number, then certainly this would mean that the variable has that number as its value. Analogously, if $\widetilde{A}$ were an ordinary set, the value of x would be that set. It is hence only consequent to take (1.2) as meaning that the value of x is indeed a fuzzy set, not some element thereof.

On the other hand, fuzzy numbers (which are fuzzy sets) are readily employed as an extension of intervals representing imprecision of measurements. In this context, (1.2) could be interpreted as denoting the range of values x could possibly have, together with a measure of possibility for each such value, which is given in the form of a possibility distribution defined by $\mu_{\widetilde{A}}(x)$. This variant reading is nicely supported by the fact that extension of real algebra to fuzzy algebra is an extension of interval algebra as used in tolerance analysis [Dubois 80].

However, in this usage the fuzzy set is a dynamic range restriction (as opposed to the static range restriction in the declaration of a variable, compare Section 2.2.1) that is more adequately expressed by

$$x : \widetilde{A} \tag{1.3}$$

meaning that $\widetilde{A}$ is the range of values x stands for. (The different interpretations of (1.2) and (1.3) have been the subject of much debate; see, for example, [Dubois 93] for an overview. Note that the interpretations of (1.2) and (1.3) are independent of $\widetilde{A}$ being a fuzzy number.)

1.2.3 Fuzziness in Medicine

Statistics and probability theory have long been the only branches of mathematics broadly appreciated by clinicians as providing useful tools. It is hence only natural that any new theory competing with probability theory will encounter considerable resistance.

Despite these traditional ties, researchers have made various attempts to introduce fuzzy set theory to medical decision making and other medical disciplines. Surveys on the first 20 odd years of fuzziness in medicine can be found in, for example, [Adlassnig 82] and [Maiers 85]. During the last decade, fuzzy set theory has not exactly taken medicine by storm; while it has been very successful in technical application domains (fuzzy control, particularly in Japan), medicine seems somewhat reluctant to take notice of its benefits.

Interestingly, Zadeh anticipated very early that medical diagnosis would be the most likely application domain of his theory [Zadeh 68]. However, despite its obvious expressive power in modelling vague concepts so typical of medical knowledge, fuzziness in medicine is still far from being mainstream. Quite to the contrary, after the flaws of MYCIN's certainty factor model have been identified and generally agreed upon, probability-based models such as Bayesian networks seem to have come to the fore again [Heckerman 92].

Besides, fuzzy set theory competes with a number of other upcoming theories taking account of uncertainty; among these, neural networks could develop as one of the most appealing to the medical community, particularly as it respects and learns from the still accepted superiority of the human mind over machine in diagnostic matters (see, for example, [Miller 92] for a review).

Consequently, it comes as no surprise that of all monitoring frameworks discussed in the present work none declares its use of fuzzy sets. There are, however, numerous dedicated solutions published that could be integrated in a monitoring framework allowing for fuzziness, for example, [Schecke 91, Greenhow 92, Sittig 92b, Drakopoulos 93].

1.3 Scope and Aim of Work

Uckun provides us with a simple classification scheme of intelligent monitoring projects [Uckun 93b]:

- monitoring tools and facilities,
- specialized monitoring systems, and
- general-purpose monitoring architectures.

My present work provides some of each:

- tools and facilities for trend detection, disease history tracking, etc.,
- a number of specialized monitors for intensive care and outpatient monitoring and
- a general framework for the construction of diagnostic monitors.

The aim of my work is to pave the way for the design and implementation of practical diagnostic monitors soundly embedded in a theoretical framework that stakes out the possibilities and limitations of the approach.

The work is solely dedicated to clinical monitoring; no claims concerning its transferability to other problem domains are being made. While the crucial role of time is heavily emphasized and taken account of by trend detection and state tracking, other, equally important, aspects of diagnostic monitoring are only touched on. In particular, all uncertainty-related issues other than vagueness are not addressed.

1.4 Outline

The present work is organized in a theoretical and a practical part. The theoretical part consists of two chapters. Chapter 2 develops a general framework of monitoring including formal representations of time-dependent information and general elements of information processing. Chapter 3 fills this framework with substance by presenting a variety of systems dedicated to various monitoring subtasks. Trend detection based on fuzzy parameter courses as introduced in Section 3.7 and tracking the patient's state with fuzzy automata as presented in Section 3.9 are the core novelties of my work.

The practical part is concerned with putting framework and systems into reality. Chapter 4 sketches the design of the software components required to compile an individual monitor. Implementational issues are only addressed to an extent deemed necessary to realize the specification of the theoretical part. Chapters 5 and 6 treat one application domain each: the former evaluates the practicability of the framework in densely sampled intensive care settings, highlighting the performance of trend detection and state tracking; the latter shows that trend detection works equally well in sparsely sampled data domains.

Chapters 7 and 8 conclude the work with a general discussion of the chosen approach and its performance in contrast with others.

Chapter 2

A Framework for Diagnostic Monitoring in the Medical Domain

Diagnostic monitoring is a generic problem that has many instantiations, each bringing its own, individual characteristic. The generic nature of the problem and its solutions is reflected in the following two definitions; it is important that they are differentiated.

Definition 2.1 (monitoring framework)
A *monitoring framework* is a conceptual environment for the development of monitors. It constrains the elements and structure of concrete monitors.
■

Elements of a monitor are the data to be processed and the means of data processing. Structure addresses the arrangement and linking of elements in forming a whole out of parts.

Definition 2.2 (diagnostic monitor)
A *diagnostic monitor* is a concrete device that interprets clinical time series of data. It is an instantiation of a monitoring framework individually designed to suit the needs of a particular monitoring problem.
■

The remainder of this chapter is organized as follows: firstly, a collection of requirements that both monitoring framework and diagnostic monitors have to fulfil is presented. Then, a primitive information model is defined that specifies the form of all information processed in a monitor.

The information model is tightly linked to data acquisition and representation, the theoretic framework of mapping real-world continuous observations into a digital form suitable for computers. The succeeding section on synchronization addresses problems of temporal coordination resulting from practically constrained data acquisition.

With systems the active parts of a monitor, the elements of information processing are introduced to the framework. The aggregation of systems can serve to model processes of the human body and to synthesize a diagnostic monitor designed to satisfy specific monitoring needs.

Some thoughts on context dependency and context-sensitive systems and a discussion of other monitoring frameworks and related work conclude this chapter.

2.1 Requirements

For a field as general as diagnostic monitoring a comprehensive and consistent requirement specification is too ambitious a task to be pursued here. Rather, the following list is an attempt to tackle the most important requirements.

Basic Requirements

1. *Generality*
   Any monitoring framework with a general claim must be flexible enough to cover a large spectrum of clinical monitoring tasks.
2. *Customizability*
   Hardly any two monitoring cases are identical, so that monitors must be adapted to the specific situation. This holds not only for the kinds of variables measured, but also, and more significantly, for their interpretation, which always depends on a clinical context (basic disease, treatment, etc.). While the mere technical devices monitoring a body signal (sensors, transducers, displays and primitive alarms) are mostly application-independent, any subsequent processing must adapt to the specificities of the monitored case if its behaviour is to be called intelligent.
3. *Explicit consideration of time*
   The role of time in clinical monitoring is so fundamental that it must pervade all elements of a monitor. Iterative evaluation of time-ignorant one-shot diagnostic systems is not sufficient.
4. *Information reduction*
   The most stringent need in close clinical monitoring is to reduce data streams to their significant information content. Ideally, a monitor produces temporally stable state abstractions that only change if a significant change has manifested itself in the patient's condition. This requirement subsumes tasks frequently regarded as stand-alone such as trend detection and intelligent alarming.
5. *Handling of missing values*
   Contrary to the need for data reduction is that for explicit handling of missing values. Missing values that cannot be reconstructed invariably introduce uncertainty into clinical monitoring.
6. *Integration of different data types*
   Both numeric and symbolic information occurs at all levels of monitoring. An integration of symbolic and numeric information beyond the mere separation into numeric (low) level and symbolic (high) level processing is therefore necessary.
7. *Integration of different sampling times*
   In a general monitoring situation, data is acquired at arbitrary and unpredictable times. To be able to relate data from different sources, a means of synchronization must be provided.
8. *On-line operability*
   Even if contemporary diagnostic monitors are not likely to be routinely used in clinical practice, for results to be transferable to practical domains they must be obtained

under on-line conditions, i.e., data must be accessed in the same order it becomes available and output must not rely on a look-ahead in the data stream. The meeting of real-time constraints, although a crucial requirement for all practical deployment, seems not so important at this current stage of development of intelligent monitors; rather, "make it work first before you make it work fast"[1].

Implications

The above requirements have practical implications which themselves can be added to the requirement list. Among those are:

1. *Openness*
   For a monitoring framework to be general it must be open to integration of methods of most different origin. Ideally, a new monitoring component only needs to be "plugged in" to assume operation.
2. *Definition of a uniform interface*
   The definition of a uniform interface shared by all monitoring components is the prime implication of openness and generality. Such an interface specifies the data to be transported, the functionality provided, and the synchronization mode.
3. *Provision of a tool set of monitoring components*
   The design of an individual monitor is greatly facilitated by the availability of standard monitoring components. These components can be readily usable or provide templates that need to be instantiated (by setting certain parameters determining their operation) to fulfil a particular monitoring subtask.
4. *Implicit control flow*
   While the static design of a monitor is guided by the monitoring problem and should be comparatively easy, the dynamic interaction of its components can become arbitrarily complex. In order not to burden the designer with control flow issues, the control flow should be implicit or at least determined by the static structure of the monitor.
5. *Self-explanatory components*
   The display of all output derived by a monitor is essential for its transparency and trustworthiness. If no assumption about the configuration of a particular monitor can be made, no standard user interface can be provided. In order not to require an explicit definition of the user interface of each monitor (which will usually mirror the structure of the monitor and thus imply largely redundant work), the components need to be self-explaining and self-displaying.
6. *Abstraction hierarchy*
   Abstraction is *the* human means of dealing with complexity. Data reduction should thus be guided by the desire to achieve clinically significant, temporally stable abstractions. However, because high-level abstractions necessarily hide information, they may require explanation. This explanation is best provided by offering information at different levels of abstraction. Organizing a monitor as an abstraction hierarchy does not only provide this service automatically, it also reflects the natural data flow in modularized signal interpretation.

Some readers may miss implications they find worthy of mention. For example, one could argue that deployment of some kind of model of the system being monitored is inevitable if monitoring is to succeed in highly complex tasks. On the other hand, there is always the danger of requirements being biased by the solutions one has in mind[2] and I decided to leave out all paradigmatic issues.

[1] quoted in [Bentley 88]

[2] Admittedly, the above list is not free of this.

Restrictions

Quite clearly, the list of requirements and its implications is superficial, and any practical attempt to create a monitor general enough to satisfy all will encounter several major issues not addressed. To reduce this underspecification of monitoring demands and to be able to accomplish all stated requirements with limited effort, I make the following restrictions:

1. *Chronological data stream*
   All data processed by a monitor must be available in strictly chronological order. No proper regard of predated data, i.e. data received subsequent to its occurrence, is possible (compare discussion in Section 2.8.1).
2. *Acyclic data flow*
   To release monitor design from stability considerations, the data flow through the monitor is restricted to be acyclic. This means in particular that no feedback is allowed.

2.2 Information Model

It is a simple but fundamental observation that in a monitoring environment the patient is the sole object of concern [Steimann 94b]. The richness of sophisticated information models employing inter-object relations or type subsumptions contributes little to an adequate representation of the patient. Rather, all data relevant for monitoring can be represented in the form of *variables*.

2.2.1 Variables

In a clinical setting the patient is sufficiently described by a number of attributes, directly observable or derived properties expected to reveal his clinical status. Because the values of these attributes are patient-dependent and vary with time, I will call them *variables*[3] and model them as functions mapping patients and time to a respective attribute value space:

Definition 2.3 (variable)
A *variable* is a function

$$x : P \times T_c \rightarrow V_x \tag{2.1}$$

where

x	is the name or *label* of the variable,
P	is the set of patients,
T_c	is a totally ordered set modelling continuous time (usually the real line), and
V_x	is the codomain or *value range* of x.

■

The value of a variable x for a patient $p \in P$ at time t_0 is denoted by

$$x(p, t_0).$$

Following mathematical convention, t_0 is used here to denote one distinct, yet arbitrary, point from the time-continuum T_c. The expression

$$x(p, t_0) = v \tag{2.2}$$

with v being some value from V_x explicates the value of x at t_0. (2.2) is a special form of the quadruple

$$(x, p, t_0, v)$$

[3] Synonyms frequently found in the literature include *parameters*, *attributes*, *features*, *qualities*, *measurements* etc.

used by other authors (e.g., [Haimowitz 94, Sadegh-Zadeh 94]). However, (2.2) explicates the function property of *x*: every variable of a patient has one and only one value at one time, or, more formally, for any two quadruples (x_1, p_1, t_1, v_1) and (x_2, p_2, t_2, v_2) from $x_1 = x_2$, $p_1 = p_2$ and $t_1 = t_2$ it follows that $v_1 = v_2$.

(2.2) is also called *declaration of a variable*; apart from introducing its name, it specifies the data type of the variable, i.e. the set of values *x* can take on together with the operations that can be performed on it.

2.2.2 Data Types

In a monitoring environment, most properties observable in a patient are quantitative, i.e., numeric. Although numeric coding of non-numeric (qualitative, comprising nominal and ordinal) variables (e.g. *male* = 0 and *female* = 1) is still common practice, doing so obscures the different properties of variables. For example, body temperatures and heart rates can be arithmetic operands, while sexes and names cannot. To take account of these differences, different data types are distinguished in this framework. These are:

- *numeric*
  providing standard arithmetic operations as well as ordering and equality relations,
- *degree*
  as a generalization of the truth (Boolean) data type providing logical connectives,
- *symbolic*
  as a general means to encode abstractions as well as non-numeric properties, and
- *fuzzy set*
  as a generalization of set-valued functions extended to capture gradation.

Numeric

Numeric is the natural data type of the vast majority of directly observable variables. Most numeric variables are continuous in nature [Challis 90], which informally means that their potential for change is limited by time. Discontinuity is mostly a result of quantization and thus artificial; the quantization error is usually small and assumed to be negligible for the scope of this work.

Example

Despite its continuous progression, the age of an adult patient is usually stated in years, which is granular enough to take account of age-dependency. Drug dosages and apparatus settings are usually administered in discrete units, keeping the number of possibilities to be considered small.

◊

The natural total ordering of numeric values is supplemented by difference as a measure of distance so that not only the direction, but also the speed of change can be derived. Notions of slope and trend are therefore natural features of numeric variables.

Degree

A degree represents the degree of fulfilment of a proposition. The data type *degree* will therefore be associated with variables of propositional or binary character. A degree is represented by a value between 0 and 1. The extreme degrees 0 and 1 correspond to the classical Boolean values *false* and *true*, respectively. The Boolean data type may thus be regarded as a special case of *degree*.

Example

Tachycardia is a finding associated with an increased heart rate. If the abnormality of the

heart rate of a patient is only moderate, then tachycardia might be present with a degree of, say, 0.5. This is encoded by the expression

$$tachycardia(p, t_0) = 0.5.$$

◊

For the relation of degree-typed variables and fuzzy sets see below. The employment and interpretation of degrees includes *degree of match*, *degree of compatibility*, *measure of similarity*, or even *degree of illness* (compare [Adlassnig 82]).

Symbolic

The *symbolic* data type provides the common basis for encoding inherently discrete observations and derived abstractions. It resembles the PASCAL programming language enumeration type.

Typical examples of variables with symbolic value ranges are *sex* and *ventilator mode*[4]. Obviously, there is no natural ordering between the ranges' elements *male* and *female* or *controlled mandatory ventilation* and *assist mode*. Also, differences between symbols are clear-cut and the discontinuity between sexes and ventilator modes is very natural. These variables are inherently symbolic: they are observed as symbols. I will therefore call them *observed-symbolic*.

Frequently, however, symbols are employed as abstractions of inherently numeric variables. These abstractions are either obtained by assigning qualitative terms to typical ranges of values, or by associating a distinct (physiological or pathophysiological) state with a characteristic constellation and history of values. Because these variables are derived from data originally present in a different form, I will call them *derived-symbolic*.

Derived-symbolic variables that encode observations qualitatively are the basis of qualitative reasoning. This approach exploits the fact that every symbol represents a certain contiguous range of values that can be calculated with. On the other hand, derived-symbolic variables that encode states are used to model natural disease histories [Coiera 89] by specifying a sequence of states, transition among which is characterized by well-defined conditions.

As will be shown, discontinuity of derived-symbolic variables is not always natural and can cause problems, particularly if it is the result of arbitrary abstraction of an application domain lacking clear-cut definitions such as the medical. Fuzzy set types as presented next are introduced to overcome this deficiency.

There is a certain relation between symbolic and degree types: any symbolic-typed variable $x : P \times T_c \rightarrow \{s_1, \ldots, s_n\}$ can be transformed to n degree-typed variables $s_i : P \times T_c \rightarrow [0, 1]$, where

$$x(p, t_0) = s_i \quad \Leftrightarrow \quad s_i(p, t_0) = 1, s_j(p, t_0) = 0 \quad \text{for all } 1 \le j \le n, j \ne i,$$

i.e., a symbolic type is equivalent to a set of mutually exclusive Boolean types.

Example

An observed-symbolic variable describing the ventilator mode

$$mode : P \times T_c \rightarrow \{CMV, ASSIST, T-PIECE\}$$

relates to three degree-typed variables

$$CMV : P \times T_c \rightarrow [0, 1]$$
$$ASSIST : P \times T_c \rightarrow [0, 1]$$
$$T-PIECE : P \times T_c \rightarrow [0, 1]$$

[4] Although it may be argued that *ventilator mode* is not a property of the patient, the ventilator is explicitly dedicated to the patient and its variables can in fact be regarded as the patient's, e.g., "the patient is ventilated in assist mode".

The fact that *mode* is symbolic suggests that the states encoded by *CMV, ASSIST* and *T-PIECE* are mutually exclusive. Consequently,

$$mode(p, t_0) = CMV \quad \Leftrightarrow \quad CMV(p, t_0) = 1, ASSIST(p, t_0) = 0, T-PIECE(p, t_0) = 0$$

etc.

◊

Fuzzy Sets

The inclusion of fuzzy sets in the information model is a tribute to the medical domain's lack of crisp and universally applicable borders. Fuzzy sets are employed to weaken the all-or-nothing nature of derived-symbolic abstractions. They enable smooth transition from one concept, be it a qualitative term or a physiological state, to another.

Fuzzy set-valued variables encode the position of the variable's value in the space between discrete landmarks. Because they maintain the notion of symbols while introducing the concept of graduation, they combine features of the continuous observation and its derived-symbolic abstraction.

Example

The qualitative terms *normal, low* and *high* for body temperature are not only weakly defined, they are also too broad for many diagnostic problems. An actual body temperature of 38°C could thus be more adequately encoded by the fuzzy set

$$\{(low, 0.0), (normal, 0.2), (high, 0.6)\},$$

meaning that the actual temperature is neither entirely compatible with *normal* nor with *high*, yet closer to *high*.

◊

Operations defined for fuzzy sets are the membership test, i.e., the evaluation of the membership function for a certain element, and the fuzzy set-theoretic operations of Section 1.3.1.

Analogously to symbolic variables, a fuzzy set-typed variable is related to degree-typed variables such that

$$x(p, t_0) = \{(s_i, \mu_i) | 1 \le i \le n\} \quad \Leftrightarrow \quad s_1(t_0) = \mu_1, \ldots, s_n(t_0) = \mu_n |.$$

Example

Consider a symbolic variable declared as

$$oxygenation : P \times T_c \rightarrow \tilde{\wp}(\{hypoxemia, normal, hyperoxemia\})$$

and three degree-typed variables

$$hypoxemia : P \times T_c \rightarrow [0, 1]$$
$$normal : P \times T_c \rightarrow [0, 1]$$
$$hyperoxemia : P \times T_c \rightarrow [0, 1]$$

Transferring the (semantic) equivalence of the previous example to the fuzzy case, the observation

$$oxygenation(p, t_0) = \{(hypoxemia, 0.3), (normal, 0.7), (hyperoxemia, 0.0)\}$$

is equivalent to

$$hypoxemia(p, t_0) = 0.3$$
$$normal(p, t_0) = 0.7$$
$$hyperoxemia(p, t_0) = 0.0$$

◊

2.2.3 The Course of a Variable

Although both patient and time are independent variables, in diagnostic monitoring one is usually only interested in the dependence on time, as monitoring does not deal with analysing the change of variables across patients. Specification of the patient will therefore be omitted throughout the following. However, one should bear in mind that each variable also depends on the patient, but that this patient remains the same throughout a monitoring case and is hence considered implicit.

The development of a variable over time is referred to as its *course*. Obviously, the course of a variable contains more information than its instantaneous value, and it is characteristic of the monitoring task to deal with courses rather than individual snapshots in time.

$$x(t)$$

denotes the continuous-time *course of the variable x*. $x(t)$ is also called *signal*—although both terms are used as synonyms throughout this thesis, *signal* is the preferred choice in technical contexts, whereas it is *course of a variable* in medical terms.

Note that $x(t)$ only denotes the course of the variable—it does not specify it. An expression of the form

$$x(t) = f(t)$$

where f is a function well-defined on T_c, specifies the value of x at all times, i.e.,

$$x(t_0) = f(t_0) \quad \text{for all } t_0 .$$

Example

The sex of a female patient is specified by

$$sex(t) = female$$

where *female* is a function constant in t. The body temperature, however, although certainly defined at all times, is difficult to specify continuously.

◊

In the real world, time is, or at least seems, continuous, and so are most physiologic variables. In the computer, both time and values must be represented in a digital form. While the conversion of continuous to discrete values is of little interest to this work, the transition from continuous to discrete time is crucial and the subject of the following.

2.3 Data Acquisition and Representation

Data acquisition, the first step in digital information processing, has strong technical aspects. However, this section is only interested in the outcome of data acquisition, i.e. in its informational aspects.

2.3.1 Sampling

By definition, sampling is the conversion of a continuous-time to a discrete-time signal. While the continuous-time signal is thought of as a function, its sampled discrete-time counterpart is represented by a *sequence of samples*

$$\langle x[t_n] \rangle ,$$

where each $x[t_n]$ denotes an individual sample taken at time t_n, the *sampling time*. Square brackets are used to indicate the discreteness of the independent variable, which may in fact be regarded as an index to an array rather than an argument to a function.

Formally, a sequence is a function of integers. However, recalling that x itself is a function of time one finds that $x[t_n]$ depends on t_n and only indirectly on n. Directly dependent on n is $\langle t_n \rangle$, a sequence of sampling times in chronological order ($t_n < t_{n+1}$ for all n) which I call the *sampling sequence*. x applied on $\langle t_n \rangle$ is then a sampled version of $x(t)$ as defined by

$$x[t_n] = x(t_n).$$

Figure 2.1 a) and b) help to explain the terminology.

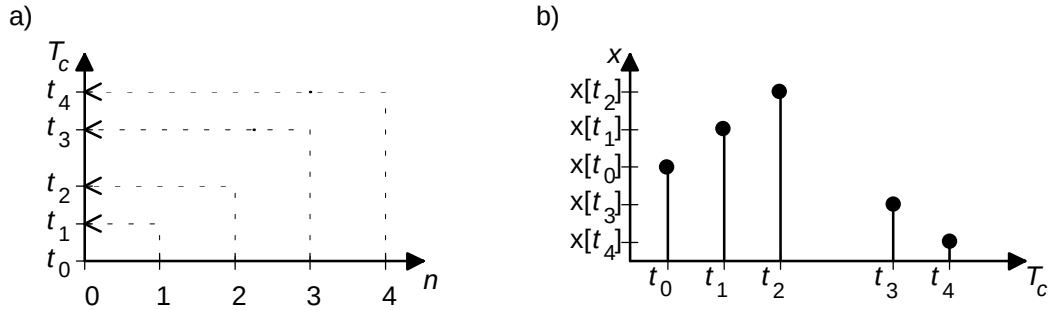


Figure 2.1: Sampling
a) sampling sequence as a function of n
b) sequence of samples

If $t_{n+1} - t_n$ is constant for all n, sampling is said to be *periodic* with a *sampling period* $T = t_{n+1} - t_n$, $1/T$ is called the *sampling rate* or *frequency*, and $\tau = t_0$[5] is the *phase-shift* of sampling. In the literature, periodic samples are often denoted by $\langle x_n \rangle$, where

$$x_n = x[t_n] \quad \text{with } t_n = nT + \tau.$$

For practical reasons, periodic sampling is very rare in the medical environment. Even if samples are taken automatically as is the case in intensive care or operating room monitoring, artefacts, interventions, or equipment failure will lead to missing data and thus aperiodic availability of sampled values. It is therefore not realistic to assume that any sampling in the medical field is truly periodic.

Arbitrary signals and sampling sequences provided, it should be clear that the same sequence of samples can be the result of sampling of many different signals. In other words: from a sequence of samples the original signal can generally not be reconstructed. The following two subsections are dedicated to this problem.

2.3.2 Explaining Courses

A function is called an *explaining course* $x_e(t)$ of a sequence of samples $\langle x[t_n] \rangle$ if and only if

$$x_e(t_n) = x[t_n] \quad \text{for all } t_n, \tag{2.3}$$

i.e., if $x_e(t)$ sampled with the sampling sequence $\langle t_n \rangle$ yields $\langle x[t_n] \rangle$. $x_e(t)$ is then one of (usually infinitely) many possible courses that would explain the findings $\langle x[t_n] \rangle$. Note that the original signal is always an explaining course of its samples.

Figure 2.2 shows a selection of explaining courses for three consecutive samples.

[5] In the context of sampling, t_0 is the sampling time associated with $n = 0$. It is not to be confused with t_0 in $x(t_0)$.

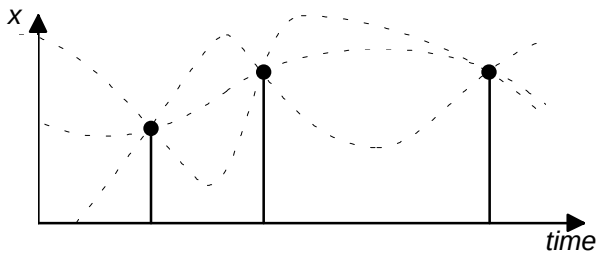


Figure 2.2: Three different explaining courses for a sequence of three samples

Explaining courses appear to be purely theoretic, yet they represent the problem space when reasoning about the true course of a sampled variable. In particular, explaining courses are the basis of trend detection as described in Section 3.7.

2.3.3 Signal Reconstruction

Whereas explaining courses merely represent hypotheses, signal reconstruction aims at reproducing the true course of the variable sampled. Quite obviously, any perfect reconstruction must satisfy (2.3), i.e., explain the findings.

Intuitively associated with sampling is loss of information: by recording the course of a variable at certain points in time only, intermediate information is lost, so that one would not expect to be able to reconstruct the original course without error. However, simple considerations immediately show that if the variable can be described by a specific analytical model, few samples may suffice to reconstruct the original course perfectly.

Example

If the course of a variable is known to be constant over a certain interval, a single sample $x(t_0) = c$ from that interval is all that is required to reconstruct the original signal as $x(t) = c$ over that interval. Accordingly, the course of a linearly evolving variable is sufficiently specified by two samples.

◊

In general, if a parametrical description of the course of a variable over an interval is known to apply, the number of samples required to determine these parameters is sufficient to reconstruct the original course perfectly. Applicability of a well-specified model hence compensates for ignorance during unsampled intervals.

Perfect Reconstruction and the Nyquist-Frequency

One special case of parametrical specification of signals is the Fourier transform of bandlimited, periodic signals. The sampling theorem guarantees that if the sampling frequency is at least twice as high as the highest frequency component in a bandlimited signal, then this signal can be reconstructed without error. (For a more thorough treatment, the reader is referred to standard digital signal processing literature, e.g., [Lee 88, Oppenheim 89].)

Imperfect Reconstruction: Interpolation

Unfortunately, mathematical models for biomedical signals are hard to identify. In fact, as long as the nature of the underlying physiological process is unknown, there is little reason to believe that it is (reasonably) bandlimited or that high-order polynomial models describe the variable's course more adequately than (piecewise) linear [Avent 90].

Linear interpolation appears to be the most popular method[6] in use: it reconstructs the missing parts of a signal between two successive samples by a straight line. Obviously, interpolation

[6] That it is also quite successful has been demonstrated in [Albridge 88].

must be replaced by extrapolation in on-line reconstruction and is not applicable when sampling is sparse (see below for definition of dense and sparse sampling).

Exponential and periodic (trigonometric) models of a signal can also be made the basis of interpolation.

Implicit Reconstruction: Trend Detection

Explicit reconstruction of the original course is not always necessary (nor even desirable) in diagnostic monitoring: recall that it is one purpose of diagnostic monitoring to "abstract away" short-term fluctuations, as would be revealed by high-frequency sampling and (near) perfect reconstruction. If one is only interested in an abstraction of the course, explicit reconstruction need not be done as long as the samples taken uniquely identify that abstraction. Such is the case in trend detection: if a certain trend is identified in a variable, reconstruction of the missing intervals is implicit in the assessment of the variable as following that trend.

2.3.4 Dense and Sparse Sampling

A quality commonly associated with sampling is its density, which is usually classified as either sparse or dense. At first glance, classification could be based on the sampling rate, where dense sampling would be a matter of, say, seconds, while sparse would be one of hours, weeks, or even years. However, as the following example demonstrates, this definition is at odds with the intuitive understanding that a dense sample is sufficient to reveal what one is looking for, while a sparse one lacks information and is consequently not, or, at least, not always.

Example

> A sampling period of three months may be perfectly acceptable to detect growth disorders in juveniles, so that respective sampling would be classified *dense*. In contrast, the sampling period of one day for blood gases, although much shorter, may be far too long not to miss life-threatening perturbations in an intensive care setting, so that it would be classified *sparse*.

◊

Justified by this example, the following definition seems reasonable:

Definition 2.4 (dense and sparse sampling)
Sampling is called *dense* if the original signal can always be reconstructed from the samples with acceptable accuracy, and *sparse* otherwise. Accuracy is acceptable if the difference of original and reconstructed signal is marginal, i.e., if the original and the reconstructed signal have essentially the same meaning with respect to the given monitoring problem.
■

[Haimowitz 94] adds that sampling may also be considered sparse if a decision must be made long after the last, yet still before the next sample is available.

According to this definition, classification of the sampling rate depends on the availability and choice of an interpolation method, which may be explicit or implicit (see previous section). In practice, sampling may be regarded as dense when one is confident that nothing significant can occur (variable changes steadily) in between any two consecutive samples, so that linear interpolation seems acceptable for signal reconstruction.

Note that the definition does not rely on the availability of equidistant samples, although it suggests that the unspecified gap should be of limited size for an interpolation method to be applicable.

In the medical domain dense sampling, although always desirable, is sometimes too costly, dangerous, or simply impossible due to technical reasons. A typical contemporary intensive

care setting therefore provides a combination of both: dense samples with limited diagnostic value are sparsely supplemented by expensive, yet more informative, tests.

Example

Oxygenation of patients with respiratory deficiency can densely be monitored through pulse oximetry determining arterial oxygen saturation. However, because neither pulse oximetry is very reliable nor saturation is of high diagnostic value, the assessment of oxygenation is supplemented by examination of blood samples drawn at longer intervals.

◊

2.3.5 Missing Values

Sampling sequences disregarding the technical availability of a variable may lead to sampling attempts at times at which the variable's value cannot be determined. In order to be able to record this situation explicitly, i.e., rather than eliminating these times from the sampling sequence, I use the symbol $\perp$ for *unknown* or *missing value* and write

$$x[t_n] = \perp. \tag{2.4}$$

Deviating from the mathematical interpretation, $\perp$ does not mean that there is no value—rather, one may assume that the variable has a value at that time, but which one is not known.

Example

Endotracheal suctioning manoeuvres interrupt mechanical ventilation so that variables determined by the ventilator are rendered invalid. For example, if the manoeuvre is performed from t_i to t_j,

$$respiratoryRate[t_n] = \perp \quad \text{for } t_i \le t_n \le t_j.$$

◊

Theories of persistence of observations (e.g. [Dean 88, Shahar 92a]) can help to reduce uncertainty imposed by missing values, as do interpolation methods. However, as most methods can only approximate the true value for periods of ignorance, more sophisticated information models would employ a concept of gradual uncertainty covering the whole range from exact knowledge of a value via vague restrictions to total ignorance, as realized for the probabilistic case in VENTPLAN [Ruttledge 90a, b] and outlined for the fuzzy case in [Steimann 94b].

Note that missing values are not only a consequence of the failure to sample: data validation or handling of interventions as sketched in the above example may classify an acquired value as invalid and, in the absence of an applicable reconstruction method, render it unknown.

2.4 Synchronization

Data sampling times cannot always be dictated by the monitor; rather, the monitoring environment determines the times at which data becomes available. If several uncoordinated sampling devices are employed, different variables are invariably sampled at different times, thus making explicit synchronization of data inevitable.

2.4.1 Changing the Sampling Sequence

Changing the sequence based on which a variable was sampled may be desirable for several reasons. The prime purpose within this work is to be able to relate, i.e., bring together different variables sampled at different sampling sequences.

Ideally, the sampling sequence is changed by first reconstructing the original signal $x(t)$ and then sampling this signal with the new sequence $\langle t_m \rangle$. In the absence of an applicable reconstruction method the intermediate representation

$$x_i(t) = \begin{cases} x[t_n] & \text{if } t = t_n \\ \perp & \text{else} \end{cases}$$

may be thought of as the reconstructed signal that, when sampled at times for which $x(t_m) = \perp$, yields (2.4).

Note that if $\langle t_m \rangle$ contains sampling times not contained in $\langle t_n \rangle$, $\langle x[t_m] \rangle$ necessarily includes samples with missing values. Changing the sampling sequence is lossless if $\langle t_n \rangle$ is a subsequence of $\langle t_m \rangle$, i.e., if $\langle t_m \rangle$ contains at least all samples of $\langle t_n \rangle$.

2.4.2 Global Sampling Sequence

In clinical practice, sampling of different variables, no matter whether automatically or manually, is performed at different rates and phase-shifts. Recalling that relating variables may require concurrence, there is an obvious need for global coordination and synchronization of sampling sequences within a monitor.

Theoretically, synchronization can be achieved by changing the sampling sequences of all involved variables to one global or monitor-wide sampling sequence $\langle t_M \rangle$, which is preferably chosen so that no sample is lost. Such a sequence exists; it can be constructed by merging the individual sampling sequences of all variables into one. In practice, however, this approach may appear too rigorous: different phase-shifts or slightly different sampling rates unnecessarily boost up the sampling sequence. In these cases, common sense can be applied to find a practical sampling sequence. For example, a predetermined temporal grid can be defined, and all sampling times can be rounded to the nearest time of the grid.

Example

In an on-line monitoring setting with multiple devices connected to a patient, sampling rates of each device may be set to one sample per ten seconds. However, unless all devices sample at precisely the same time, each device contributes its own sampling time to the global sampling sequence. Adhering to a temporal grid of six samples per minute, however, each measured variable would be assigned to the nearest *n*-fold of ten seconds.

◊

2.5 Elements of Information Processing: Systems

The previous sections dealt with the acquisition and representation of patient-specific information in a monitoring environment. The purpose of diagnostic monitoring, however, is to process this information so as to generate high-level abstractions that allow proper interpretation of body signals. This processing is delegated to systems as the fundamental units of information processing.

Definition 2.5 (system)
A *system* implements a transformation that maps a number of input variables to a number of output variables.

■

Systems have been named *mechanisms* [Doyle 89], *modules*, *processes* [Russ 86, Factor 90], etc. Formally, a system with *j* input and *k* output variables is declared by

$$x_1(t), \ldots, x_j(t) \rightarrow y_1(t), \ldots, y_k(t)$$

in the continuous-time and

$$\langle x_1[t_n]\rangle, \ldots, \langle x_j[t_n]\rangle \rightarrow \langle y_1[t_n]\rangle, \ldots, \langle y_k[t_n]\rangle \tag{2.5}$$

in the discrete-time case, the latter of which we are primarily concerned with. Note that the mapping involves the whole sequence rather than single samples; the output of a system may depend on input at any (including all) times. Graphically, systems are commonly depicted as block diagrams as shown in Figure 2.3. Input and output sampling sequences of a system are usually identical—in an on-line employment, a system produces a new output each time a new input becomes available.

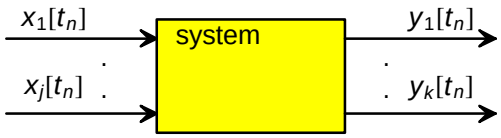


Figure 2.3: Block diagram of a system

Note that (2.5) *declares* a system—it does not *specify* it. A system may be specified using any algorithmic description; the most simple form is an equation expressing an output sample in terms of input samples.

Example

The net change of body temperature from one sample to the next may be derived by a system specified as

$$temperatureChange[t_n] = temperature[t_n] - temperature[t_{n-1}]$$

◊

To express that a particular output depends on subsequences of a system's input, I write

$$y[t_n] \leftarrow \langle x_1[t_m] \,|\, p_1(t_m)\rangle, \ldots, \langle x_k[t_m] \,|\, p_k(t_m)\rangle$$

where the p_i are predicates constraining the sampling times. In the following, the output is assumed to be dependent on one input variable—the definitions can be extended to many wherever required.

2.5.1 Coupling and Aggregation of Systems

Quite obviously the output of one system can serve as the input of another. Congruence of the respective input and output data types provided, systems can therefore be aggregated to form compounds with a more complex behaviour. The aggregation of systems also has input and output so that it may itself be regarded as a system. Figure 2.4 shows three coupled systems that form an aggregation. To indicate that each system contributes to raising information to higher levels of abstraction I arrange them vertically, with the input located at the bottom and the output placed at the top.

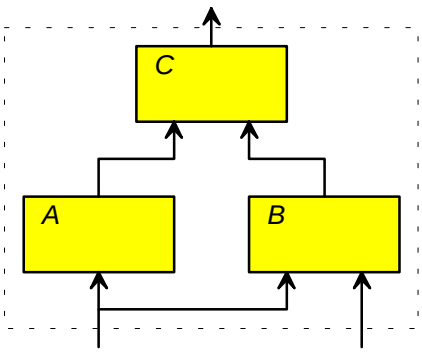


Figure 2.4: Aggregation of coupled systems

Within the scope of this work coupling of systems is restricted to be acyclic (compare Restriction 2 of Section 2.1). An aggregation of systems must therefore not contain feedback loops. This restriction is motivated by computational considerations: feedback requires iterated evaluation of systems on the same input and is always a potential source of instability.

2.5.2 Classification of Systems

The notion of system is very general; its liberal definition allows just about any transformation from input to output to be regarded as a system. Classification of systems is hence arbitrary—many properties can be isolated to form the basis of classification. Most of the following properties originate from signal processing theory (see, for example [Oppenheim 89]) and address the dependence of a system's output on a specific subset of its input.

A system for which a time shift in its input leads to a corresponding time shift of its output is called *time-invariant*. Time-invariance is a necessary property of all systems treated in this work.

Systems that regard only past and current input to derive their output, reflected in

$$y[t_n] \leftarrow \langle x[t_m] \,|\, t_m \le t_n \rangle$$

are called *causal*. Some causal systems' output depends on their concurrent input only, i.e.,

$$y[t_n] \leftarrow x[t_n].$$

Because these systems need not memorize the input they have been presented with before, they are called *memoryless*. For reasons given below, I will call causal systems that are not memoryless *history-sensitive*.

Memoryless systems are very unproblematic, which is probably why they are frequently being used in monitoring projects. Their inability to account for temporal developments, however, makes all attempts to tackle nontrivial monitoring problems with memoryless systems alone appear impractical.

Some systems derive their output from a bounded neighbourhood of their current input. This neighbourhood is specified by an interval $[w^-, w^+]$ called *window* such that

$$y[t_n] \leftarrow \langle x[t_m] \,|\, t_n + w^- \le t_m \le t_n + w^+ \rangle,$$

which is depicted in Figure 2.5 a). Such systems are called *window-based*.

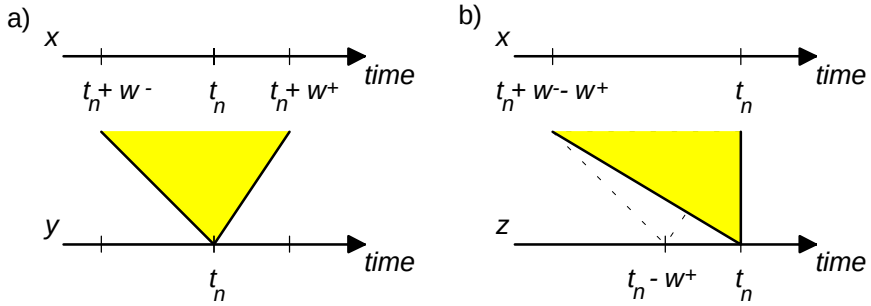


Figure 2.5: Window-based noncausal system and its causal, delayed equivalent

Note that if w^+ is negative, the corresponding system is causal (and thus history-sensitive). However, most window-based systems require a symmetric window and are hence noncausal, e.g. moving average and median filters (see below). As a by-pass, every noncausal window-based system can be transformed to a causal one

$$z[t_n] \leftarrow \langle x[t_m] \,|\, t_n + w^- - w^+ \le t_m \le t_n \rangle$$

by translating its window by $-w^+$ to $[w^- - w^+, 0]$ as done in Figure 2.6 b), which is equivalent to the noncausal system cascaded with the delay

$$z[t_n] = y[t_n - w^+],$$

provided that the translated system is time-invariant.[7]

A *recursive* system derives its output from concurrent input and its previous output. This definition implies that recursive systems can only be applied in the discrete-time domain, as *previous* is not defined for the time-continuum. Recursive systems require minimum memory and computation; they are therefore particularly well-suited for real-time monitoring.

Note that recursion is not to be regarded as a feedback loop and thus does not violate the requirement of acyclic coupling. Rather, the previous output is held in the system's memory; a recursive system is thus history-sensitive.

Finally, *context-sensitive* systems are systems that process some of their input variables in a context set by others. Context dependency and context sensitivity will be addressed in Section 2.7 below.

2.5.3 Prospective and Retrospective Employment of Systems

Basically, transformation of input to output can be implemented as a sequential process, which means that the output sequence $\langle y[t_n] \rangle$ is derived one by one by incrementing n and so proceeding through time.

In an on-line environment, employment of systems is inherently prospective. t_n is then linked to the current time and future input at times $t_m > t_n$ is yet unknown. All systems used in on-line monitoring must therefore be causal.

On the other hand, off-line processing of recorded time series is retrospective in nature. Availability of data beyond t_n allows a look-ahead in the data stream and hence more powerful (i.e., faster and more accurate) systems.

Example

A one-step-ahead predictor in an off-line operated monitor can be implemented as

$$\hat{x}[t_n] = x[t_{n+1}],$$

which obviously outperforms any causal system for prediction.

◊

Although all systems put forward in this work were evaluated retrospectively, none utilizes a look-ahead, i.e., all systems are causal.

2.6 Human Body and Monitor as Systems

On the highest level of aggregation two systems can be identified in the medical monitoring setting: the human body and the diagnostic monitor. The coupling of both is depicted in Figure 2.6. Note that within this framework coupling must be acyclic; this means that even if the choice of therapeutic actions taken by the medical staff is influenced by the monitor's output, the control loop is always open.

[7] Note that this equivalence is an ideal assumption holding only in the case of periodic sampling; a delay of aperiodic sequences requires special treatment (see Sections 3.1 and 3.5).

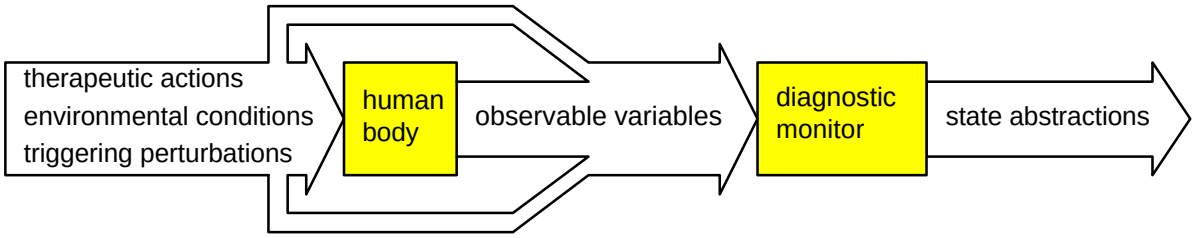


Figure 2.6: Body and monitor as coupled systems

The behaviour of the human body is influenced by many external factors. Therapeutic actions include machine settings, drug dosages and infusions. Among environmental conditions are sociological factors, nutrition and exercise. Perturbations such as dysfunction, infections and traumata are considered triggering if they lead to conditions with their own dynamics which can only indirectly be related to these causes.

Example

The adult respiratory distress syndrome (ARDS) is a form of lung failure triggered by one or more of a number of initiating events (including sepsis, polytraumas, burns, etc.). Independent of the original perturbation, the syndrome manifests itself in lung oedema caused by pulmonary capillary leak. [Niemer 92]

◊

Observability of a variable in Figure 2.6 is determined by the availability of a measurement method. Variables resisting direct observation may be observed indirectly by monitoring their effects on other variables. The diagnostic monitor in turn must derive temporarily stable state abstractions from both directly and indirectly observed variables. It relies not only on the variables found in the patient, but also on all known external influences on the human body.

The human body can be thought of as a system composed of specialized subsystems that interact and so try to maintain certain equilibria and balances of variables. It is not only influenced by external factors, but also hosts various internal signal sources: the autonomic nervous system, although influenced by environmental and internal factors, and genetic coding determining obvious specificities such as sex as well as subtle idiosyncrasies provide for and influence a vast number of variables, as indicated in Figure 2.7.

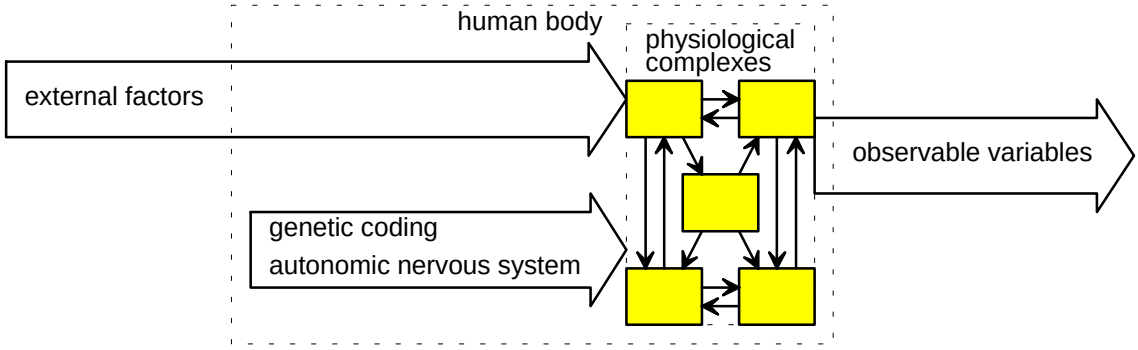


Figure 2.7: Internal and external influences on variables

A diagnostic monitor on the other hand is a hierarchical aggregation of elementary systems arranged to derive the desired output. Figure 2.4 depicts such an aggregation. The lowest level of systems is determined by the directly observable variables, while the highest level is oriented towards the need for informative, yet easily conceptualized, information comprising the current and past status of the patient. The design of a diagnostic monitor should reflect the structure of the monitored problem; it may then be regarded as a model of medical knowledge in the respective domain.

Implied by their nature, body and monitor have very different characteristics, and so have input and output of both. In short, the human body as a system is characterized by an extremely complex behaviour resulting from mutual dependency among most of its many variables. Any practical model of the body is inevitably a simplification. This is why expected and observed behaviour of the body are often at odds, and, although its behaviour is assumed to be causal, it often appears to be indeterministic, i.e., random.

On the contrary, the monitor as a technical device is usually deterministic.[8] Its behaviour is totally specified by its implementation and therefore predictable and transparent. Its output is solely dependent on its input—ideally, no internal parameters influence its diagnosis. This contrast hints on the problems of monitoring: all models employed in a monitor are necessarily abstractions and hence only approximations—deviation of the observed variables from model-predicted values may be due to inadequacy of the model or to perturbations, a dilemma that cannot be resolved unless the precision of the model can be specified in absolute terms.

2.7 Context Dependency and Context Sensitivity

Sooner or later all practical approaches to monitoring will be confronted with the problem of context dependency of body variables. This is so because the meaning of certain variables' values can only be assessed correctly in relation to their context.

Example

Fluid balance and dosages of all drugs affecting the cardiovascular system must be regarded as the context of blood pressure.

◊

As Figure 2.7 suggests, there are many factors that potentially influence the value of a variable: some are environmental, others inside the body; some are measurable, others can only be speculated about; some have a definable impact, the effect of others is not or only vaguely known. Adequate regard of context dependency must therefore be flexible enough to cover all facets of the problem.

On the other hand, context dependency must carefully be distinguished from other functional dependencies. The fact that a variable depends on others is often the reason why this variable is being monitored: by observing its value one hopes to obtain information about some underlying perturbing process which itself is not directly observable. The monitored variable is then regarded as *symptomatic* of the perturbation.

Example

A rapid decrease in blood pressure may hint at massive loss of blood due to some internal bleeding (haemorrhage), which itself is hard to measure directly. Intuitively, the bleeding would not be regarded in the context of the blood pressure decrease, as would be delivery of diuretic or vasodilating drugs.

◊

Figure 2.8 depicts the relation of context, a context-dependent variable, and a context-sensitive system; respective definitions follow below.

[8] The Monte Carlo method of integration [Avent 87] is the only non-deterministic contribution to monitoring I know of.

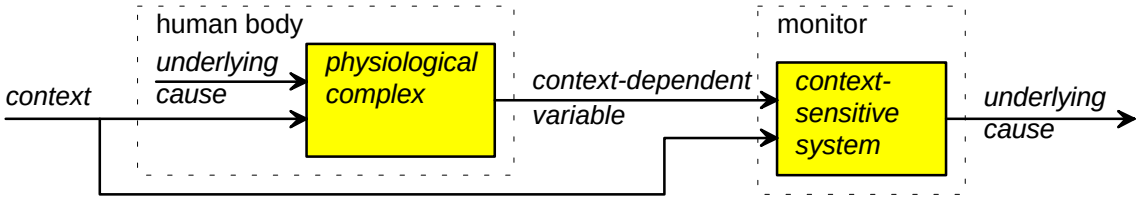


Figure 2.8: Reversal of context dependency by a context-sensitive system

Definition 2.6 (context dependency)
A variable is called *context-dependent* if there is a number of measurable factors (directly observable variables) on which its value depends.
■

Following this definition, context dependency is a property of variables. Factors that are commonly regarded to form a context of monitoring include

- sex,
- age of the patient,
- diseases known to be present in the patient and
- therapy (ventilation mode, ventilator settings, drug delivery, etc.).

A system's compensation for context dependency is called *context sensitivity*. Context-sensitive systems aim at eliminating the effects of context-forming factors on the value of a variable with the purpose of revealing the nature of an underlying, unknown variable. Obviously, a context-sensitive system takes all these factors together with the context-dependent variable as its input.

Definition 2.7 (context sensitivity)
A system is called *context-sensitive* if it regards the context dependency of its input, i.e., if it evaluates some of its input variables taking others as the context of evaluation.
■

Context sensitivity is thus a property of systems.

Example

Adjustment of the fraction of inspired oxygen (F_IO_2) is a therapeutic measure to control the oxygenation of a patient and may thus be regarded as a context of oxygenation. Assuming that arterial partial pressure of oxygen (P_aO_2) in a healthy person is proportional to F_IO_2, context sensitivity of the assessment of basic oxygenation can be achieved by regarding the quotient P_aO_2/F_IO_2 rather than P_aO_2 alone. The effect of the therapeutic effort is so neutralized, and pathological oxygenation, if present, becomes apparent.

◊

Note that the distinction between context-sensitive and ordinary systems is often possible only on a semantic level:

Example

Revisiting the previous example, the same system can be employed to derive the oxygenation index, which is a frequently used indicator for acceptable oxygenation.

◊

It therefore follows that context-sensitive systems do not differ technically from ordinary ones having two or more inputs: of any system with multiple input variables some can always serve as the context of others. This is reflected in the presentation of systems in the next chapter, where there is no one class solely dedicated to the treatment of context dependency.

Practical consideration of context dependency suffers from the fact that even if one is aware of the existence of dependencies, one cannot always describe them accurately, let alone quantify them. The reversal of a context dependency as suggested in Figure 2.8 is therefore not always as straightforward to implement as in the previous example. A common by-pass is then the introduction of symbolic contexts such as states, which serve as the basis of a (usually weak) abstraction of a functional dependency resisting analytical specification.

2.8 Other Monitoring Frameworks and Related Work

Several monitoring frameworks have been proposed, partly in the course of, and partly independent of particular patient monitoring projects. This section will only discuss the work addressing significant aspects of a monitoring framework as defined in Definition 2.1; a general discussion of different approaches to monitoring will follow in Chapter 7.

2.8.1 Elements

The duality of variables and systems is adopted by many authors dealing with intelligent monitoring; model-based reasoning employing causal (e.g., [Doyle 89]) and qualitative models as well as heterogeneous frameworks such as the process trellis [Factor 89, 90] are examples of this.

Uncertainty

In a realistic setting, uncertainty pervades most data and its processing. Note that processing such as inference does not only propagate uncertainty inherent in the premises to the conclusions, but also imposes additional uncertainty on the outcome reflecting the uncertainty inherent in the reasoning process itself.

Ever since MYCIN [Shortliffe 76], the integration of uncertainty in medical expert systems has been a prime matter of concern. Currently, probabilistic methods such as Bayesian networks ([Rutledge 90b, Charniak 91, Berzuini 92, Heckerman 92]) compete with possibilistic and fuzzy approaches.

However, treatment of uncertainty seems not as widespread in diagnostic monitoring as it is in consultation systems: VM as one of the first, although employing the MYCIN formalism, did not make use of the inherited certainty factor model [Fagan 84]. In the sequel, monitoring projects seem to have favoured hypothesis-based approaches to handle uncertain diagnoses (for example, YAQ [Uckun 92a, 93a], or TRENDX [Haimowitz 93a, 94]) over distribution-based ones (e.g., VENTPLAN [Rutledge 90a, 93]).

Apart from its explicit regard of vagueness via fuzzy sets and a special employment in Chapter 6, this monitoring framework does not address issues of uncertainty. This renunciation implies strong assumptions about the quality of all processed data and the modelled functional dependencies:

- all measurements are sufficiently precise (measurement error not greater than the inevitable discretization error),
- all functional dependencies implemented by systems are of deterministic, unambiguous nature, and
- dense samples only are used as the basis of trend detection (the only exception is the stand-alone employment of trend detection of Chapter 6, in which uncertainty in the output is obvious).

Quite obviously, these assumptions are unduly restrictive. In particular,

- all measurements are subject to error (which is in fact still large for non-invasive measurements such as transcutaneous oximetry);

- according to the principle of incompatibility [Zadeh 73] (stating that the mutual exclusion of precision and significance of statements about a system increases with its complexity), systems can be specified unambiguously only at primitive levels where data processing relies on strong physical and mathematical laws; and
- as pointed out before, all sampling is practically subject to occasional equipment failure, interventions and other external factors and therefore invariably sparse (compare Definition 2.4).

Nevertheless, adequate handling of uncertainty has been abandoned, because

- imprecise measurements would require a recognition of uncertainty that can be dealt with by subsequent processing (note again that fuzziness in the form of degree- and fuzzy-set-typed variables does not provide for uncertain values, but for graduation in the form of partial fulfilment of diagnostic criteria or partial applicability of descriptive terms); and
- sparse sampling implies periods of ignorance that inevitably torpedo all attempts to identify trends in a signal with certainty, requiring all high level abstractions dependent on the presence of trends in sparse samples to make clear the fact that they are based on assumptions.

Time-Dependency

[Kahn 91c] provides a recent overview about the modelling of time in different medical decision aid programs. Personally, I find the discussion about whether time-related observations are more adequately represented by time points or intervals somewhat artefactual: all physiological and environmental variables are continuous-time in nature. Consequently, pointwisely recorded observations are partial descriptions of the continuous-time reality, and interval-based recordings are piecewise abstractions of continuous-time reality. From this point of view, notions of facts and events or equivalents prominent in many artificial intelligence approaches to modelling time (see, for example, [Shoham 87] including a critical treatise of [McDermott 82] and [Allen 84]) are rather abstract—even variables commonly associated with events are naturally observed as continuous-time.

Example

If a stroke is defined as the sudden obstruction of a cerebral blood vessel, then the event "stroke at t_0" would be observed as the step function

$$vesselObstructed(t) = \begin{cases} 0 & \text{for } t < t_0 \\ 1 & \text{else} \end{cases}$$

or, indirectly, as some respective neurological deficiency. (Theoretically, the event could be associated with a pulse of infinite amplitude, corresponding to the first derivative of *vesselObstructed* at t_0.)

◊

Nevertheless, the derivation of stable temporal abstractions is one declared goal of diagnostic monitors constructed within this framework. The main difference to interval-based abstractions is that the temporal abstractions are conveyed pointwisely, for example in the form of a trend detected in a certain interval, or as a state comprising current and past observations (see Sections 3.7 and 3.9).

Dealing with Context Dependency

VM, using a slightly different terminology, adjusts its expectations to the current state of the patient/ventilator complex [Fagan 84]. These expectations are manifested in classification rules labelling parameters as *ideal*, *acceptable*, *very unacceptable* and *impossible*.

The context sensitivity of VM is limited to symbolic contexts in the form of states. However, it could easily be extended to cover dependencies on arbitrary contexts, either by introducing conditions other than state transitions for a change of expectations, or by directly coding them in its MYCIN-style rules.

Slightly different approaches to context sensitivity are taken by TOPAZ and RÉSUMÉ: motivated by the problem of alternating treatment protocols, a context is introduced to restrict the scope of temporal abstraction to the administration period of one protocol. In both systems contexts are required to be inherently symbolic (discrete), as opposed to originally continuous-valued factors such as drug dosages crudely abstracted to simplify context sensitization.

Integration of Predated Data

Russ's Temporal Control Structure (TCS) [Russ 86, Russ 90] is a framework for handling information that arrives out of chronological order. Beside maintaining conventional notions of systems (called *modules*) and input and output, TCS introduces *history* variables passing information forward in time to affect future decisions and *oracle* variables passing data backward to revise past decisions.

TCS is based on a discrete time line. All input is processed by so-called *processes*, instances of modules individually assigned to the time of the input. Each process maintains a state that is made available to others via history and oracle variables.

On arrival of new data, past interpretation of old data is revised by allowing a look-ahead, i.e., by regarding information that has become available later in time.[9] This allows two variants of non-standard reasoning: firstly, if an output (based on implicit assumptions) is proved wrong by subsequent findings, then this output can be corrected, and secondly, if an input becomes available with a time lag, evaluation of past processes is repeated regarding the delayed information. Doing so, the TCS is able to maintain an up-to-date interpretation of all information available for all (including past) times.

Revision of past reasoning (particularly if triggering a long chain of re-evaluations) is computationally expensive and requires careful bookkeeping to maintain transparency, especially as practical decisions such as therapy delivery based on assumptions subsequently revised cannot be undone. TCS appears to be designed primarily for therapy planning based on infrequent examinations and findings, and employment in high-frequency sampling domains would require massive temporal abstraction prior to TCS reasoning. Nevertheless, the TCS overcomes Restriction 1 of Section 2.1 and is undoubtedly a major contribution towards realistic on-line employment of diagnostic monitors.

It remains to be noted that the RÉSUMÉ system [Shahar 92b] utilizes a truth maintenance system to update its temporal abstractions in the spirit of TCS.

2.8.2 Structure

Certain structures are common to most larger monitoring systems: modularization and hierarchical organization are popular means to handle complexity not only in monitoring projects.

Hierarchical Architectures

Factor's process trellis [Factor 89, Factor 90] is certainly the most direct competitor of my monitoring framework. A monitor designed as a process trellis program consists of a number of arbitrary, heterogeneous systems (called *processes*) hierarchically arranged to model some part of the problem domain. Information flow is primarily from lower levels (the bottom of the hierarchy) to higher levels (the top of the hierarchy), although higher-level systems can make their current state visible to lower level systems to serve as a context. The process trellis

[9] TCS's systems, the processes, are hence not causal.

makes no assumptions about the individual processes as long as they satisfy simple interface specifications. Particular data types are not defined.

Factor's main work focuses on the implementation and execution of process trellis monitors on parallel hardware. Because he allows bidirectional information flow in the otherwise acyclic hierarchy, he has to address the problem of stability (termination of iterative evaluation of all processes directly or indirectly affected by new input), which I avoid by allowing acyclic dependencies and bottom-up information flow only.

The process trellis architecture has been successfully instantiated for several monitoring tasks, including, from the medical domain, DYNASCENE [Cohn 90] and the Intelligent Cardiovascular Monitor (ICM) [Cohn 91]. Presumably, all monitors presented in Chapter 5 could also be implemented as process trellises. However, whereas the process trellis is a project for its own sake, the theoretical, application- and implementation-independent framework presented in this chapter mainly serves to integrate my diverse work in the field.

The multi-trellis, a successor of the process trellis, allows the integration of process trellises operating at different sampling frequencies. A multi-trellis is thus able to integrate very low level signal processing tasks requiring high frequency samples (such as, for example, the derivation of the heart rate from the ECG signal) and high level processing (interpretation) based on low sampling rates [Factor 92]. The framework presented in this chapter does not provide for such integration.

Blackboard Architectures

COMPAS [Sittig 89] is a prototype *com*puterized *p*atient *a*dvice *s*ystem based on a blackboard control architecture combining data from an underlying hospital information system (the HELP system) with independent, self-activating knowledge sources (programs that perform the reasoning) and a therapy scheduler. Although the control structure of COMPAS was originally implicit (invocation of knowledge sources handled by the *data drive mechanism* of the underlying HELP hospital information system [Sittig 90a]), it is reported to have been changed to an explicit chaining of the programs implementing knowledge sources to improve performance [Sittig 89].

GUARDIAN's architecture [Hayes-Roth 89, 92] is driven by practical requirements including real-time constraints and therefore very pragmatic. It is basically that of a big system organized in layers centred around a blackboard. GUARDIAN's architecture is more than a conceptual environment and thus more than a general framework of monitoring—it remains to be seen if it fulfils its authors' ambitious expectations with respect to generality and scalability. Like all systems with explicit control flow GUARDIAN makes it hard to envisage integration of independently developed subsystems (compare Chapter 4 on the implementation of this framework).

Early work by Dawant and Jansen also emphasizes the blackboard (Dawant 91). In their approach, the blackboard serves as the central medium of information interchange between (the knowledge bases of) two collaborating experts, the domain and the signal processing expert. The domain expert places its requests and the signal processing expert reports its results on the blackboard. Special to the approach is the fact that it allows what the authors call an opportunistic approach to signal interpretation, which basically means that the signal processing expert autonomously decides which feature extraction method to use, dependent only on signal characteristics and specific parameters received (via the blackboard) along with the domain expert's request, an idea that was already put forward in the HASP/SIAP project [Nii 82]. This clear separation of domain and signal processing expertise may be expected to improve transferability of knowledge bases to different application domains.

Modular Architectures

Following common software engineering practice, most other monitoring systems are modular. SIMON for example consists of three main modules: a patient model, a data abstraction module, and a display module [Dawant 94]. The data abstraction module employs measurement objects (each dedicated to a sensor or other input device), quantity objects (the variables of the patient model), and fault and artefact objects[10]. Quantity objects, the output of the data abstraction module, are derived from measurement objects via a network of feature extraction and data validation methods dynamically configured to suit the current monitoring situation.

Compared with my approach, SIMON certainly employs the more specialized monitoring framework: it encourages data validation by explicitly differentiating between measurement and quantity objects. Nevertheless, it is a framework, as all signal processing methods are taken from a standardized library, so that monitors can be configured individually. The basic difference is that SIMON separates the patient model and display from the signal processing.

General Structure Proposals

Coiera identifies several conceptual layers of interpretation: signal acquisition, validation (noise elimination and artefact rejection), pattern recognition (signal-to-symbol conversion), inference, and task layer [Coiera 93].[11] Transferred to system design, conceptual layers induce a layered architecture with strongly specified interfaces. The above layers, for example, imply that only symbols are passed between the pattern recognition and the inference layer. While such a strong separation accords with system modularization being advantageous in large software projects, it is also rather rigid and induces strong bidirectional data flow. For example, if signals are to be interpreted in a context derived in the inference layer, this (symbolic) context is to be passed down to the pattern recognition layer.

Also, the representation of information need not always be correlated with its level of processing: classification of the heart rate as normal, for example, is on a more primitive level than a numerical score (severity index of a disease) derived by complex evaluation regarding trends and other patterns. Nevertheless, each component of a monitor can surely be assigned to one of the suggested layers, so that respective layers can serve as a classification scheme of monitoring components, independent of the system architecture.

[10] The term *object* is used in an object-oriented sense; each object has not only data, but also (individual) functionality associated with it.

[11] Similar layers can be found in [Sittig 89, Factor 90, Miksch 93] etc.

Chapter 3

Elementary Systems of Monitoring

The previous chapter defined a framework for the design of diagnostic monitors. This chapter provides the basic building blocks, the elementary systems of which a monitor is composed.

The following sections address one class of elementary systems each. Some of the systems are standard technology, others reflect current research of other authors, and others represent my own work in the field. Trend detectors based on fuzzy courses (Section 3.7) and fuzzy automata (Section 3.9) are treated in length: they represent my own contribution to diagnostic monitoring. It should be clear that none of the described systems alone makes a diagnostic monitor; yet, they all are indispensable in constructing practical monitors.

Delays implement a backward time shift of signals. They can be used as primitive means of synchronizing variables with the output of systems implicitly delayed (compare Section 2.5.2).

Mutes suppress the values of a variable over periods specified by some triggering condition. Mutes eliminate samples found to be void by data validation; they also serve to take account of interventions, situations known to affect the normal behaviour of an observed process.

Data interpolation is somewhat complementary to muting: it derives the values of a variable for times at which they are unknown. Its main purpose is to synchronize signals of different origin, i.e., to bring together variables sampled at different points in time.

Data smoothing techniques form the basis of every attempt to achieve stable temporal abstractions. They aim at stripping signals from noise and other unwanted short-term fluctuations. Some data smoothing techniques are even capable of completely removing artefacts in the form of outliers.

Differences, *divided differences* and *derivatives* allow an assessment of the development of a variable by relating the current with its past (and future) values. Divided differences and derivatives are sometimes referred to as trends. Throughout this work, however, the term *trend* is reserved for a more comprehensive concept.

Discriminators and *trend detectors* perform number- and signal-to-symbol conversion, respectively. Discriminators cover the mapping of instantaneous observations to sets or classes of values while trend detectors take the development over time into account. The trend detectors differ from other approaches to signal-to-symbol conversion in that their output is also a signal, but that this signal's value is symbolic rather than numeric.

Discrete-valued functions are representative of a number of higher-level, yet flat (i.e. irrespective of the temporal dimension and hence memoryless) functions commonly employed to manipulate or operate on symbols. Discrete-valued functions are presented here as a perspective allowing integration of tables and rules into the monitoring framework.

Finally, stable temporal abstractions in the form of *states* can be achieved by finite automata implementing notions of state and state transition. A state represents a distinct condition of a patient that has evolved in the natural course of a disease, and *state transitions* model possible changes of state along with their characteristic conditions. Work on fuzzy automata extends the concept by formalizing a notion of smooth transition from one state to the next, introducing continuous positions in the state space.

The systems are presented in the framework set by Chapter 2. This allows uniform specification and guarantees straightforward coupling. Composition of monitors itself is not the subject of this chapter. For details and examples of construction refer to Chapters 4, 5 and 6.

Figures employed to visualize the functionality of systems follow a few basic conventions: continuous-time signals are depicted using solid lines, while discrete-time sequences of samples are denoted by dots connected with the time scale through thin, vertical lines. Input and output of a system, if presented in one diagram, are coloured grey and black, respectively. Whenever the output of a system would conceal its input, the input is slightly offset in time to make it visible.

3.1 Delay

In a continuous-time environment, a delay is a system defined by

$$y(t) = x(t-d)$$

where d is its *delay*. Figure 3.1 a) depicts input and output of such a system. For its discrete-time counterpart

$$y[t_n] = x[t_n - d]$$

the output is only defined if x is sampled at $t_n - d$. This is certainly the case if $\langle t_n \rangle$ is a periodic sampling sequence (with sampling period T) and $d = nT$ for some positive integer n. In all other cases, however, samples of x will get lost, i.e., not appear as output of the system. In these cases, insertion of an interpolator (Section 3.3) is advisable.

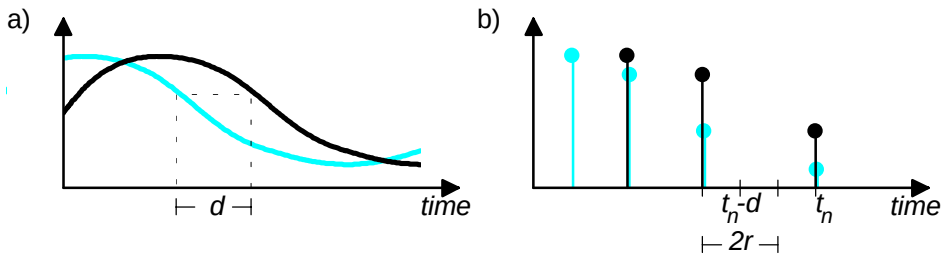


Figure 3.1: Delay d
a) of a continuous-time signal
b) of a discrete-time signal with tolerance r

A practical extension of the definition of a delay is given by

$$y[t_n] = \begin{cases} x[t_m] & \text{if} \quad |t_n - t_m - d| \le r \ \wedge \ \forall t_k : |t_n - t_m - d| \le |t_n - t_k - d| \\ \perp & \text{else} \end{cases} \tag{3.1}$$

where t_m is the sampling time closest to $t_n - d$ not deviating by more than *tolerance* or *radius* r from $t_n - d$, as shown if Figure 3.1 b). (3.1) implicitly assumes a variant of zero-order interpolation as implemented by latches defined in Section 3.3.1.

A delay is usually a side effect of window-based systems that would otherwise require a look-ahead, such as the data smoothing systems of Section 3.4. In its own right, a delay can be used to synchronize a signal with another that has side-effectively been delayed.

3.2 Muting

Mutes are primitive systems designed to suppress signals whenever their values are known to be invalid. A mute takes two inputs, an arbitrary type variable x to be muted and a degree type variable x_c conveying the triggering condition. A mute can be tuned by setting a threshold α to a value in]0,1], the exceeding of which causes x to be suppressed:

$$y[t_n] = \begin{cases} \perp & \text{if} \quad x_c[t_n] \ge \alpha \\ x[t_n] & \text{else} \end{cases}$$

Mutes replace samples by unknown values which may be useful in data validation:

Example

Of two sources of the heart rate, pulse oximetry and electrocardiogram (ECG), the latter is usually considered the more reliable. If the oximeter's heart rate persistently deviates from the ECG's, there is good reason to believe that the sensor has become displaced so that other measured variables such as transcutaneous arterial oxygen saturation are erroneous, too. These variables are then muted so that dependent systems perceive their values as unknown.

◊

Mutes also prove useful in adequate handling of interventions that are known to affect physiologic variables in an unpredictable way:

Example

Interventions such as endotracheal suctioning can significantly influence blood gases and other variables such as the heart rate. For the detection of longer-term trends (exceeding in duration the period of the intervention) these variables can be "turned off" by a mute triggered on the intervention. Trend detection and other dependent systems will then make do with undefined values for the period of intervention.

◊

3.3 Data Interpolation

As mentioned above, a sequence of samples defines a partial function that leaves a large portion of the course of a variable unspecified. The general purpose of interpolation is to reconstruct the original signal $x(t)$ from the given samples as closely as possible, i.e., to derive some reconstructed signal $x_r(t)$ from $\langle x[t_n] \rangle$ such that

$$d(x(t), x_r(t))$$

is minimzed for some metric $d(\cdot, \cdot)$.

Interpolating systems can thus be employed to replace unknown values in sequences of samples with reconstructed values:

$$y[t_n] = \begin{cases} x[t_n] & \text{if} \quad x[t_n] \neq \perp \\ x_r(t_n) & \text{else} \end{cases} \tag{3.2}$$

A variety of interpolation methods can be found in the literature of, for example, numerical mathematics (e.g. [Fröberg 85]). A collection that I found useful in this context is presented below.

The adequacy of a particular interpolation method and its accuracy depends on qualities of the signal and the sampling rate; selection of the interpolation best suited therefore requires some basic understanding of the underlying process. In any case, however, the benefit of interpolation is limited by the following dilemma: if sampling is dense, interpolation is accurate, yet merely needed to synchronize signals sampled at different rates or phase-shifts; if sampling is sparse, reliable estimates of intermediate values are urgently needed, yet cannot be guaranteed by interpolation (compare Definition 2.4 of sparse and dense sampling).

3.3.1 Latch

A latch is a system that holds its most recent input value[12], either infinitely or for a limited time. In either case, a new input always overrides its predecessor. The output of a latch is thus a (partial) step function.

A latch with infinite hold as depicted in Figure 3.2 a) is defined by (3.2) and

$$x_r(t) = x[t_n] \quad \text{where } t_n = \max\langle t_m | t_m \leq t \rangle|.$$

Example

Changes in the operation mode of a technical device will usually only be reported once, and it is understood that the device remains in that mode until a new one is selected. Because adequate interpretation of certain variables may depend on the mode, it can be made available continuously by expanding it with a latch with infinite hold.

◊

A latch with hold time *h* is defined as

$$x_r(t) = \begin{cases} x[t_n] & \text{if } t_n = \max\langle t_m | t_m \leq t \rangle \geq t - h| \\ \perp & \text{else} \end{cases}$$

which is exemplified in Figure 3.2 b).

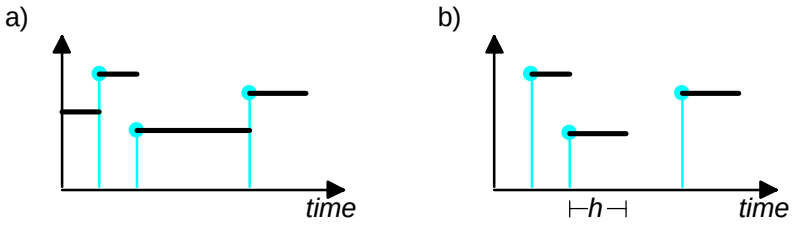


Figure 3.2: Latches a) with infinite and b) with finite hold *h*

Example

Bone age is a variable that changes only fairly slowly. A respective measurement may thus be considered to remain valid for a certain period, a fact that can be modelled by a latch with that period as hold time. More sophisticated treatise of persistence is addressed in Section 3.3.5.

◊

[12] which can be of arbitrary type

The interpolating function of a latch is piecewisely constant; it is therefore also called zero-order interpolation. Consequently, the reconstructed signal $x_r(t)$ is generally discontinuous at t_n, which suggests that it is not a good approximation of physiological signals.

3.3.2 Linear Interpolation

Linear interpolation reconstructs a continuous-time signal by connecting consecutive samples through straight lines. The so reconstructed signal is piecewisely defined as

$$x_r(t) = x[t_n] + (t - t_n)\frac{x[t_{n+1}] - x[t_n]}{t_{n+1} - t_n} \quad \text{for } t_n \le t \le t_{n+1}\,.$$

Note that $x_r(t)$ is not continuously differentiable at sampling times t_n, a property that would be expected of biomedical signals. This circumstance hints at the fact that linear interpolation is hardly ever accurate. In fact, accuracy of linear interpolation depends on the sampling rate: if it is significantly greater than the Nyquist rate, the signal will not vary significantly between samples so that linear interpolation will be acceptably accurate [Oppenheim 89].

Unlike the zero-order hold, linear interpolation requires knowing the sample immediately following the time for which a value is interpolated, a fact that makes it noncausal. In an on-line monitoring environment this means that the system has to wait for the next sample t_{n+1} of the input to be available before it can derive the interpolated values for $t \in\,]t_n, t_{n+1}[$, i.e., the output is delayed for an indeterminate time. Linear interpolation is therefore rarely used in prospective, on-line monitoring. In retrospective analysis of recorded data, however, it is a very popular form of signal reconstruction: in fact, it is argued that any nonlinear development can be approximated by the first term of its Taylor Series expansion [Avent 90], which is in fact linear. In experiments, it has also proven to be the best single technique for missing data inference [Albridge 88].

3.3.3 Linear Extrapolation

Linear extrapolation overcomes noncausaltity at the cost of a likely greater error rate. It obtains the reconstructed signal as

$$x_r(t) = x[t_n] + (t - t_n)\frac{x[t_n] - x[t_{n-1}]}{t_n - t_{n-1}} \quad \text{where } t_n = \max\langle t_m \,|\, t_m \le t\rangle|.$$

The strategy of linear extrapolation is similar to that of a latch in that it projects past observations into the future. The difference is that it holds the mean slope derived from the last two samples (rather than the absolute value of the last) to reconstruct missing values. The choice between the two depends on whether the physiological model suggests that the slope or the absolute value remains constant. One should bear in mind though that extrapolation may lead to unphysiological values, particularly if the gap to be filled is long.

3.3.4 Nyquist Interpolation

Perfect reconstruction of a signal sampled periodically at a Nyquist rate $1/T$ is given by

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[t_n]\frac{\sin(\pi(t - t_n)/T)}{\pi(t - t_n)/T} \tag{3.3}$$

[Oppenheim 89]. Clearly, this is not the definition of a causal system—in fact, it is not even window-based. Window-based approximations, however, are feasible because (3.3) converges and

$$\lim_{|t-t_n|\to\infty} \frac{\sin(\pi(t - t_n)/T)}{\pi(t - t_n)/T} = 0\,.$$

3.3.5 Theories of Persistence

In the RÉSUMÉ system [Shahar 92a], maximum-gap functions specify the acceptable period of ignorance between two consecutive temporal abstractions that can be bridged to obtain one contiguous abstraction. Temporal abstractions are coded as symbols considered to comprise the state of a variable over a certain interval, and the bridging abstraction is derived from a so-called abstraction-inference table [Shahar 92a].

Example

Two consecutive temporal abstractions for blood pressure *rising* and *falling* can be joined into one named *nonmonotonic*, taken that the separating gap does not exceed the maximum allowable gap for this join.

◊

The maximum-gap function depends on the variable, namely on general knowledge about the inertia of its behaviour, and on the duration for which the involved abstractions persist: it assumes some kind of persistence of observations. Other approaches dealing with the persistence of temporal propositions include VM ([Fagan 84] at least giving it a mention), [Dean 88] providing a probabilistic framework, and [Rutledge 90] defining a half-life for the time-decay of the belief in a measurement represented in VENTPLAN's Bayesian network.

Persistence-based interpolation methods are particularly suited for interpolation of symbolic variables where no continuous transition from one value to another is provided for. Note that fuzzy set type variables overcome this lack of graduation in symbolic domains, as the following example suggests.

Example

Interpolation between two consecutive samples

$$MAP[t_1] = \{(normal, 1), (falling, 0)\}, MAP[t_2] = \{(normal, 0), (falling, 1)\}$$

of the fuzzy set type variable *MAP* declared as

$$MAP : T_c \rightarrow \tilde{\wp}(\{normal, falling\})$$

can be given by

$$MAP_r(t) = \left\{ \left(normal, \frac{t_2 - t}{t_2 - t_1} \right) + \left(falling, \frac{t - t_1}{t_2 - t_1} \right) \right\} .$$

◊

3.4 Data Smoothing

Although close surveillance of a patient as achieved by high frequency sampling is considered a major advance of state-of-the-art clinical monitoring devices, it also reveals properties of physiological variables and measurement instrumentation that where formerly easy to ignore: all technical instrumentation is affected by measurement noise and physiological variables are subject to natural fluctuations superposing patterns of clinical significance.

Example

Instantaneous arterial blood pressure depends primarily on the action phase of the heart. Its maximum is called *systolic* (SAP) and its minimum is called *diastolic arterial blood pressure* (DAP). More significant for tissue perfusion, however, is the *mean arterial pressure* (MAP), which is defined as the arithmetic mean of the instantaneous pressure and obtained via integration over one stroke cycle, as depicted in Figure 3.3 a).

Among other dependencies, MAP decreases with decreasing venous return. Reduced venous return in turn may also have multiple reasons: massive loss of blood, increased

resistance of pulmonary arteries, and dilatation of pulmonary veins are three examples. However, while the former two are indicative of severe complications and deserve notification, the latter is a natural effect occurring during the inhale phase of the normal breathing cycle, as a result of decreasing intrathoracal and consequently increasing transmural pressure of the lung veins. The course of MAP is therefore rhythmically influenced by a bounded fluctuation dependent on the respiratory phase as depicted in Figure 3.3 b), which is natural and of no diagnostic interest. [Schmidt 87]

If no means of eliminating this second-order fluctuation from the measurements are employed, any clinically significant trend will be superposed by the effects of the patient's respiratory effort, as shown in Figure 3.3 c).

◊

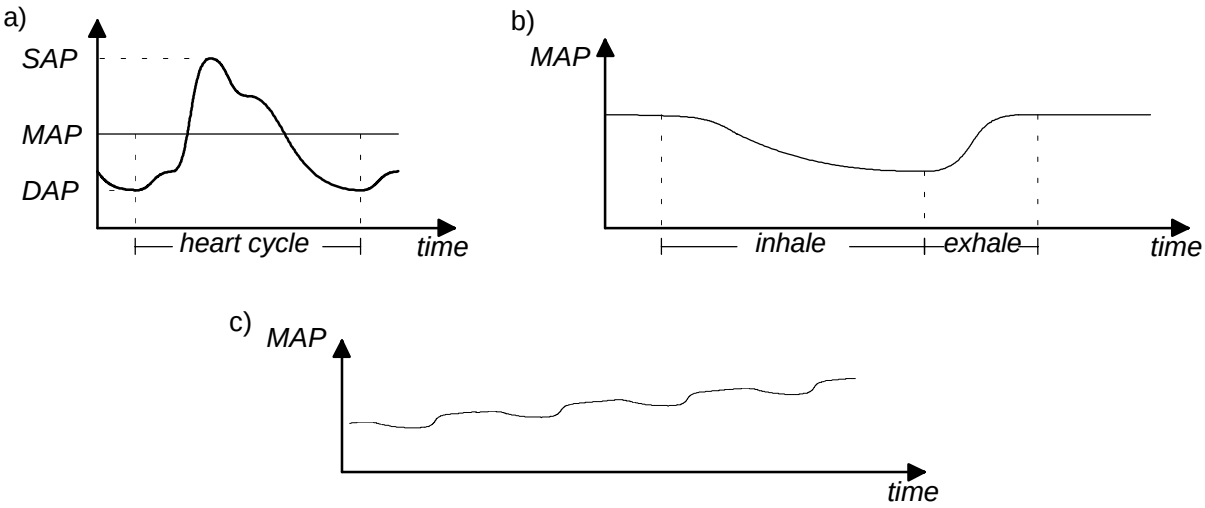


Figure 3.3: Blood pressure
a) first-order dynamics of blood pressure
b) effects of respiration on mean arterial pressure
c) rising blood pressure superimposed by the second-order fluctuations of b)

Data smoothing aims at the elimination of short-term fluctuations that should not influence interpretation of a signal. Technically it is often related to low pass filtering, although there are smoothing methods that work in environments where noise and signal share the same high frequency band (e.g. the *median filter*, see below). In case of the previous example, a perturbation manifesting itself in a change of MAP with a time constant significantly longer than a respiratory cycle can be isolated by data smoothing.

Data smoothing is typically applied to dense samples. If sampling is sparse (for example, if only few samples of a fast changing variable are available over a long period of time), smoothing unduly assimilates these samples.

Data smoothing mostly relies on mathematical and statistical methods: its input and output are therefore numeric or degree-typed. Generally, however, derived symbolic variables may also be subject to noise and other fluctuations, so that an analogue need for smoothing arises. This is served in part by median filters (see below) and by temporal abstraction methods as discussed in Sections 3.6.4 and 3.9.3.

3.4.1 Moving Averages

The moving average is employed in clinical monitoring primarily to eliminate randomness in series of data. Its main tuning parameter is the time span (window width) *w* over which the average is derived, which is equivalent to the number of samples in case of periodic sampling.

The moving average of a continuous-time signal is defined as

$$y(t) = \frac{1}{w} \int_{t-\frac{w}{2}}^{t+\frac{w}{2}} x(u)du, \tag{3.4}$$

which makes it a window-based noncausal system. Recall that in a prospective environment such as on-line monitoring the window can be shifted back in time resulting in a causal definition of moving average with an implicit delay of $w/2$ (compare Section 2.5.2). Its discrete-time counterpart

$$y[t_n] = \frac{1}{k} \sum_{t_m : t_n - w \le t_m \le t_n} x[t_m] \quad \text{with } k = |\{t_m | t_n - w \le t_m \le t_n\}| \tag{3.5}$$

relies on implicit signal reconstruction (see below). Note that (3.5) produces results biased to denser-sampled periods of the variable if sampling is not periodic.

The idea behind the moving average is that short-term deviation from the basic signal is distributed equally around zero, so that summation will eventually eliminate its influence. However, the quality of noise elimination increases only with the window width, which in turn increases the output's inertia to react to changes in the underlying signal.

Example

Abrupt increase in the fraction of inspired oxygen (F_IO_2) in a poorly oxygenated patient should result in a step change in the patient's arterial oxygen saturation S_aO_2. However, a moving average transforms a step change of an otherwise steady signal to a ramp over the full window width as shown in Figure 3.4 a), blurring the immediate effect.

◊

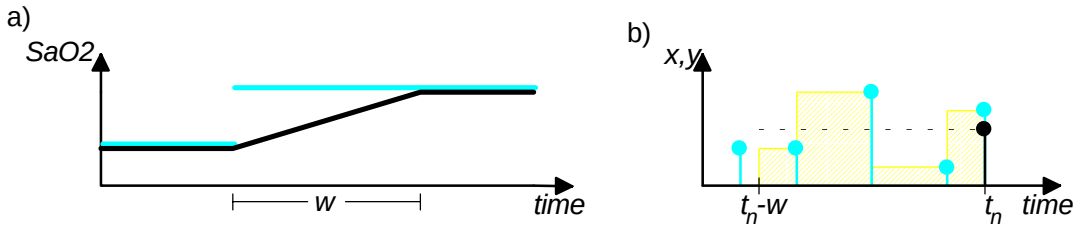


Figure 3.4: Moving average
a) reaction to a step change
b) the weighted moving average of (3.6)

Ideally, to obtain unbiased results in case of aperiodic sampling the original signal is first perfectly reconstructed by an interpolation method and then (3.4) is applied to the reconstruction. Alternatively, samples can be weighted to account for the time span they represent, as done in

$$y[t_n] = \sum_{t_m : t_n - w \le t_m \le t_n} \frac{t_m - \max(t_{m-1}, t_n - w)}{w} \cdot x[t_m], \tag{3.6}$$

which corresponds to cubic integration as shown in Figure 3.4 b). Again, this is the causal, implicitly delaying version obtained by a translation of the window by $w/2$.

Weighted Moving Averages

More generally, a weighted moving average over k periodic samples is commonly defined as

$$y[t_n] = \sum_{m=1}^{k} a_m x[t_{n-k+m}]$$

where $\langle a_k \rangle$ is a sequence of weights such that $0 < a_1 < \ldots < a_k < 1$ and $\Sigma a_i = 1$ [Allen 83]. The general idea behind assigning higher weights to more recent values is to make the average

more sensitive to current developments. In particular, an exponentially weighted moving average (known as EWMA) of infinite length is defined by

$$y[t_n] = \alpha x[t_n] + \alpha(1-\alpha)x[t_{n-1}] + \alpha(1-\alpha)^2 x[t_{n-2}] + \ldots = \alpha \sum_{m=0}^{\infty} (1-\alpha)^m x[t_{n-m}] \quad (3.7)$$

which can be recursively defined as

$$y[t_n] = \alpha x[t_n] + (1-\alpha) y[t_{n-1}]$$

and thus easily computed. Note that the EWMA is usually applied to periodic samples.

Moving Average and Trend Detection

Causal definitions of moving averages, in particular the EWMA and its descendants, can be used as one-step-ahead predictors by interpreting their current output as a forecast of the next input [Allen 83, Avent 90]. If the deviation of the actual from its predicted value is interpreted as indicative of a trend, an EWMA can be used as a trend detector (compare discussion of Kalman filtering in Section 3.7.9). (3.7) then serves as a model of the signal based on past input only; however, this model is only mathematically motivated and hence only loosely related to model-based monitoring methods discussed in Section 7.2.1.

3.4.2 Median Filter

Despite their simplicity and outstanding properties, median filters have only recently found attention in biomedical signal processing. A median filter is a window-based system defined as

$$y[t_n] = \text{median}(\langle x[t_m] | t_{n-k} \le t_m \le t_{n+k} \rangle), \quad (3.8)$$

where median(·) selects the $k+1^{th}$ element of the sorted permutation of a sequence with $2k+1$ elements. Note that the input need not be numeric—a median filter works for totally ordered sets of symbols, too. This property makes it particularly well-suited to smooth the output of monotonic discrimination—see below.

Because the median filter requires an odd number of samples, its window in (3.8) is defined in terms of numbers of samples rather than temporal extent (which is equivalent for periodic sampling). However, in environments where aperiodic sampling must be assumed, its definition must be adapted to allow for more liberal sampling sequences. One possibility is to reconstruct the original signal by means of an adequate interpolation method, then sample this reconstruction periodically, and apply (3.8). Another, more direct one, allows arbitrary numbers of samples and alternatively takes the k^{th} element (or some interpolated value from the k^{th} and $k+1^{th}$) as the median. A causal version of such a system is defined by

$$y[t_n] = \text{median}(\langle x[t_m] | t_n - w \le t_m \le t_n \rangle)$$

again with an implicit delay of $w/2$.

Just like the discrete-time moving average, the median filter imposes a bias if sampling is aperiodic, which can be counteracted by weighting samples [Hämäläinen 94]. The following highlights the most characteristic properties of median filters; more comprehensive discussions can be found in the literature of digital image and speech processing and in [Gallagher 81].

It follows from the definition of the median filter that monotonic signals pass unchanged, yet with a delay of k samples. Moreover, a signal passes unchanged if any two of its successive monotonic subsequences (of opposite direction) are separated by a sequence of at least k constant samples [Gallagher 81]. Informally speaking, a median filter of length $2k+1$ favours courses where the turnaround takes at least k samples. Because window width is its only design parameter, this observation may be used as a design guideline.

The most characteristic property of median filters is, however, that they eliminate pulses shorter than half their window width while preserving edges. Median filters can therefore

totally remove artefacts in the form of outliers (without their output being affected, as is the case for moving averages), which makes them a tool predestined for data validation.

3.4.3 Bounded Trail

The following recursive definition makes a simple smoothing method which I, not having found any mention in the literature, call *bounded trail*:

$$y[t_n] = \left\{ \begin{array}{lll} x[t_0] & & \text{initially} \\ \left.\begin{array}{lll} x[t_n]-r & \text{if} & x[t_n] > y[t_{n-1}]+r \\ y[t_{n-1}] & \text{if} & y[t_{n-1}]-r \le x[t_n] \le y[t_{n-1}]+r \\ x[t_n]+r & \text{if} & x[t_n] < y[t_{n-1}]-r \end{array}\right\} & & \text{thereafter} \end{array} \right.$$

Its behaviour is depicted in Figure 3.5 a): monotonic changes in one direction cause the output to follow these changes with a distance (radius) of r, and subsequent changes in the opposite direction are ignored (i.e. leave the output unchanged) until they deviate by more than $2r$ from the previous peak. This filter's main disadvantage is that its output lags behind and may never catch up with the input, even if the latter remains constant as shown in Figure 3.5 b). Also, due to its fixed radius the system can only eliminate noise which is limited in amplitude by r.

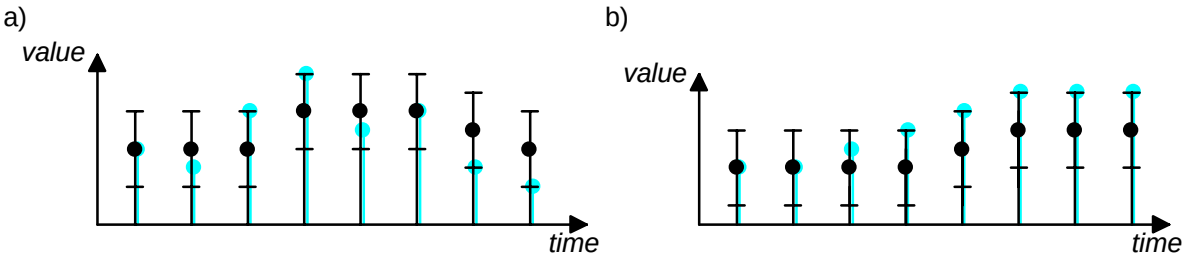


Figure 3.5: Bounded trail

a) $y[t_n]$ is the centre of an interval $[y[t_n]-r, y[t_n]+r]$. As long as the input $x[t_n]$ resides within the interval, the output remains unchanged. Upon departure, the output is adjusted so that the input just fits into the new interval.

b) failure of the output to converge with asymptotic input

3.4.4 Smoothing and Abstraction

Smoothing is intuitively associated with transforming a fluctuating signal to one with longer monotonic intervals by removing noise and other periodical fluctuations of negligible origin. Data smoothing may thus be regarded as a first step towards stable temporal abstractions.

On the other hand, the purpose of abstraction is to distract attention from unnecessary detail and to emphasize significant aspects of a signal. If abstraction is carried out over a temporal interval ("temporal abstraction"), then it is expected to "abstract away" insignificant fluctuations and noise, which parallels the effect of data smoothing.

Example

If the temporal abstraction "heart rate normal for three hours" corresponds to the output of a system producing

$$heartRate[t_n] = normal$$

for three consecutive hours, then this output is certainly a smoothed derivative of some input.

◊

Methods that produce temporal abstractions may thus be considered data smoothing. See Sections 3.6.4 and 3.9.3 for examples of this.

3.5 Differences, Divided Differences and Derivatives

The absolute value of a variable is the product of many factors, some of which persist for a whole monitoring session. As pointed out before, the effects of these factors, even if known to exist, cannot always be reversed. A perturbation superposing those factors is then reflected in a characteristic change of the variable rather than adoption of a typical value. To detect and classify such perturbations, changes are quantified by a measure of distance, which is usually based on the difference of numeric values.

3.5.1 Net Change

The net change of a numeric (or degree type) variable denotes its absolute (as opposed to relative) alteration over a fixed period of time d as determined by

$$y(t) = x(t) - x(t-d)$$

in the continuous-time and

$$y[t_n] = x[t_n] - x[t_m] \quad \text{for } t_m = t_n - d$$

in the discrete-time domain. Note that if sampling is aperiodic, x is not likely to be sampled at all $t_n - d$, so that y remains undefined for some t_n. As a bypass, the net change can be specified in terms of x and x delayed by d such that

$$y[t_n] = x[t_n] - x_d[t_n],$$

where $x_d[t_n]$ is defined as in (3.1). Note that the net change is affected by short-term fluctuations and noise; however, the longer the period (and consequently the greater the net change) becomes, the smaller is the relative error.

The divided differences

$$\frac{x(t) - x(t-w)}{w}$$

and

$$\frac{x[t_n] - x[t_m]}{t_n - t_m},$$

where t_m is defined as above, introduce further time normalization and are dealt with in Section 3.5.3.

3.5.2 First Derivative and its Approximations

To reason about the dynamics of a variable, its relative rather than its net change is of interest. The first derivative of a variable provides this information for any instant in time. However, the definition of the first derivative of $x(t)$ at t_0

$$x'(t_0) = \frac{dx(t_0)}{dt} = \lim_{h \to 0} \frac{x(t_0 + h) - x(t_0)}{h}$$

requires $x(t)$ to be continuous in proximity of t_0, which is obviously not the case for discrete-time observations.

If, however, the original signal can be reconstructed from its samples by interpolation, then the derivative can be determined from the interpolation formula. Such is the case with variables sampled at a Nyquist rate, where

$$x_r(t) = \sum_{m=-\infty}^{\infty} x[t_m] \frac{\sin(\pi(t - t_m)/T)}{\pi(t - t_m)/T}$$

so that

$$x'_r[t_n] \approx \frac{x[t_{n+1}] - x[t_{n-1}]}{T}$$

for periodic sampling with sampling period T is an approximation [Lee 88].

Accordingly, if linear interpolation is acceptable, the first derivative can be obtained from the linear interpolation formula. However, as linear interpolation approximates the original signal as piecewise linear, the interpolation is not generally differentiable at t_n, i.e.,

$$\lim_{h\to 0} \frac{x_r(t_n - h) - x_r(t_n)}{h} \neq \lim_{h\to 0} \frac{x_r(t_n + h) - x_r(t_n)}{h} \quad \text{for most } t_n .$$

In that case, the arithmetic mean of the left and right-sided derivative

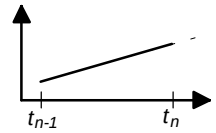
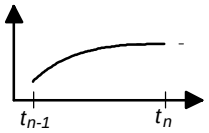
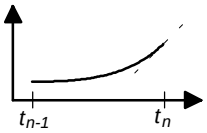
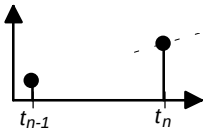
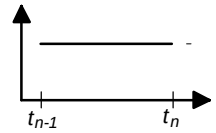
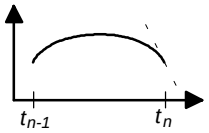
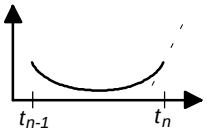
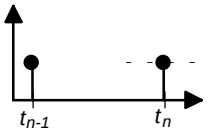
$$x'[t_n] \approx \frac{1}{2}\left(\frac{x[t_n] - x[t_{n-1}]}{t_n - t_{n-1}} + \frac{x[t_{n+1}] - x[t_n]}{t_{n+1} - t_n} \right)$$

may be a good, yet causal, approximation. If a delay cannot be afforded, the derivative of the extrapolation

$$x'[t_n] \approx x'_r[t_n] = \frac{x[t_n] - x[t_{n-1}]}{t_n - t_{n-1}} \tag{3.9}$$

may be a better choice. However, particularly if sampling is sparse, (3.9) is error-prone, as the following table demonstrates.

Table 3.1: Qualitative comparison of actual and extrapolated derivatives of sparsely sampled signals

ORIGINAL COURSES[1]			SAMPLES[2]
			
			
compliance		*non-compliance*	*with extrapolated derivative*

[1] dashed lines indicate derivative at t_n

[2] dashed lines indicate extrapolated derivative

It may be argued that the instantaneous derivative is of little value as it is inherently subject to noise and other insignificant fluctuations and hence not a reliable indicator of the overall development of a variable. Rather, the derivative averaged over a longer period of time would be preferable.

3.5.3 Mean Slope

For a continuous-time signal $x(t)$ the mean slope $\overline{x'_w}$ over a window of width w is defined as

$$\overline{x'_w}(t) = \frac{1}{w} \int_{t-\frac{w}{2}}^{t+\frac{w}{2}} x'(u)du = \frac{1}{w} x(u) \Big|_{t-\frac{w}{2}}^{t+\frac{w}{2}} = \frac{x(t+\frac{w}{2}) - x(t-\frac{w}{2})}{w},$$

which is totally independent of $x'(t)$(!), so that its causal, implicitly delaying version of the discrete-time domain

$$\overline{x'_w}[t_n] = \frac{x[t_n] - x[t_m]}{w} \quad \text{for } t_m = t_n - w$$

yields the true mean slope, no matter whether the original signal can be reconstructed from the sample or not. Note that the continuous-time definition of mean slope is that of the moving average (3.4) applied to the derivative—however, whereas a good moving average for aperiodic samples is nontrivial to derive, the same for the first derivative is almost free. Problems arise again from aperiodic sampling: in most cases, there will be no $t_m = t_n - w$ so that the following piecewise definition based on a tolerance r should be used instead:

$$y[t_n] = \begin{cases} \frac{x[t_n]-x[t_m]}{t_n-t_m} & \text{if } |t_n - t_m - w| \le r \wedge \forall t_k : |t_n - t_m - w| \le |t_n - t_k - w| \\ \perp & \text{else} \end{cases}$$

which is analogous to (3.1).

3.6 Discriminators

A discriminator is a (memoryless) system that transforms instantaneous numeric input to symbolic output by associating quantitative values with qualitative terms. A discriminator is usually implemented as a simple classification process.

One could expect discrimination to have a smoothing effect, as ideally its output changes less frequently than its input. However, as it turns out, this smoothing effect depends on rather arbitrary factors, elimination of which is a goal of more advanced discrimination.

3.6.1 Monotonic Discrimination

A frequent employment of discrimination is to abstract an observed value by one of a collection of predefined typical ranges usually labelled *normal*, *low*, *high*, etc. The idea behind this is that subsequent processing benefits from a reduction of cases to be considered. The abstraction is specified by identifying sets (equivalence classes) designed so that each of its elements is equivalent in meaning with respect to the given problem.

A discriminator on a variable $x : T_c \to V_x$ is specified on the basis of a number of classes $C_1, \ldots, C_n \subset V_x$ where

$$C_1 \cup \ldots \cup C_n = V_x$$

and

$$C_i \cap C_j = \varnothing \quad \text{for all } 1 \le i < j \le n\,.$$

A discriminating system is then defined as

$$y[t_n] = \begin{cases} \text{"}C_1\text{"} & \text{if } x[t_n] \in C_1 \\ \text{"}C_2\text{"} & \text{if } x[t_n] \in C_2 \\ \vdots & \vdots \\ \text{"}C_n\text{"} & \text{if } x[t_n] \in C_n \end{cases}, \tag{3.10}$$

where "C_i" denotes the label of the class so that (3.10) actually assigns a symbol to the output of the discriminator.[13]

If V_x is totally ordered by the ordering relation < and if the classes C_i are convex (i.e. contiguous), an ordering < is implied on them such that

$$C_i < C_j \quad \Leftrightarrow \quad \exists x_1 \in C_i, x_2 \in C_j : x_1 < x_2 \quad \Leftrightarrow \quad \forall x_1 \in C_i, x_2 \in C_j : x_1 < x_2\,.$$

If V_x is numeric, classification is a monotonic mapping from numbers to an ordered set of symbols, as increasing values result in "increasing" symbols and decreasing numbers in "decreasing" symbols (see Figure 3.6 a)). Note that a classification comprising *low* and *high* as *abnormal* is nonmonotonic, because *abnormal* is not a convex set.

[13] This transformation is necessary because of semantic considerations (compare Section 1.2.2).

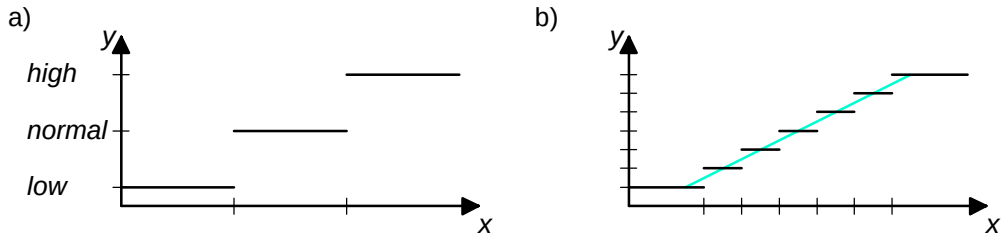


Figure 3.6: Monotonic discrimination

Monotonic discrimination is the most common employed for number-to-symbol conversion in monitoring systems. However, it suffers from considerable drawbacks that many monitoring systems ignore. In particular,

- it is discontinuous at the classes borders,
- it lacks context sensitivity, and
- it does not take patient-specific idiosyncrasies into account.

Idiosyncrasies are a general problem hard to tackle. Although generally recognized, only few authors suggest solutions (e.g. [Doyle 94, van der Aa 90])[14]. The other two points are considered in the following.

3.6.2 Fuzzy Discriminators

Fuzzy discrimination is motivated by the following example (compare discussion in [Fagan 84]):

Example

Consider a simple discriminator that classifies blood pressure between 90 and 110 mm Hg as *normal*, while pressures below 90 and above 110 mm Hg are classified as *low* and *high*, respectively. It is then hard to see why a blood pressure of 89 mm Hg should be regarded *low* and trigger some action, while one of 90 mm Hg should not. The problem we face is that in our understanding there is a considerable difference between normal and low blood pressure, while there is hardly any between the readings 89 mm Hg and 90 mm Hg. On the other hand, blood pressures of 90 and 110 mm Hg are, despite their considerable difference, treated as if they were the same. This is a supposed consequence of our binary understanding of the world, where the border has to be drawn somewhere.

◊

An obvious solution to the problem is to introduce intermediate classes, a procedure that, if applied recursively, approximates a continuous mapping[15] from number to symbol as indicated in Figure 3.6 b).[16] However, on a syntactic level discontinuity is maintained: even if a change of symbols may be interpreted as insignificant on a semantic level, discontinuity may be propagated to the output of the monitor, where it is perceived as an abrupt change of state that invariably attracts attention. Recalling that this is at odds with the goal of deriving stable temporal abstractions (Requirement 4 of Section 2.1), the following considerations are worthwhile.

Continuity in discrimination requires a smooth transition in membership from one class (represented by a symbol) to another. As the value of a variable changes in one direction, this

[14] The tenor in their approaches is to observe the patient for a while, preferably in a healthy state, and "learn" his specific normal ranges.

[15] Informally, a mapping is continuous if small changes in the input result in small changes of the output.

[16] Note that quantization is a special form of monotonic discrimination implemented by analogue/digital converters.

smoothness can be achieved if graduation induces degrading membership in one set and growing in another.

To achieve fuzzy discrimination, the crisp sets C_1 through C_n on which classification is based are replaced by fuzzy sets $\widetilde{C}_1$ through $\widetilde{C}_n$ encoding the meaning of terms with blurred boundaries. The process of continuous discrimination is then implemented by a system

$$y[t_n] = \left\{ (\widetilde{C}_i, \mu_{\widetilde{C}_i}(x[t_n])) \,\middle|\, 1 \le i \le n \right\} \tag{3.11}$$

which derives a level 2 fuzzy set as its output ($y : T_c \rightarrow \wp(\{\widetilde{C}_1, \ldots, \widetilde{C}_n\})$) and which I call a *fuzzy discriminator*.

Figure 3.7 a) presents three fuzzy sets forming the basis of a fuzzy discrimination process. Figure 3.7 b) presents the output of a respective fuzzy system to an assumed input signal; to improve readability, input and output are continuous in time.

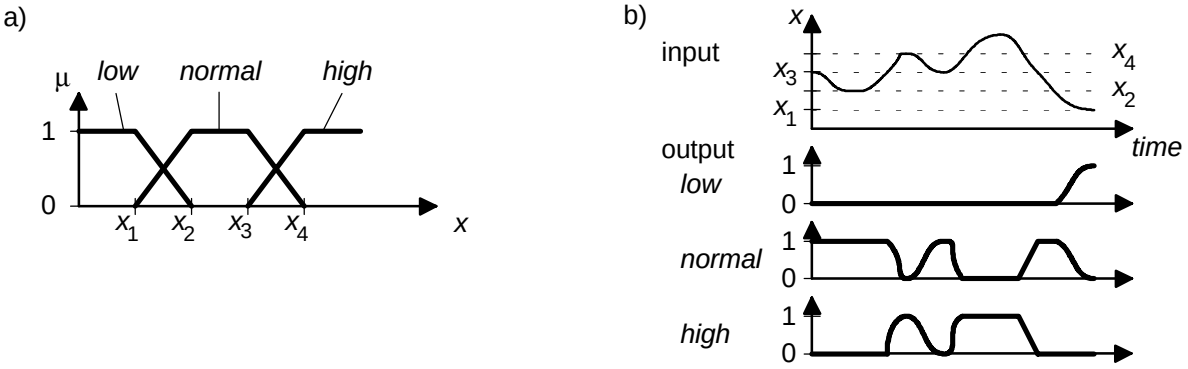


Figure 3.7: Fuzzy discrimination
a) fuzzy sets forming the basis of discrimination
b) output of discrimination in response to a continuous-time input

It is hard to see the advantage of this perspective: rather than one, three continuous signals are to be observed and interpreted, and a certain redundancy in the output cannot be denied. However, fuzzy discrimination combines nicely with fuzzified symbolic systems (see Sections 3.8.3 and 3.9.2). Besides, fuzzy discrimination has proven extremely effective in fuzzy control, where continuity is a prerequisite for achieving stable controller behaviour (see, for example, [Mamdani 93] for a discussion).

If the fuzzy sets are normalized and convex and if

$$\sum_{i=1}^{n} \mu_{\widetilde{C}_i}(x) = 1 \quad \text{for all } x \in V_x,$$

the degrees of membership associated with each value can be interpreted as normalized distance measures between the actual value and the classes. Note that these requirements are satisfied by the fuzzy sets of Figure 3.7 a).

The mapping implemented by the discriminator based on such fuzzy sets can be visualized as shown in Figure 3.8 a), where the classes are points in one-dimensional space (the *y-axis*), and points between classes reflect the position (distance) of an observed value relative to its neighbouring classes. This is possible because the above conditions imply that every observed value either entirely belongs to one class or lies between two adjacent classes $\widetilde{C}_i$ and $\widetilde{C}_j$ such that $\mu_{\widetilde{C}_i}(x) + \mu_{\widetilde{C}_j}(x) = 1$.

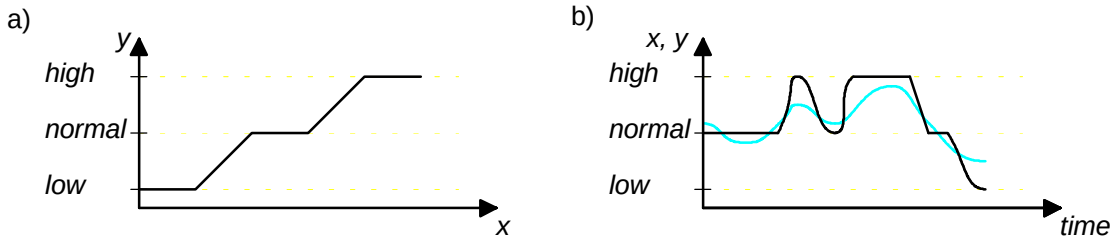


Figure 3.8: Fuzzy discrimination suggesting a "fuzzy position" between classes
a) mapping of *x* to a continuous position in the symbol space
b) transformation of input to an output conveying the "fuzzy position" between symbols (classes)

The result of discriminating the input of Figure 3.7 b) can then be visualized as depicted in Figure 3.8 b). This kind of presentation appears more natural, as it conveys the concept of smooth transition from one abstraction to another.

3.6.3 Context-Sensitive Discriminators

The value of a physiological variable depends on many factors, some of which (usually the ones that are externally controlled or otherwise known) are regarded as *context*, and some of which are the interest of monitoring. Consequently, discrimination such as the assessment of a variable as abnormal should take all known factors into account.

Generally, context-sensitive discrimination can be achieved either

- by reversing the context dependency of the observed signal, or
- by adapting the definition of classes on which discrimination is based by simulating the context dependency on the individual classes' elements.

Because not all context dependencies are reversible (compare Section 2.7), but maybe mostly because it is more intuitive, the second alternative is preferred by most authors (see below for discussion). The classes are then dynamically specified by

$$C_i[t_n] = \{x | \, p_i(x, f_c[t_n])\},$$

where f_c is a context-forming function dependent on the values of other observable variables.

Note that because precise specification of context dependency would require reliable numerical models of the involved physiological complexes, it is frequently approximated by a crude classification in which the context is abstracted to one of a number of discrete states.

3.6.4 Discrimination with Hysteresis

While discontinuity may be tolerated in a static environment, it is particularly awkward for interpretation of time series, as the following example demonstrates.

Example

Revisiting the strict classification scheme of the previous example it is hard to accept that a sequence of samples normally distributed around 90 mm Hg should be abstracted to a continuously alternating sequence of classifications as normal and low. This is particularly true as it makes minor fluctuations around a signal mean of 90 mm Hg indistinguishable from an oscillation between, say, 110 and 60 mm Hg, reflecting serious perturbations or sensor faults.

◊

If the purpose of discrimination is to reduce information content by deriving a stable abstraction of a fluctuating signal, then we must observe that this stability (defined as the reciprocal frequency of change in the symbolic output) heavily depends on the signal mean relative to the classes boundaries. Figure 3.9 a) and b) show the frequency of output of discrimination of a

signal normally distributed around the centre of a class and around the border of two adjacent classes, respectively.

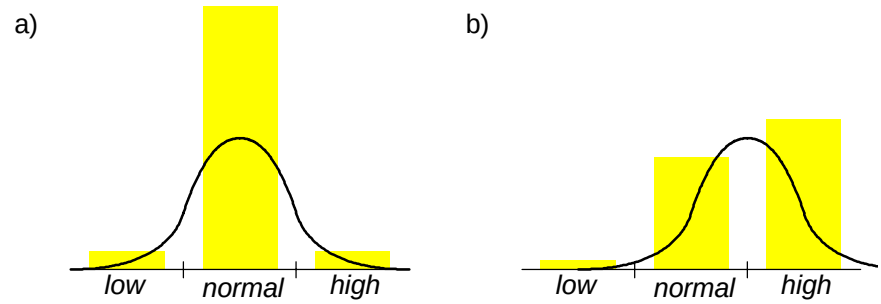


Figure 3.9: Histograms indicating the dependency of discrimination on the choice of the classes' borders
a) signal normally distributed around the centre of *normal* and
b) normally distributed around the border of *normal* and *high*

More awkward than the differing histograms (suggesting a fundamentally different distribution) is the dependence of temporal instability on the choice of the classes borders. In case of Figure 3.9 a), discrimination is fairly stable, while in case of Figure 3.9 b) it is not. Although instability hints on the borderline nature of the classified signal, this output is hardly desirable if stability is required.

An elegant solution to the problem is to make the current classification dependent on the previous: if the previous sample was classified *normal*, then the next should also be considered normal unless it crosses a certain threshold added to (or subtracted from, respectively) the class's original margin, as shown in Figure 3.10. The strict boundaries between classes are so replaced by "elastic constraints".

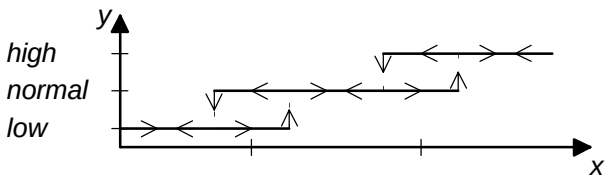


Figure 3.10: Mapping of discrimination with hysteresis

Note how this kind of discrimination incorporates time, as it depends on the sequence in which samples are classified. Indeed, memorizing the previous output requires memory. This can be implemented by the deterministic automaton or finite state machine depicted in Figure 3.11. The transitions are labelled by conditions that reflect the crossing of a threshold *t*, events that trigger transitions from one state to the next. The current state of the automaton is then the output of the discriminator.

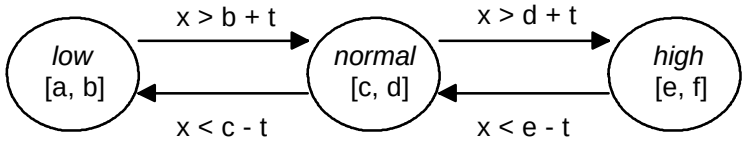


Figure 3.11: Standard three-class discriminator with hysteresis

3.6.5 Related Work

The problem of discrimination is sometimes reduced to that of generating alarms, which corresponds to a binary classification of values as *normal* or *abnormal*. Discrimination as presented

here, however, is much more general: discriminators perform number-to-symbol conversion and can thus be used to access tables or to drive symbolic reasoners or qualitative simulation.

In PREMON, a predictive monitoring system described in [Doyle 89], sensor thresholds are adjusted dynamically by simulating a causal model that predicts the behaviour of its variables. An alarm is raised whenever a discrepancy between the actual and predicted value of a variable is observed. (This procedure is typical for model-based diagnosis; compare discussion in Section 7.2.1).

The same author pursues a totally different approach to anomaly detection in SELMON [Doyle 94]. Motivated by good examples and justified by considerable success, the author argues that anomaly is associated with infrequent patterns rather than crossing of thresholds. He defines a measure of distance for frequency distributions (basically the histograms of Figure 3.9) of nominal and observed behaviour. The author admits that the influence of the partitioning (the choice of number of ranges and their boundaries) on the discriminatory power still needs investigation, yet observes that current partitioning with three to eight classes is already very encouraging.

The principle of context-sensitive discrimination was already integrated in VM [Fagan 84]. There, the number-to-symbol conversion depends on the current state of the patient/ventilator complex. Whenever a state transition is triggered, so-called initializing rules are invoked that redefine the upper and lower limits of parameter value ranges abstracted by symbols.

Fagan also suggests to use hystereses in triggering changes in therapy: in VM, different conditions are provided for transition from therapy mode A to therapy mode B and vice versa [Fagan 84]. This reflects a natural decision making behaviour, which will be discussed in more detail in Section 3.9.

Other authors have addressed context-sensitive discrimination in (slightly) different ways, which are not always clearly separated from idiosyncratic issues. In his approach to intelligent alarms, van der Aa employs adjustable feature baselines and a low and high threshold for each variable to discriminate normal from abnormal values [van der Aa 90]. The baseline is set to the current average by pressing a “reset baseline” button when the current average of the variable seems acceptable. Deviations from the average by a certain percentage are then reported as symbols *up* and *down*, respectively. Baseline reset is a simple, yet effective, means of taking account of patient-specific idiosyncrasies. In a first prototype, it has also been employed to adapt to a change in the monitoring environment such as adjustment of ventilator settings. In a second prototype this was replaced by an automated feature baseline reset, recalculation of baselines based on a model of the breathing circuit triggered by ventilator adjustments, which proved to work equally well [van der Aa 90].

3.7 Trend Detection

It has frequently been noted that the trend in a variable is more characteristic of the patient's condition than its absolute value. Trend detection therefore plays a crucial role in diagnostic monitoring.

Before turning to my approach to trend detection in Section 3.7.4, I give a short overview of what a trend is commonly regarded to be and which problems are associated with its detection.

3.7.1 Common Definitions of Trend

Many definitions of the term *trend* are in use. Definitions from the biomedical field include

- "general direction of the mean level of the data" ([Allen 83])
- "presence of a slow, consistent, unidirectional change in the value of a variable" (quoted in [Blom 85])
- "steadily rising or steadily falling pattern" ([Challis 90])
- "any change in the underlying signal dynamics slower than the system's time constants" (quoted in [Avent 90])
- "clinically significant pattern in a sequence of time-ordered data" ([Haimowitz 94])

Some of the definitions regard a trend as a feature that can be extracted from the course of a variable, while others suggest that a trend is defined as some kind of pattern that occurs in a signal. Consequently, depending on their definition the authors favour either feature extraction or pattern recognition methods to detect trends.

Feature extraction approaches to trend detection can be thought of as bottom-up, as they employ some (usually mathematically motivated) computation to derive signal(s) indicative of the trend. However, because these signals are not oriented towards some domain-specific abstraction of trends, they in turn require interpretation. Divided differences, derivatives, and regression are typical examples of this.

On the other hand, pattern recognition approaches to trend detection can be regarded as top-down, as they start with certain predetermined notions of trends (usually specified in the form of templates) that are continuously matched with the input in which the trend is to be detected. Their output is propositional in nature, i.e., it encodes, more or less directly, whether the input follows any of the expected trends.

Note that pattern recognition methods can be employed to classify trends derived by feature extraction methods. This is done by several authors, for example [Konstantinov 92, Sittig 92b].

3.7.2 Problems of Trend Detection

The problems encountered in trend detection are diverse and numerous. The following paragraphs sketch only a few; yet, no current trend detection method should be expected to solve all of them.

The Problem of Complexity

Although the basic problem of trend detection is similar to that of discrimination[17], it suffers from the combinatorial explosion induced by the temporal dimension.

Example

If the current blood pressure of a patient reads 100 mm Hg, sampling is discrete (resolution 1 mm Hg) and periodic, and the change rate of blood pressure is bounded by

[17] Briefly, discrimination is number-(i.e., one-sample-)to-symbol conversion, while trend detection is signal-(i.e., sequence-of-samples-)to-symbol conversion.

3 mm Hg per sampling period in either direction, the next ten consecutive samples result in 282,475,249 ($= 7^{10}$) theoretically possible different sequences of samples.

◊

Problems with Dense Samples

Dense samples reveal noise (which is usually unbounded) and other short-term fluctuations that superpose an underlying trend. However, their influence can largely be eliminated by respective filters (see Section 3.4 on data smoothing), so that dense samples are a comparatively good basis of trend detection.

Problems with Sparse Samples

The picture is different with sparse samples. Long periods of ignorance (compare Definition 2.4 of dense and sparse sampling) allow for significant deviation from any hypothesized trend, so that trend detection in sparsely sampled variables is inherently plagued by uncertainty.

The Problem of Superposition

A single variable may be indicative of different underlying trends the effects of which superpose. Sophisticated trend detection systems should be able to take account of possible superposition, for example by utilizing correlation with other, non-superimposed variables and by employing explicit models of superposition.

The Problem of Duration

Many trends are not fixed in duration, requiring a certain temporal flexibility of trend detection. Applied to feature extraction methods this means that the features have to be derived over a variable length interval. Applied to pattern recognition methods this means that the occurrences of patterns can be scaled, i.e., condensed or stretched, in time. Note that in either case the problem of duration adds to the problem of complexity.

The Problem of Graduation

Medical conceptions of trends are frequently “elastic” rather than rigid, which is justified by the fact that all clear-cut borders suffer from the kind of arbitrariness discussed in the context of fuzzy discrimination (Section 3.6.2). Trend detection is therefore more adequately realized as a matter of degree.

Miscellaneous Problems

There is no need to stress that context dependency of variables in which trends are to be detected requires context-specific trend detection. Frequently disregarded, however, remains the fact that most variables are physiologically bounded, and their behaviour at their ranges’ borders is different from that around their normal ranges.

3.7.3 Retrospective and Prospective Employment of Trend Detection

Depending on their employment, trend detection methods can emphasize different aspects of the same problem. In retrospective analysis of recorded data, for example, one may be particularly interested in when a certain trend occurred in the course of a variable.

Example

The time of onset of infection can play a crucial role in the diagnosis of infectious diseases. Although seropositive samples prove that an infection has taken place, the time of acquisition can only be determined by trend detection methods. A detailed example of this can be found in Chapter 6.

◊

On the contrary, in a prospective, on-line monitoring environment one is more interested in the current degree of presence of a trend—past occurrences are then subsumed by a state variable comprising the patient's disease history (see Section 3.9.1). In this case, the time of onset is fixed, namely tied to the current time.

Example

The proposition "blood pressure stable for one hour" derived in an on-line environment implies that the onset of the trend was one hour ago (from now).

◊

3.7.4 A Framework for Trend Detection

It seems reasonable (and compatible with all above definitions) to think of a trend as a (temporal) abstraction comprising courses with a meaning identical with respect to the specific monitoring problem. Following this understanding, the problem of detecting a trend in the course of a variable can be reduced to that of deciding whether some part of the course is properly abstracted by that trend. In particular, if the trend is represented by a set of courses, trend detection can be implemented as a test for set membership. This in turn is particularly efficient if the set is specified by some easily computable characteristic function.

The above considerations give rise to the following definition:

Definition 3.1 (trend)
A *trend* $\widetilde{C}_x$ on a variable $x : T_c \rightarrow V_x$ is a fuzzy set of courses specified by

$$\widetilde{C}_x = \{(c(t), \mu_{\widetilde{C}_x}(c(t))) | c : [0, d] \rightarrow V_x\},$$

where $\mu_{\widetilde{C}_x}$ is the membership function that decides for any course $c(t)$ to which degree it belongs to that trend. $\Delta\widetilde{C}_x = d$ is called the *duration of* $\widetilde{C}_x$ and is part of its specification.

■

Example

$\widetilde{C}_{bp} = \{(c(t), \mu) | c : [0, 10\,\text{min}] \rightarrow V_{bp}, \mu = 1 \text{ if } \forall 0 \le t_0 \le 10\,\text{min} : c'(t_0) > 0, \mu = 0 \text{ else}\}|$

is the specification of the trend "rising blood pressure for ten minutes" comprising all courses whose first derivative is positive over an interval of ten minutes.

◊

Note that the definition of trends is based on continuous-time courses. It is independent of the representation of the courses of variables they are to be detected in.

Definition 3.1 implies that every trend has a fixed duration, a restriction that may not be acceptable in the medical context. Section 3.7.9 will discuss the problem of variable length trends.

Finally note that Definition 3.1 gives trends the form of normalized templates, as all constituent courses start at time zero. Trend detection will thus involve a temporal offset, the time at which the trend begins to occur in the observed variable, the so-called *time of onset*. The occurrence of a trend in a continuous-time signal is then defined as follows:

Definition 3.2 (occurrence of a trend)
The degree ω to which a trend $\widetilde{C}_x$ *occurs in the course of a variable* $x(t)$ *beginning at a time of onset* t_Ω is defined as

$$\omega(\widetilde{C}_x, x(t), t_\Omega) = \mu_{\widetilde{C}_x}(c(t)) \quad \text{where } c(t_0) = x(t_0 + t_\Omega) \text{ for all } t_0 \in [0, \Delta\widetilde{C}_x] \quad .$$

■

In the discrete-time environment of a digital monitor the occurrence of a trend can at best be tested indirectly. In fact, if no perfect (or at least near perfect) reconstruction method is applicable, it can generally not be decided if a given sequence of samples represents a course in which the trend occurs. This is particularly true if sampling is sparse so that the value of a variable may change significantly and unpredictably during unsampled periods. Trend detection then invariably makes guesses about unobserved intervals.

Definition 3.3 (compatibility with a trend)
The *degree of compatibility* γ *of a sequence of samples* $\langle x[t_n]\rangle$ *with the trend* $\widetilde{C}_x$ *relative to a time of onset* t_Ω is defined as

$$\gamma(\langle x[t_n]\rangle, \widetilde{C}_x, t_\Omega) = \sup_{x_e(t)} \omega(\widetilde{C}_x, x_e(t), t_\Omega)$$

■

The explaining course required in the definition of compatibility presents a guess about the true course of the variable. Trend detection as defined here in terms of compatibility is opportunistic in that it chooses explaining courses that best suit the trend to be detected. On the other hand, a medically sound specification of trends can guarantee that no unphysiological explaining courses are relied upon to explain the findings.

Once again, compatibility of a sequence of samples with a trend does not necessarily imply that the trend really occurs in the sampled continuous-time variable—alternative conditions under which compatibility guarantees actual presence of a trend are listed as follows:

1. The true course can be perfectly reconstructed as $x_r(t)$ and the trend occurs in $x_r(t)$.
2. Some valid model of the signal helps to constrain the set of courses $x_e(t)$ explaining $\langle x[t_n]\rangle$ and the trend occurs in all of them.
3. The collection of trends specified for the variable together covers the whole spectrum of possible courses, and $\langle x[t_n]\rangle$ is only compatible with one of them (and not with any other).

If one of the condition holds, trend detection is an implicit form of signal reconstruction: classifying a sequence of samples as being compatible with only one trend (excluding all others) means that the actual course has been reconstructed as being one of the trend's constituent courses. Accuracy of reconstruction then depends on the specificity of the trend.

However, because none of the above generally applies, trend detection is a major source of uncertainty in monitoring.

3.7.5 Trend Detecting Systems

Embedded in the monitoring framework of Chapter 2, a trend detector is a system that continuously derives the degree of compatibility of the current sequence of samples with its internal specification of a trend:

$$y[t_n] = \gamma(\langle x[t_n]\rangle, \widetilde{C}_x, t_n - \Delta\widetilde{C}_x) \quad .$$

Note that trend detectors are window-based systems: the window $[t_\Omega, t_\Omega + \Delta\widetilde{C}_x]$ implicit in the definition of compatibility is tied to t_n such that $t_\Omega = t_n - \Delta\widetilde{C}_x$.

If the trend detector "knows" a family of trends $\widetilde{C}_{i,x}$ on the same variable x, its definition

$$y[t_n] = \{\,(\widetilde{C}_{i,x}, \gamma(\langle x[t_n]\rangle, \widetilde{C}_{i,x}, t_n - \Delta\widetilde{C}_{i,x}))\}$$

is remarkably similar to (3.11), the definition of a fuzzy discriminator. In fact, the only difference between trend detection as realized below and fuzzy discrimination is that the

specification of a trend is that of a class extended by the temporal dimension covering the course of a variable over time.

Example

Instead of classifying the instantaneous value of a variable as *low*, *medium*, or *high*, the course of the same variable may be classified as *low*, *medium*, or *high for ten minutes*, respectively.

◊

3.7.6 Fuzzy Courses and Trends

After precise definitions of trend, occurrence of a trend in a signal, and compatibility of a sequence of samples with a trend have been presented, practical means of specifying trends are required. Recalling that a trend is defined as a fuzzy set which in turn is entirely specified by its membership function, all that needs to be done is to devise a suitable format of membership functions. However, this function need not only be easy to specify, it also has to satisfy computational requirements, which are nontrivial regarding the problems addressed in Section 3.7.2. The specification of the membership function must therefore be carefully chosen.

Recall that fuzzy numbers are sometimes employed to fuzzily restrict the range of a variable the exact value of which is not precisely known (Section 1.2.2). (Note again that this employment of fuzzy sets is different from the employment in fuzzy discriminators, where a fuzzy set denotes the unity of elements comprised under one term.) If the concept of fuzzy numbers is extended by the temporal dimension, one obtains what I call a *fuzzy course*.

Definition 3.4 (fuzzy course)
A *fuzzy course* of a variable $x : T_c \to V_x$ with $V_x \subset \Re$ is specified by a function $\tilde{x} : T_c \to \tilde{\Re}$, where $\tilde{\Re}$ is the set of normalized and convex fuzzy subsets of $\Re$ (fuzzy numbers).

■

Note that fuzziness extends only in the value dimension—the time scale, the independent variable, remains crisp.

Just like a fuzzy number can be interpreted as a restriction of the instantaneous value of a variable to a fuzzy set, a fuzzy course is a fuzzy restriction of the course of a variable. In particular, a fuzzy course expresses the fact that the value of the variable x at time t_0 is not precisely known, yet fuzzily restricted by $\tilde{x}(t_0)$. Graphically, fuzzy courses can be depicted as shown in Figure 3.12. All three depictions are different aspects of the same (trapezoidal) fuzzy course.

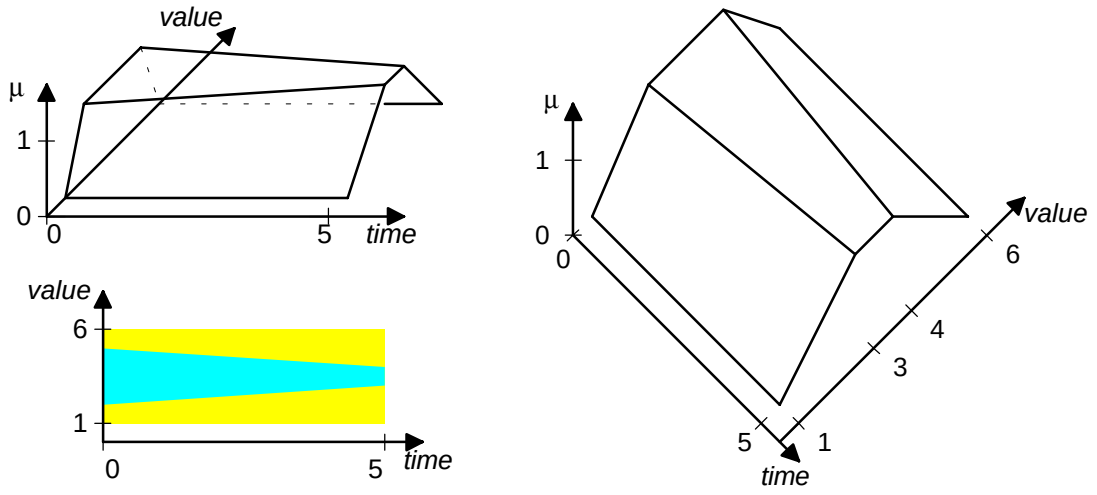


Figure 3.12: Three different depictions of the same fuzzy course; the transition from zero to one in the two-dimensional representation (left, bottom) is not visible and to be considered implicit

Figure 3.12 suggests that a fuzzy course can be interpreted as a fuzzy relation $\tilde{R}$ on $T_c \times V_x$ where $\mu_{\tilde{R}}(t, v) = \mu_{\tilde{x}(t)}(v)$. It should be noted, however, that not every binary fuzzy relation on the Cartesian product of time and value of a variable makes a fuzzy course: the relation must be a "fuzzy function", i.e., the set of values related with a point in time must be a fuzzy number, which is a fuzzy relaxation of the to-one mapping property required for ordinary functions.

Recall that the information model of Section 2.2 does not provide for the representation of findings as fuzzy numbers, nor for the modelling of signals as fuzzy courses.[18] Instead, fuzzy courses will be employed here to specify trends.

If $\tilde{x}(t)$ is a fuzzy course defined on the interval $[0, d]$, then a fuzzy trend $\tilde{C}_x$ with $\Delta\tilde{C}_x = d$ is defined in terms of $\tilde{x}(t)$ by

$$\tilde{C}_x = \left\{ (c(t), \mu) \middle| \mu = \inf_{t_0 \in [0,d]} \mu_{\tilde{x}(t_0)} c(t_0)) \right\}$$

or, equivalently,

$$\mu_{\tilde{C}_x}(c(t)) = \inf_{t_0 \in [0,d]} \mu_{\tilde{x}(t_0)} c(t_0). \qquad (3.12)$$

Figure 3.13 a) through c) show the fuzzy course of Figure 3.12 defining a trend and three signals with full, gradual and non-membership, respectively.

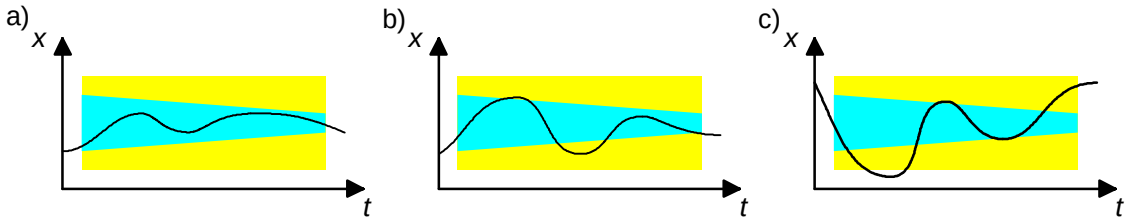


Figure 3.13: Fuzzy course specifying a trend and three signals
a) with full (degree = 1), b) partial (degree = 0.5), and c) non-membership (degree = 0) in trend

Definition 3.3 and (3.12) suggest that deriving the compatibility of a sequence of samples with a trend specified via a fuzzy course is computationally expensive. However, the following theorem allows an extremely efficient computation of compatibility:

Theorem 3.5
If $\tilde{C}_x$ is a trend defined by a fuzzy course as in (3.12), then

$$\gamma(\langle x[t_n] \rangle, \tilde{C}_x, t_\Omega) = \min_{t_m : t_\Omega \le t_m \le t_\Omega + \Delta\tilde{C}_x} \mu_{\tilde{x}(t_m - t_\Omega)}(x[t_m]) \quad .$$

■

In words: the compatibility of a sequence of samples with a trend specified by a fuzzy course is determined by the smallest degree of membership of a sample $x[t_m]$ in the fuzzy course at t_m.

Proof

(3.12) allows it to rewrite Definition 3.2 as

$$\omega(\tilde{C}_x, x(t), t_\Omega) = \inf_{t_0 \in [0, \Delta\tilde{C}_x]} \mu_{\tilde{x}(t_0)}(x(t_0 + t_\Omega)) \quad . \qquad (3.13)$$

Because $\tilde{x}(t_0)$ is a normalized fuzzy set for all t_0, there must be some ideal course $x_i(t)$ such that $\mu_{\tilde{x}(t_0)}(x_i(t_0)) = 1$ for all t_0. For the explaining course $x_{\hat{e}}(t)$

[18] Yet, it could; see, for example, [Steimann 94b].

$$x_{\hat{e}}(t) = \begin{cases} x[t_n] & \text{if } t = t_n \\ x_i(t) & \text{else} \end{cases} \tag{3.14}$$

and for all other explaining courses $x_e(t)$ of $\langle x[t_n] \rangle$

$$\inf_{t_0 \in [0, \Delta\widetilde{C}_x]} \mu_{\widetilde{x}(t_0)}(x_{\hat{e}}(t_0 + t_\Omega)) \geq \inf_{t_0 \in [0, \Delta\widetilde{C}_x]} \mu_{\widetilde{x}(t_0)}(x_e(t_0 + t_\Omega)) \tag{3.15}$$

obviously holds. Because of (3.13), (3.15) and (3.14)

$$\begin{aligned} \gamma(\langle x[t_n] \rangle, \widetilde{C}_x, t_\Omega) &= \sup_{x_e(t)} \omega(\widetilde{C}_x, x_e(t), t_\Omega) \\ &= \sup_{x_e(t)} \inf_{t_0 \in [0, \Delta\widetilde{C}_x]} \mu_{\widetilde{x}(t_0)}(x_e(t_0 + t_\Omega)) \\ &= \inf_{t_0 \in [0, \Delta\widetilde{C}_x]} \mu_{\widetilde{x}(t_0)}(x_{\hat{e}}(t_0 + t_\Omega)) \\ &= \inf_{t_0 \in [t_\Omega, t_\Omega + \Delta\widetilde{C}_x]} \mu_{\widetilde{x}(t_0 - t_\Omega)}(x_{\hat{e}}(t_0)) \\ &= \min_{t_m : t_\Omega \leq t_m \leq t_\Omega + \Delta\widetilde{C}_x} \mu_{\widetilde{x}(t_m - t_\Omega)}(x[t_m]) \end{aligned}$$

■

Note that the degree of compatibility is easily computed; a constant implementation of $\widetilde{x}(t)$ provided, computational effort is linear in the number of samples included in the window.

3.7.7 Floating Level Trend Detection

Trends specified via fuzzy courses as shown above constrain absolute signal values rather than relative changes, a property that makes them unsuited for certain clinical trends.

Example

The trend “heart rate stable for ten minutes” is characterized by a fairly constant course of the heart rate, no matter at which level.

◊

The behaviour required to neglect constant offsets is adopted if

$$\gamma(\langle x[t_n] + b \rangle, \widetilde{C}_x, t_\Omega) = \gamma(\langle x[t_n] \rangle, \widetilde{C}_x, t_\Omega) \quad \text{for all } b \in V_x \quad . \tag{3.16}$$

Such is naturally the case if $\widetilde{C}_x$ is specified via the first derivative as discussed in the next section. This section presents a method that automatically derives a constant offset so that the sequence of samples best matches the trend. Doing so effectively leaves the offset of a course variable.

Definition 3.6 (floating level compatibility)
Floating level compatibility γ_f of a sequence of samples with a trend is defined as

$$\gamma_f(\langle x[t_n] \rangle, \widetilde{C}_x, t_\Omega) = \sup_b \gamma(\langle x[t_n] - b \rangle, \widetilde{C}_x, t_\Omega) \; . \tag{3.17}$$

■

Note that this definition of γ_f satisfies (3.16). Although computation of γ_f appears to be computationally expensive, a few considerations lead to a solution that is linear in the number of samples involved, given that the trend is specified via a fuzzy course as in (3.12) and intersection of fuzzy numbers can be performed in constant time.

Applying Theorem 3.5 to (3.17) yields

$$\gamma_f(\langle x[t_n] \rangle, \widetilde{C}_x, t_\Omega) = \sup_b \min_{t_m : t_\Omega \leq t_m \leq t_\Omega + \Delta\widetilde{C}_x} \mu_{\widetilde{x}(t_m - t_\Omega)}(x[t_m] - b) \, . \tag{3.18}$$

Next consider a single sample $x[t_m]$. Its compatibility γ with a trend specified by the fuzzy course $\tilde{x}(t)$ relative to t_Ω depends on b such that

$$\gamma(b) = \mu_{\tilde{x}(t_m - t_\Omega)}(x[t_m] - b) =: \mu_{\tilde{b}_m}(b), \tag{3.19}$$

which is itself the membership function of a fuzzy number labelled $\tilde{b}_m$. For each sample $x[t_m]$ from the window $[t_\Omega, t_\Omega + \Delta\tilde{C}_x]$ such a fuzzy set $\tilde{b}_m$ can be found. Offsets for which all samples from the window are compatible with the trend are then derived as the intersection of all $\tilde{b}_m$, and the maximum compatibility as the height of that intersection so that

$$\gamma_f(\langle x[t_n]\rangle, \tilde{C}_x, t_\Omega) = \text{hgt}\left(\bigcap_{t_m : t_\Omega \le t_m \le t_\Omega + \Delta\tilde{C}_x} \tilde{b}_m \right),$$

which is the fuzzy-set-theoretic interpretation of (3.18).

Example

Consider the fuzzy trend defined by a trapezoidal fuzzy course as depicted in Figure 3.14 a). Compatibility of the sequence of samples also shown in Figure 3.14 a) is then derived as the height of the intersection of the fuzzy numbers $\tilde{b}_1, \tilde{b}_2$, and $\tilde{b}_3$ derived as in (3.19), which is shown in Figure 3.14 b). Finally, Figure 3.14 c) shows that translating the course by *b'* best matches the findings.

◊

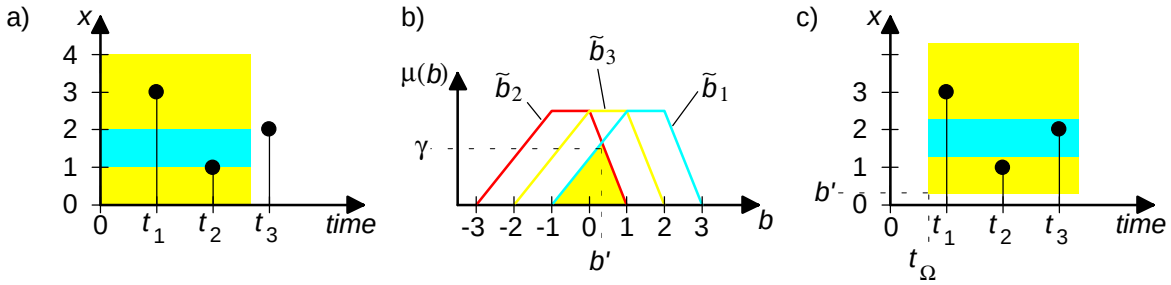


Figure 3.14: Floating level trend detection
a) sequence of three samples and (trapezoidal) fuzzy course specifying the trend to be detected
b) fuzzy numbers constraining the offset *b* that makes individual findings of a) compatible with the trend; the intersection of all (grey triangle) denotes the offset *b´* that leads to the maximum degree of compatibility γ
c) trend offset by *b´* best matching all samples

3.7.8 Fuzzy First Derivatives and Trends

Natural language specifications of trends such as “rising”, “steady”, or “falling” suggest that trends constrain the first derivative (also informally called *slope*) of a signal rather than its absolute course. The first derivative of a continuous-time signal $x(t)$ is itself a continuous-time signal defined as

$$x'(t) = \frac{dx(t)}{dt}.$$

Derivatives ignore constant offsets of signals so that

$$\frac{d(x(t)+b)}{dt} = \frac{dx(t)}{dt} \quad \text{for arbitrary } b,$$

a property that is often desired in trend detection, particularly if b models some context dependency or patient-specific idiosyncrasy.

As pointed out before, the first derivative cannot be derived for a general sequence of samples (see Section 3.5.2). This makes the matching of a trend specified via its courses’ derivatives with a sequence of samples difficult. The employment of explaining courses for trend detection

is an elegant solution to the problem: an explaining course, while subsuming all samples, is continuous and can therefore be derived. The derivative of the explaining course then substitutes for the sequences of samples in checking for compliance with the trend specification.

Just like the absolute course, the first derivative can be specified fuzzily. If $\tilde{x}'(t_0)$ denotes the fuzzy number restricting the first derivative at time t_0, a trend $\tilde{C}_x$ can be specified in analogy to (3.12) by relating its membership function to that of a fuzzy course $\tilde{x}'(t)$, namely,

$$\mu_{\tilde{C}_x}(c(t)) = \inf_{t_0 \in [0, \Delta\tilde{C}_x]} \mu_{\tilde{x}'(t_0)}(c'(t_0)). \tag{3.20}$$

Despite its similar definition, compatibility is not as easily computed as in Theorem 3.5. However, as will be proved, compatibility can be derived efficiently if $\tilde{x}'(t)$ is defined as follows.

Consider two bivariate functions $x'_l(\alpha, t)$ and $x'_h(\alpha, t)$ totally defined on the temporal interval $[0, d]$ such that for all t_0

$x'_l(\alpha, t_0)$ is a continuous and strictly monotonically increasing function of $\alpha \in [0, 1]$,

$x'_h(\alpha, t_0)$ is a continuous and strictly monotonically decreasing function of $\alpha \in [0, 1]$,

and

$$x'_l(1, t_0) \le x'_h(1, t_0) \quad \text{for all } t_0 \in [0, d] .$$

The fuzzy first derivative $\tilde{x}'(t)$ can then be defined in terms of its α-cut by

$$\tilde{x}'(t_0)_\alpha = [x'_l(\alpha, t_0), x'_h(\alpha, t_0)] \quad \text{for all } t_0 \in [0, d] \text{ and } 0 < \alpha \le 1$$

so that

$$\mu_{\tilde{x}'(t_0)}(x) = \sup_{\alpha \in]0,1]} \left\{ \begin{array}{ll} \alpha & \text{if} \quad x \in \tilde{x}'(t_0)_\alpha \\ 0 & \text{else} \end{array} \right\} \text{for all } t_0 \in [0, d] \tag{3.21}$$

(compare Section 1.2.1).

Example

Immune response to infection includes the production of specific antibodies. Antibody concentration in human serum takes typical courses: after an initial rising phase beginning soon after onset of infection, antibody concentration decays and remains on a certain level which may persist for the whole lifetime. Let the trend associated with typical courses of antibody concentration be specified by the fuzzy first derivative $\tilde{x}'(t)$ constructed as in (3.21) by the two functions

$$x'_l(\alpha, t) = 1 + \alpha + \left(\tfrac{\alpha}{10} - 0.6 \right) t \quad \text{and} \quad x'_h(\alpha, t) = 6 - \alpha - \left(\tfrac{\alpha}{10} + 0.6 \right) t$$

defined on the temporal interval $[0, 10]$. Figure 3.15 a) shows $x'_l(\alpha, t)$ and $x'_h(\alpha, t)$ for $\alpha = 0$ and $\alpha = 1$, respectively. Figure 3.15 b) depicts $x'_l(\alpha, t_0)$ and $x'_h(\alpha, t_0)$ for some t_0. Figure 3.15 c) gives a depiction of $\tilde{x}'(t)$ as a whole.

◊

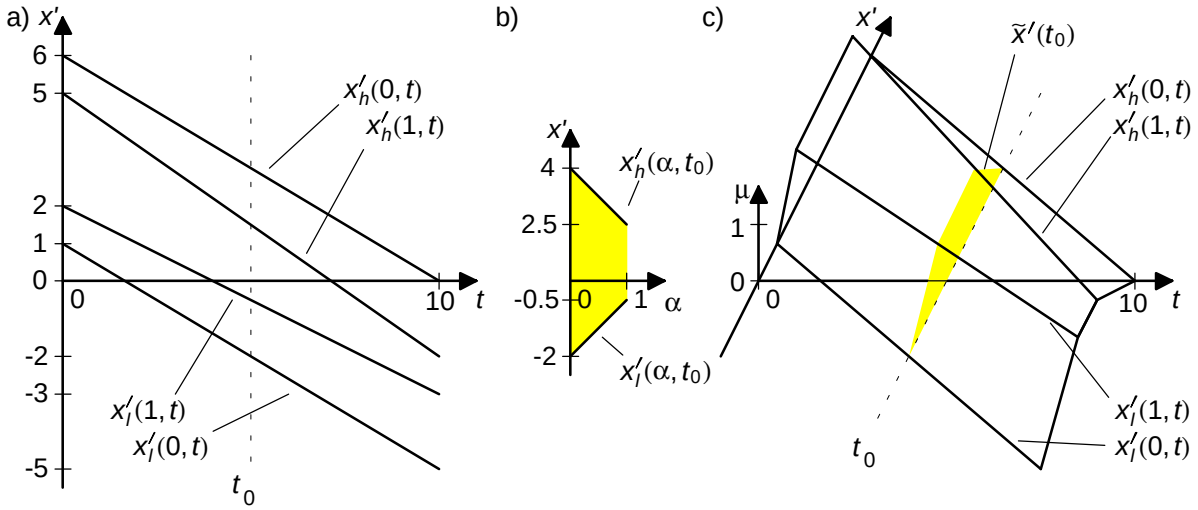


Figure 3.15: Specification of a fuzzy course according to (3.21)
a) $x'_l(\alpha,t)$ and $x'_h(\alpha,t)$ for $\alpha=0$ and $\alpha=1$, respectively
b) $x'_l(\alpha,t_0)$ and $x'_h(\alpha,t_0)$ for some t_0
c) $\tilde{x}'(t)$

Theorem 3.7
If $\tilde{C}_x$ is a trend specified via a fuzzy first derivative $\tilde{x}'(t)$ defined as in (3.21), the degree of compatibility of a sequence of samples $\langle x[t_n]\rangle$ with $\tilde{C}_x$ relative to the time of onset t_Ω can be derived as

$$\gamma(\langle x[t_n]\rangle, \tilde{C}_x, t_\Omega) = \min_{m: t_\Omega \le t_m < t_{m+1} \le t_\Omega + \Delta\tilde{C}_x} \gamma((x[t_m], x[t_{m+1}]), \tilde{C}_x, t_\Omega) \quad ,$$

where

$$\gamma((x[t_m], x[t_{m+1}]), \tilde{C}_x, t_\Omega) = \begin{cases} 0 \text{ if } x[t_{m+1}] - x[t_m] < \int_{t_m}^{t_{m+1}} x'_l(0, t-t_\Omega)dt \\ \alpha \text{ if } x[t_{m+1}] - x[t_m] = \int_{t_m}^{t_{m+1}} x'_l(\alpha, t-t_\Omega)dt \quad \text{for some } \alpha \in [0,1[\\ 1 \text{ if } x[t_{m+1}] - x[t_m] \in \left[\int_{t_m}^{t_{m+1}} x'_l(1, t-t_\Omega)dt, \int_{t_m}^{t_{m+1}} x'_h(1, t-t_\Omega)dt\right] \\ \alpha \text{ if } x[t_{m+1}] - x[t_m] = \int_{t_m}^{t_{m+1}} x'_h(\alpha, t-t_\Omega)dt \quad \text{for some } \alpha \in [0,1[\\ 0 \text{ if } x[t_{m+1}] - x[t_m] > \int_{t_m}^{t_{m+1}} x'_h(0, t-t_\Omega)dt \end{cases}$$

■

Proof

The proof is based on a few fundamental observations. Verify that from (3.21) it follows that

$$\mu_{\tilde{x}'(t_0)}(x) = \left\{ \begin{array}{l} 0 \text{ if } x < x'_l(0,t_0) \\ \alpha \text{ if } x = x'_l(\alpha,t_0) \quad \text{for some } \alpha \in [0,1[\\ 1 \text{ if } x \in [x'_l(1,t_0), x'_h(1,t_0)] \\ \alpha \text{ if } x = x'_h(\alpha,t_0) \quad \text{for some } \alpha \in [0,1[\\ 0 \text{ if } x > x'_h(0,t_0) \end{array} \right\} \text{for all } t_0$$

and consequently from (3.20) that

$$\begin{aligned}\mu_{\widetilde{C}_x}(\int x_l'(\alpha,t)dt) &= \inf_{t_0\in[0,d]} \mu_{\widetilde{x}'(t_0)}(x_l'(\alpha,t_0)) = \alpha \\ \mu_{\widetilde{C}_x}(\int x_h'(\alpha,t)dt) &= \inf_{t_0\in[0,d]} \mu_{\widetilde{x}'(t_0)}(x_h'(\alpha,t_0)) = \alpha\end{aligned} \tag{3.22}$$

Also verify that the following two implications hold:

$$x_1(t_0) > x_2(t_0) \quad \text{for all } t_0 \in [t_m, t_{m+1}] \quad \Rightarrow \quad \int_{t_m}^{t_{m+1}} x_1(t)dt > \int_{t_m}^{t_{m+1}} x_2(t)dt \tag{3.23}$$

$$\int_{t_m}^{t_{m+1}} x_1(t)dt > \int_{t_m}^{t_{m+1}} x_2(t)dt \quad \Rightarrow \quad x_1(t_0) > x_2(t_0) \quad \text{for some } t_0 \in [t_m, t_{m+1}] \tag{3.24}$$

The explaining course required in the definition of compatibility (Definition 3.3) can now be constructed piecewisely from fragments $x_{e,m}(t)$ explaining a pair of consecutive samples $(x[t_m], x[t_{m+1}])$ each, so that

$$x_e(t_0) = x_{e,m}(t_0) \quad \text{for all } t_0 \in [t_m, t_{m+1}].$$

Because $x_e(t)$ is not generally continuously differentiable at t_m, i.e.,

$$\lim_{h\to 0} \frac{x_e(t_m - h) - x_e(t_m)}{h} \neq \lim_{h\to 0} \frac{x_e(t_m + h) - x_e(t_m)}{h},$$

I define

$$x_e'(t_m) := x_{e,m}'(t_m),$$

which leaves $x_e'(t)$ discontinuous, yet totally defined.

(3.20) then allows it to rewrite Definition 3.3 to

$$\begin{aligned}\gamma(\langle x[t_n]\rangle, \widetilde{C}_x, t_\Omega) &= \sup_{x_e(t)} \inf_{t_0\in[0,\Delta\widetilde{C}_x]} \mu_{\widetilde{x}'(t_0)}(x_e'(t_0 + t_\Omega)) \\ &= \min_{m: t_\Omega \le t_m \le t_{m+1} \le t_\Omega + \Delta\widetilde{C}_x} \sup_{x_{e,m}(t)} \inf_{t_0\in[t_m,t_{m+1}]} \mu_{\widetilde{x}'(t_0 - t_\Omega)}(x_{e,m}'(t_0)) \\ &= \min_{m: t_\Omega \le t_m < t_{m+1} \le t_\Omega + \Delta\widetilde{C}_x} \gamma((x[t_m], x[t_{m+1}]), \widetilde{C}_x, t_\Omega)\end{aligned}$$

The compatibility of two consecutive findings with the trend is then derived as follows:

1.)

$$\begin{aligned}& x[t_{m+1}] - x[t_m] < \int_{t_m}^{t_{m+1}} x_l'(0, t - t_\Omega)dt \\ \Rightarrow\ & \int_{t_m}^{t_{m+1}} x_{e,m}'(t)dt < \int_{t_m}^{t_{m+1}} x_l'(0, t - t_\Omega)dt \quad \text{for all } x_{e,m}'(t)dt \\ \Rightarrow\ & x_{e,m}'(t_0) < x_l'(0, t_0 - t_\Omega) \quad \text{for some } t_0 \in [t_m, t_{m+1}] \\ \Rightarrow\ & \inf_{t_0\in[0,\Delta\widetilde{C}_x]} \mu_{\widetilde{x}'(t_0)}(x_{e,m}'(t_0 + t_\Omega)) \le \inf_{t_0\in[0,\Delta\widetilde{C}_x]} \mu_{\widetilde{x}'(t_0)}(x_l'(0, t_0)) = 0 \\ \Rightarrow\ & \omega(\widetilde{C}_x, \int x_{e,m}'(t)dt, t_\Omega) = 0 \quad \text{for all } x_{e,m}'(t)dt \\ \Rightarrow\ & \gamma((x[t_m], x[t_{m+1}]), \widetilde{C}_x, t_\Omega) = 0\end{aligned}$$

2.)

$$x[t_{m+1}] - x[t_m] = \int_{t_m}^{t_{m+1}} x_l'(\alpha, t - t_\Omega)dt \quad \text{for some } \alpha \in\ ]0, 1[$$

and (3.22) imply that

$$x_{\hat{e},m}(t) = x[t_m] + \int_{t_m}^{t} x_l'(\alpha, u - t_\Omega)du$$

is an explaining course with

$$\omega(\widetilde{C}_x, x_{\hat{e},m}(t), t_\Omega) = \alpha \ .$$

(3.20), (3.21) and (3.23) imply that for all courses $x'(t)$

$$\omega(\widetilde{C}_x, x'(t), t_\Omega) > \alpha \quad \Rightarrow \quad x[t_m] + \int_{t_m}^{t_{m+1}} x'(t)dt > x[t_{m+1}]$$

so that $x'(t)$ does not explain $(x[t_m], x[t_{m+1}])$. Consequently,

$$\gamma((x[t_m], x[t_{m+1}]), \widetilde{C}_x, t_\Omega) = \alpha \ .$$

3.)

$$x[t_{m+1}] - x[t_m] \in \left[\int_{t_m}^{t_{m+1}} x_l'(1, t - t_\Omega)dt, \int_{t_m}^{t_{m+1}} x_h'(1, t - t_\Omega)dt \right]$$

allows it to construct an explaining course

$$x_{\hat{e},m}(t) = x_l(t) + \frac{x[t_{m+1}] - x_l(t_{m+1})}{x_h(t_{m+1}) - x_l(t_{m+1})}(x_h(t) - x_l(t))$$

where

$$x_l(t) = x[t_m] + \int_{t_m}^{t} x_l'(1, u - t_\Omega)du \quad \text{and} \quad x_h(t) = x[t_m] + \int_{t_m}^{t} x_h'(1, u - t_\Omega)du \, .$$

Verify that $x_{\hat{e},m}(t)$ is an explaining course, because

$$x_{\hat{e},m}(t_m) = x[t_m] \quad \text{and} \quad x_{\hat{e},m}(t_{m+1}) = x[t_{m+1}],$$

and that

$$\omega(\widetilde{C}_x, x_{\hat{e},m}(t), t_\Omega) = 1 \ .$$

Consequently,

$$\gamma((x[t_m], x[t_{m+1}]), \widetilde{C}_x, t_\Omega) = 1 \ .$$

4.)

analogous to 2.)

5.)

analogous to 1.)

■

Note that efficient computation depends on the reversibility of $x_l'(\alpha, t)$ and $x_h'(\alpha, t)$, which is (at least theoretically) guaranteed by the required properties of the functions. Practical specifications of the functions allow a straightforward calculation of compatibility, as the following example demonstrates.

Example

Continuing the previous example, now consider a sequence of three consecutive samples $(x[t_1] = 8, x[t_2] = 8, x[t_3] = 7)$, which are shown in Figure 3.16. For convenience, the sampling sequence is specified relative to an assumed time of onset t_Ω, namely, $(t_1, t_2, t_3) = (t_\Omega + 1, t_\Omega + 7, t_\Omega + 9)$. Note that the samples suggest a fairly steady course: the mean slope between t_1 and t_2 is zero and only slightly falling between t_2 and t_3.

However, as the following calculation demonstrates, the samples are well compatible with the trend specified via its derivative depicted in Figure 3.15

Application of Theorem 3.7 then yields

$$\begin{aligned} x[t_2]-x[t_1] &= \int_{t_1}^{t_2} x_l'(\alpha, t-t_\Omega)dt \\ \Rightarrow \quad 8-8 &= \left(\tfrac{5}{100}\alpha-\tfrac{3}{10}\right)t^2+(1+\alpha)t\Big|_1^7 \\ \Rightarrow \quad \alpha &= 1 \\ \Rightarrow \quad x_{\hat{e},1}(t) &= -\tfrac{1}{4}t^2+2t+\tfrac{25}{4} \end{aligned}$$

and

$$\begin{aligned} x[t_3]-x[t_2] &= \int_{t_2}^{t_3} x_h'(\alpha, t-t_\Omega)dt \\ \Rightarrow \quad 7-8 &= \left(-\tfrac{5}{100}\alpha-\tfrac{3}{10}\right)t^2+(6-\alpha)t\Big|_7^9 \\ \Rightarrow \quad \alpha &= \tfrac{17}{18} \\ \Rightarrow \quad x_{\hat{e},2}(t) &= -\tfrac{25}{72}t^2+\tfrac{91}{18}t-\tfrac{83}{8} \end{aligned}$$

so that the degree of compatibility can be derived as

$$\gamma = \min\left(1, \tfrac{17}{18}\right) = \tfrac{17}{18}$$

Figure 3.16 shows the respective explaining courses (dashed lines) and the composed course (black line, partly superposing dashed lines).

◊

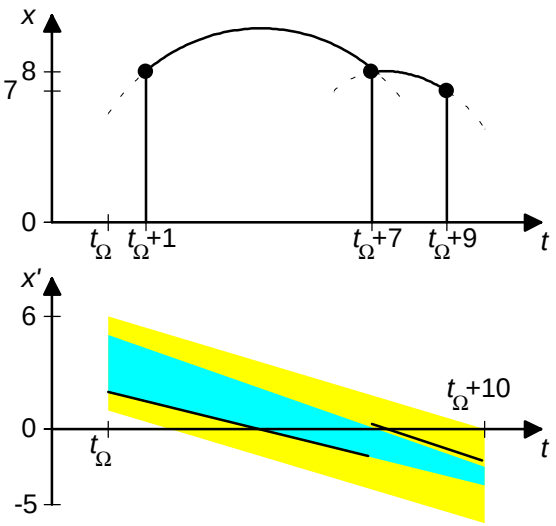


Figure 3.16: Detection of a trend specified by its first derivative; the trend is the same as that in Figure 3.15 the solid line indicates the explaining course composed of the dotted fragments; note the discontinuity in the slope of the explaining course

The concerned reader may argue that a fuzzy first derivative is nontrivial to specify and that the resulting explaining course is not very natural. Practically, however, the fuzzy first derivative will usually be specified as piecewise constant leading to piecewise linear explaining courses, which seems more natural.[19] A convincing example of this can be found in Chapter 6, Figure 6.1

3.7.9 Other Approaches to Trend Detection

The alert reader will have noticed that the TIMEEXP premise function of VM [Fagan 84] is a special case of trend detection via fuzzy courses.

Example

The rule premise

heart rate is ACCEPTABLE for greater than 30 minutes

requires the values of *heart rate* observed during the last 30 minutes to lie in the range specified by ACCEPTABLE, which may be regarded as a cubic fuzzy course.

◊

Note that the definition of ACCEPTABLE is context-dependent in VM, the TIMEEXP function thus being context-sensitive. However, the range of trends that can be expected is very limited, and it suffers from the inherent discontinuity (a minor violation suffices to discard a trend).

Curve Fitting and Regression

Curve fitting and regression-based methods aim at identifying polynomials

$$x(t) = a_0 + a_1 t + a_2 t^2 + \ldots \tag{3.25}$$

that best explain a given sequence of samples. Theoretically, a sequence of n samples uniquely determines the coefficients of a polynomial of n-1th order that passes through all n samples. Practically, however, the number of coefficients to be determined is predetermined and independent of the sample size, for example

$$x(t) = a_0 + a_1 t \tag{3.26}$$

or

$$x(t) = a_1 e^{a_2 t},$$

the latter implying all coefficients of (3.25). Various methods of curve fitting can be found in text books of numerical analysis, for example [Fröberg 85]. The following will deal only with regression and its application in diverse monitoring projects.

Rather than trying to find a polynomial course passing through all samples (which does not generally exist if the number of samples to be explained exceeds the number of coefficients that can be adjusted), regression makes do with an approximation minimizing some measure of deviation. For example, linear regression using the minimum squared error criterion aims at identifying the coefficients of (3.26) so that

$$E = \sum_n (x[t_n] - (a_0 + a_1 t_n))^2$$

is minimized [Challis 90]. The success of trend detection is not reflected in the derived coefficients, but in the error E.

Regression is made the basis of trend detection in several monitoring projects, for example [Konstantinov 92, Koski 92, Sittig 92b, Haimowitz 94]. Interestingly, in earlier versions of TRENDX a pair of functions representing value constraints similar to fuzzy courses, only non-fuzzy, were employed [Haimowitz 93a, 93b]. The current version favours low order regression, and it is noted that regression-based TRENDX, yielding a gradual measure of fit, is more robust and allows ranking of hypotheses, as opposed to its constraint-based predecessor, where a single sample "out of bounds" suffices to reject a hypotheses [Haimowitz 94]. It may be added that using fuzzy constraints such as fuzzy courses would have had a similar effect.

[19] The reason why linear functions where chosen for this example is that the equations are easy to solve.

[Sittig 92b] employs fuzzy classification of various features derived in applying linear regression, including slope of the variable, goodness of fit, and duration of the trend.

There is a certain relationship between trend detection via regression and trend detection via fuzzy courses as defined above. Suppose that $x_a(t)$ is a polynomial with fixed coefficients and that

$$E = \max_n |x_a(t_n) - x[t_n]|$$

is the measure of error, then

$$x_a(t) + \widetilde{E}, \tag{3.27}$$

where $\widetilde{E}$ is a symmetrical fuzzy number centred around zero, defines a fuzzy course, and

$$\mu_{\widetilde{E}}(E)$$

is the degree of compatibility of a sequence with the trend (all factors concerning the time of onset have been omitted for the sake of clarity).

Alternatively, the coefficients of the polynomial can be restricted by fuzzy numbers $\widetilde{a}_0, \ldots, \widetilde{a}_n$, so specifying a fuzzy set of courses making up a trend

$$\widetilde{C}_x = \left\{ (c(t), \mu) \,\middle|\, c(t) = a_0 + a_1 t + \ldots + a_n t^n \ \wedge\ \mu = \min_n \mu_{\widetilde{a}_n}(a_n) \right\} \tag{3.28}$$

(compare Definition 3.1). While (3.28) would still require a precise fit of the coefficients, addition of a fuzzy error analogously to (3.27) allows for some tolerance in the deviation of samples.

Kalman Filtering

Several trend detection methods based on forecasting have been developed and are extensively discussed in the literature, for example in [Allen 83, Challis 90, Avent 90]. Consistently it is stated that the most general method of trend detection based on forecasting was also one of the first: Kalman filtering.

Kalman filtering involves a recursive model of the process under observation and a measurement model. For example, equations defining these models for the case of a linear growth process are given by the measurement equation

$$y[t_n] = x[t_n] + e[t_n],$$

suggesting that measurement *y* depends on the current value (or state) *x* of the monitored process plus a normally distributed noise *e*, and by the process equations

$$x[t_n] = x[t_{n-1}] + b[t_n]$$
$$b[t_n] = b[t_{n-1}]$$

where *b*, the growth in *x* from t_{n-1} to t_n is defined to be constant, resulting in a linear development, periodic sampling provided (adapted from [Blom 85, Gordon 86]).

Initially, process variables such as *x*, *b* and *e* have to be estimated. Kalman filtering works by recursively predicting the next measurement and its prediction uncertainty (confidence interval) based on the above equations and the estimated process variables. Based on the difference of the prediction and the next measured value, process variable estimates are either updated to yield more precise predictions for the next sample, or, if the measured value lies outside the confidence interval, the current model is flagged invalid, meaning that a significant deviation from the trend has been detected.

Kalman filtering has been extended to allow aperiodic sampling (recall that assuming periodic sampling is generally unrealistic in clinical settings) [Gordon 86] and to monitor a process with several process models in parallel. The latter form, called multi-state Kalman filtering, is

particularly useful in clinical monitoring: rather than detecting that a particular (single) model has expired (a negative statement leaving its more interesting complement, which trend the signal rather follows, open), it matches the signal with an (ideally complete) set of expected changes in the data stream and determines which one describes the data best.

In essence, a multi-state Kalman filter involves a number of alternative models of the input data (trends!), a priori estimates of the likelihood of each model, an estimation of the variance of the input data, the filtering method itself, and a method of choosing the best model based on the derived a posteriori probabilities of each model [Sittig 92a]. A parallel implementation of multi-state Kalman filters using the process trellis [Factor 92] has been put forward in the ICM project [Sittig 90b, Sittig 92a].

It appears that Kalman filtering is particularly good at very early detection of a trend (or, rather, deviation from a trend), a property it has in common with more primitive forecasting methods such as cumulative sums [Allen 83, Avent 90]. Other trend detection methods such as the ones based on regression are certainly slower to react. However, because of the process model's recursive definition based on a limited past, the effectiveness of trend detection through Kalman filtering is restricted to domains where a trend manifests itself in very few samples, and not in overall developments involving long sequences of samples.

Also, estimation of the variance matrices seems to be a nontrivial problem. Clearly, there is no visual representation of allowable deviation that could be compared to the fuzzy courses, and, lacking respective knowledge acquisition procedures, multi-state Kalman filtering will involve a great deal of tuning. Unless standard process models for particular monitoring tasks can be established, widespread acceptance of Kalman filters in clinical monitoring will depend on the availability of suitable knowledge engineering tools.

Convincing examples of the performance of Kalman filters have been published, for example in [Blom 85, Gordon 86, Penczek 87, Sittig 90b, Sittig 92a]. Readers comfortable with excessive use of Greek matrices may refer to [Gordon 88] for a more competent treatment of Kalman filtering and its multi-state extension.

Qualitative comparison of the results of multi-state Kalman filtering published and trend detection based on short fuzzy courses presented in Section 5.2.1 suggests that their performance in trend detection is roughly equivalent. However, time lags of multi-state Kalman filters depend on the number of samples required to differentiate competing trends, while those of my approach depend on the duration of the trend, which may be longer.

Trends of Variable Length and Segmentation of Trends

In his experiments with detecting trends in the heart rate, [Sittig 92b] suggests a heuristics determining variable-length intervals over which linear regression yields a best fit (preferring longer intervals with slightly worse fit over shorter ones with better fit).

A simple extension of trend detection based on fuzzy courses allowing the detection of variable length trends has been suggested in [Steimann 94e]. There, a duration profile can be specified that lowers the degree of compatibility if only the beginning of a trend has been observed. The problem of variable length trends, however, is more general than that.

Several other approaches regard a trend as a sequence of (usually primitive) segments. The problem of trend detection is then decomposed into two subproblems:

1. the detection of trend primitives, and
2. the detection of transition from one primitive to the next.

Generally, unless segmentation can be tied to directly observable events (such as change of monotonicity), it adds a new combinatorial dimension to the problem of trend detection.

TRENDX [2] uses so-called trend templates to define and detect trends in sequences of samples. A trend template is a collection of temporal intervals each of which constrains a number of parameters (either through value constraints or by regression, see above). The temporal intervals can be of indeterminate length, the bounds are then related to other intervals or landmark points through temporal constraints.

Trends are detected in TRENDX by assigning samples to suitable intervals. For this purpose, TRENDX repeatedly generates and prunes hypotheses, instantiations of trend templates whose intervals are adapted to fit the samples' times and values.

An interesting approach to the converse problem, the planning of measurement sequences to detect a dynamic situation, is taken in [Nökel 89]. There, compatibility of measurement sequences with templates (called *situations*) is defined similar to the compatibility of sequences of samples with trends. However, a situation is made up of sequences of meeting intervals of constant parameter values, the intervals of different parameters being related by a subset of Allen's [Allen 84] temporal interval relations. The solution of the matching problem is therefore embedded in temporal logic (and actually implemented in a temporal PROLOG dialect) rather than mathematics and thus not particularly suited for continuous-valued domains such as the medical.

GUARDIAN plans to employ a real-time, on-line trend detection method based on fuzzy temporal pattern recognition (TFPR) [Drakopoulos 93]. TFPR performs a strictly sequential, non-optimal segmentation of trends based on local maxima of compound sigmoidal membership functions. However, sacrificing optimality for linear effort is a critical matter: accepting potential misclassification or even failure to classify (and thus detect!) a trend is hardly tolerable in intensive care. (The algorithm considers a segment switch for each new sample based on a comparison of its degree of membership in the current and the next segment. A temporarily higher membership in the next segment, however, may soon turn out to be a dead end, in which case continuation of the previous segment might have yielded a better result.) Nevertheless, an optimal segmentation algorithm (existent and presented by the same authors) is considered too expensive to be used in practice.

In conclusion, one may summarize that segmentation of trends, while an appealing theoretical issue, renders trend detection a highly consumptive task the practical necessity of which is not entirely plausible. For a counter-example refer to Chapter 6, where a single fuzzy course is shown to cover a wide spectrum of individual temporal developments.

Grammar-Based Trend Detection

Several authors employ grammatical representations for specifying and detecting sequences of events. Such sequences can also be regarded as trends.

State-based feature recognition (SBFR) classifies sequences of sub-trends with attributed finite state machines. The output of state machines can serve as input of others—state machines can be cascaded to form a hierarchy. Features forming the basis of trend detection (input of the lowest level of state machines) have to be obtained by suitable preprocessing of raw data. [Nelson 94]

[Barro 91] employs a context-free grammar to detect heart arrhythmia by analysing sequences of fuzzy beat labels. The fuzzy beat labels are derived in a fuzzy classification process described in [Barro 90]. By combining the membership degrees of the beat labels, sequence analysis (parsing) evaluates the "certainty" with which a given sequence represents a certain arrhythmia.

Generally, grammar-based methods can detect trends in variables whose values are discrete in nature (i.e., either observed-symbolic or sufficiently abstracted by symbols). They suffer from succession being the only temporal relation; all references to absolute time and duration

require extra treatment. Application of grammar-based approaches where internal states (the nonterminals of the grammar) carry significant meaning are discussed in Section 3.9

Summary

This section on trend detection presented more than a single trend detection method: it provided a theoretical framework for trend detection including a clear, simple definition of trend along with notions of occurrence of a trend in a signal and compatibility of a sequence of samples with a trend.

Most notably, compatibility is defined independent of any trend detection method and it applies to dense as well as sparse samples, no matter whether the trend is specified in absolute terms or via its first derivative. By contrast, several other trend detection methods rely on the availability of first and second derivatives of input data, not addressing how these can be derived from sparse (i.e., not reconstructable) samples (e.g., [Cheung 90, Konstantinov 92]).

Several published trend detection methods (or parts thereof) fall into this framework, yet the ones introduced in this section stand out due to their ease of implementation, computational efficiency, and, last but not least, naturalness of trend specification. Implemented as discrete-time signal processors, they fit smoothly into the monitoring framework of Chapter 2 and can therefore form integral parts of diagnostic monitors.

It remains to be noted that better results of trend detection via fuzzy courses might be expected from the use of compensating operators; however, the proof of Theorem 3.7 heavily relies on the use of min and max as fuzzy connectives, and the discussion of the respective properties of various compensating operators goes beyond the scope of this work.

3.8 Discrete-Valued Functions

Continuous arithmetic functions may be useful to compute derived variables from directly observed ones. Such functions are usually "memoryless", i.e., based on instantaneous and contemporaneous values, and range from simple calculations such as deriving the inspiratory to expiratory time (I:E) ratio from ventilator settings to more complex ones such as estimating the arterial oxygen partial pressure based on arterial oxygen saturation, blood pH, base excess, and body temperature.

While numeric analysis is a practical means for feature extraction and other low-level processing purposes, its contribution to data abstraction and higher-level reasoning is limited. Not only do natural functional dependencies tend to be hard to capture analytically, but means to operate on symbolic data are also necessary.

The following will present tables and rules, two popular constructs supporting objectified medical diagnosis, as systems that implement partial and piecewisely defined discrete-valued functions. These systems are memoryless and hence not suited to create temporal abstractions; Section 3.9 extends them with internal states that allow temporal developments to be taken into account.

3.8.1 Tables

Tables are a convenient notation for specifying discontinuous mappings of the form

$$f : D_1 \times \ldots \times D_k \rightarrow C_1 \times \ldots \times C_m$$

by enumerating tuples in a structured manner. Apart from being a means of approximating continuous mappings that are difficult to specify analytically, tables are particularly well-suited to map heterogeneous (numeric and symbolic) or purely symbolic data.

Tables can take different forms and are usually designed to achieve maximum readability. A very general, yet usually avoided layout of tables is shown in Figure 3.17; obviously, the rows of independent variables must be unique if the table is to present a function.

independent variables			dependent variables		
$x_1[t_n]$	···	$x_k[t_n]$	$y_1[t_n]$	···	$y_m[t_n]$
$d_{1,1}$	···	$d_{k,1}$	$c_{1,1}$	···	$c_{m,1}$
$d_{1,2}$	···	$d_{k,2}$	$c_{1,2}$	···	$c_{m,2}$
...	...	...	...	...	...

Figure 3.17: General form of a table

A table can be thought of as specifying a system the input and output of which correspond to the independent and dependent variables, respectively. The output is derived by taking the dependent variables' values from the row identified to match all input variables; the match is performed by testing for equality. If the mapping is partial, i.e., if there are constellations of input for which no entry in the table exists, the table produces the unknown value $\perp$.

To exploit regularities and to reduce the size of a table, its independent variables' entries are frequently filled with intervals or, more generally, sets, and the output is derived by testing set membership of the input. Within this monitoring framework, input to such a table is preprocessed by a discriminator, and the table is accessed with the derived sets, as exemplified in Figure 3.18. (The consequences of fuzzy discrimination are dealt with in Section 3.8.3.)

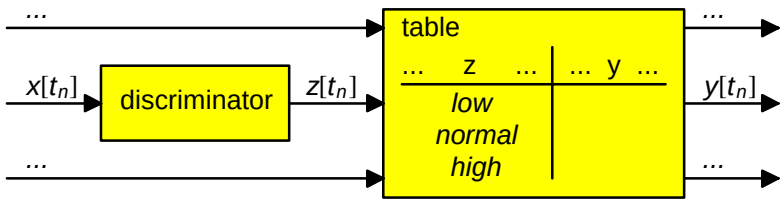


Figure 3.18: Numeric input to a table, transformed by a discriminator to symbolic values representing ranges

Mathematically, the table of Figure 3.17 represents a vector of piecewisely defined functions

$$\left(\begin{array}{l} y_1[t_n] = \left\{ \begin{array}{ll} c_{1,1} & \text{if} \quad x_1[t_n] = d_{1,1} \wedge \ldots \wedge x_k[t_n] = d_{k,1} \\ c_{1,2} & \text{if} \quad x_1[t_n] = d_{1,2} \wedge \ldots \wedge x_k[t_n] = d_{k,2} \\ \vdots & \vdots \end{array} \right. \\ \qquad\qquad \vdots \\ y_m[t_n] = \left\{ \begin{array}{ll} c_{m,1} & \text{if} \quad x_1[t_n] = d_{1,1} \wedge \ldots \wedge x_k[t_n] = d_{k,1} \\ c_{m,2} & \text{if} \quad x_1[t_n] = d_{1,2} \wedge \ldots \wedge x_k[t_n] = d_{k,2} \\ \vdots & \vdots \end{array} \right. \end{array} \right).$$

Note that if the constants of the table's output columns are replaced by variables, it implements a multiplexer.

Example

Some body signals such as the heart rate are monitored through several devices. While the ECG is considered the most reliable source, in case of ECG failure the heart rate derived from the pulse oximeter may be used as backup. This selection process can be implemented by the table of Figure 3.19.

◊

$ECGfailure[t_n]$	$heartRate[t_n]$
0	$ECGheartRate[t_n]$
1	$pulseHeartRate[t_n]$

Figure 3.19: Table implementing a multiplexer

3.8.2 Rules

Tables as defined above may be regarded as a special, regular form of rules: each row *i* of Figure 3.17 can be translated to

$$\textbf{if } x_1[t_n] = d_{1,i} \wedge \ldots \wedge x_k[t_n] = d_{k,i} \textbf{ then } y_1[t_n] = c_{1,i} \wedge \ldots \wedge y_m[t_n] = c_{m,i}\,.$$

More general rule formats as employed in rule-based medical expert systems leave the number and type of propositions and connectives in the conditional part of the rule open.[20] It should be noted, however, that within the monitoring framework of Chapter 2 every proposition must explicitly relate to a variable. More specifically, for a rule to be regarded as a system, its propositions must be of the form $x[t_n] = c$. Tests other than that for equality with a constant can again be shifted to a discriminator inserted before the rule or to systems dedicated to performing relational operations, producing degree-type output.

Generally, each rule specifies exactly one possible value for each variable *y* in its conclusion. In order to form a system (with that variable as its output), all rules affecting the value of one variable are grouped and encapsulated in a system which is then specified as

[20] Note again that measures of uncertainty of observations (propositions) and implications that can be found in rule-based systems are generally not addressed in this work.

$$y[t_n] = \begin{cases} c_1 & \text{if } < cond_1 > \\ \vdots & \vdots \\ c_m & \text{if } < cond_m > \end{cases}$$

which gives rules the appearance of piecewisely defined functions. Compared to tables, the difference is that the conditional part is less structured and may indeed be more complex. Again, if c_1 through c_m are replaced by variables, the system acts as a multiplexer.

Note how regarding rules as systems gracefully solves three problems inherent to conventional rule-based systems: firstly, an obvious, hierarchical structure is imposed on the rulseset, secondly, computational complexity is reduced (no maintenance of and lookup in a fact base is required), and thirdly, the problem of inconsistency is locally restricted to the scope of a system.

Also note that a monitoring framework allowing the integration of systems of different origin is in fact more general than any purely rule-based approach: although rules can be used, they are certainly not the only means to represent medical knowledge. Consequently, one should always check before designing a rule-base whether tables or other systems are better suited to do the job. During the design process, the interpretation of rules as partial or piecewisely defined functions provides a neutral perspective to decide when rules are appropriate.

In a typical monitor, rules are employed on higher levels of abstraction where information is already in a symbolic form. The next section will show that employment of "memoryless" rules of the above kind (disregarding time) is insufficient when data is to be reduced to obtain stable temporal abstractions. Just like extending discrimination by a temporal dimension results in trend detection, adding memory to tables and rules leads to finite state machines.

3.8.3 Extension to Fuzzy Arguments

Tables and rules as defined above present exact knowledge (or, rather, an exact version of the knowledge) of the problem domain. With this approach, notational precision is traded for continuity and thus, because most physiological dependencies are continuous, for naturalness; yet, symbolic functions are more readily specified than continuous functions taking possible graduation into account.

Fortunately, the extension principle (see Section 1.2.1) allows the introduction of continuity at no cost at all: feeding the extended system (based on the same specification) with fuzzy (and hence, graduated) input yields fuzzy output.

Example

Consider a discrete-valued system specifying some tolerance level of a patient depending on his age as in

$$tolerance[t_n] = \begin{cases} high & \text{if } age[t_n] = young \\ low & \text{if } age[t_n] = old \end{cases}$$

which is obviously very crude. However, when presented with the input

$$age[t_n] = \{(young, 0.3), (old, 0.7)\},$$

its fuzzified version yields

$$tolerance[t_n] = \{(high, 0.3), (low, 0.7)\},$$

which may be interpreted as some intermediate value closer to *low* than to *high* (compare Sections 2.2.2 and 3.6.2)

◊

Note that if *high*, *low*, *young* and *old* are themselves fuzzy sets, the fuzzy input and output of the above example are level 2 fuzzy sets. This is of particular interest in connection with fuzzy

discriminators and trend detection: while tables and rules allow the specification of functional dependencies in a simplified, discrete form, fuzzification somehow reverses the discretization process necessary to make them applicable to continuous findings. In certain cases, the output can even be defuzzified to continuous values. It must be borne in mind, however, that a fuzzified discrete system is in most cases merely a pragmatic approximation of the modelled continuous dependency. While it seems that the approximation error is often negligible in feedback control systems (this explains the success of fuzzy control), the same cannot generally be claimed for diagnostic systems.

Note that fuzzification of discrete-valued functions gracefully incorporates the propagation of uncertainty in rules: if an output depends on many input values (a multivariate function or a rule with multiple premises joined with logical and), the extension principle implies that the minimum is propagated to the output. It does, however, not cover uncertainty imposed by the inference itself: statements in the MYCIN [Shortliffe 76] style

if *premise* **then** *conclusion*, **CF** = *cf*

or CADIAG-2 [Adlassnig 86] style

if *antecedent* **then** *consequent* **with** (*O*,*C*)

are not provided for.

3.9 States and State Transitions

From an observational standpoint, intelligent behaviour can be characterized by a system's varying response to identical input. Yet, intelligence is to be distinguished from indeterminacy—an intelligent agent can always explain its decisions, only that the decision criteria and path can become arbitrarily complex and may be difficult to follow. In particular, the explanation of current behaviour will involve observations made in the past, which implies that intelligent systems need to have memory.

History-sensitive, that is, window-based, causal systems, have memory; they derive their output based on a limited past or history of their input. Because in digital systems (discrete-time systems with digitized variables) the number of possible constellations in the input window is finite, the current past can be subsumed by a finite domain *state* variable on whose value the transformation of the current input to the current output depends.

The behaviour of such systems can be modelled by what is known in automata theory as a *transducer*. Although definitions vary slightly from author to author (e.g., [Dougherty 88, Aho 86]), it can be identified that a transducer consists of

Q,	a nonempty finite set of states,
$q_0 \in Q$,	an initial state,
I,	a nonempty set of input symbols,
O,	a nonempty set of output symbols,
$\delta : Q \times I \to Q$,	a state transition function, and
$\varepsilon : Q \times I \to O$,	an output function.

If δ and ε are deterministic functions, the transducer is also labelled *deterministic*. The set of final states that can be found in other definitions is omitted here, as monitoring is an iterative process that is terminated on exhaustion of input data rather than on arrival at some final state.

The state transition function δ usually maps the current state and input to the next state. Within the monitoring framework of Section 2 I decided to use a slightly different definition of the form

$$q[t_n] = \delta(q[t_{n-1}], i[t_n]), \tag{3.29}$$

where q is the variable that holds the current state of the transducer at time t_n.

It should be noted that both input and output of a transducer can be tuples of arbitrary (heterogeneous) type, and the term *symbol* in the definition of transducers does not exclude numbers from being input or output as long as their domains are finite.

Remembering the previous section, transducers can be specified and interpreted as rules of the form

$$\textbf{if } q[t_{n-1}] = q \wedge x[t_n] = i \textbf{ then } \; q[t_n] = q' \wedge y[t_n] = o$$

as well as as tables of the form shown in Figure 3.20, which both integrate memory in the form of q.

$q[t_{n-1}]$	$x[t_n]$	$q[t_n]$	$y[t_n]$
q_1	i_1	q'_1	$o_{1,1}$
q_2	i_1	q'_2	$o_{1,2}$
...	...	...	...
q_1	i_2	q''_1	$o_{2,1}$
...	...	...	...

Figure 3.20: Table specifying a transducer

Note that for the transducer to be deterministic, each constellation of input variables must cause only one transition.

More easily perceivable than textual or tabular definitions of transducers are state transition diagrams. As suggested by their name, their emphasis is on the states and transitions between states. States are depicted by circular nodes and transitions by directed arcs pointing from the source to the target state. The initial state is usually indicated by an ingoing arc without a source state. Input and output are placed alongside the transitions and can be omitted if only the possible transitions interest (and not the conditions on which they take place). Figure 3.21 presents a state transition diagram whose transitions are labelled with input only.

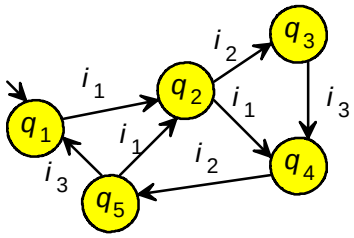


Figure 3.21: State transition diagram

General history-sensitive systems are not reasonably implemented in the form of transducers: for example, the specification of moving averages or median filters using a transducer would require an enormous number of states and is extremely inefficient. Things are different, however, when the number of states is small, and every state is given a particular, distinguishable meaning within the scope of the monitoring problem. Such is the case when modelling natural disease histories.

3.9.1 Modelling Natural Disease Histories

During the course of a disease, the patient typically moves through a sequence of different characteristic states. His current state can be encoded in the form of a symbolic variable. A state transition diagram can then be used as a simple model of the disease by specifying

possible sequences of states corresponding to natural disease histories. State transitions are augmented by conditions, events that trigger transition from one state to the next.

Example

Cardiovascular monitoring is confronted with the problem of differentiating multiple diseases all resulting in a shared set of symptoms. However, each disease has its own scenario, i.e. its own, characteristic sequence of findings, so that differentiation can be based on sequence analysis. Accordingly, hypovolemia, a scenario depicted in Figure 3.22 a), can be distinguished from, say, sepsis, which involves similar states, yet in a different sequence. [Cohn 90]

◊

Example

Disorders in the acid-base status of a patient also take typical courses. Respiratory acidosis for example may be caused by CO_2 retention due to insufficient respiratory effort or diseases of the lung tissue. In its progression, increased serum bicarbonate concentration as a result of renal compensation may lead to a normalized blood pH, a state that is called *compensated respiratory acidosis*. Alternatively, ventilator therapy can neutralize a respiratory acidosis, leading to a state called *corrected respiratory acidosis*. In case of aggressive ventilator therapy, blood pH may further increase, resulting in a state called *overcorrected respiratory acidosis*. A respective state transition diagram is shown in Figure 3.22 b). [Uckun 92a, Uckun 93a]

◊

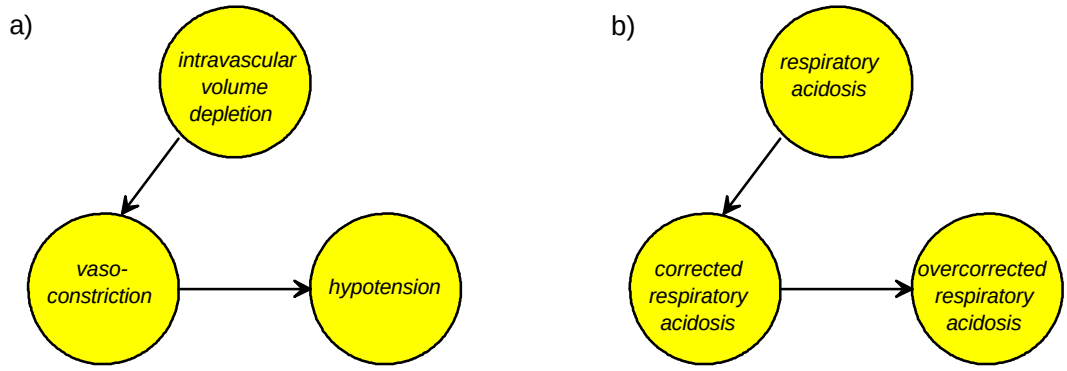


Figure 3.22: State transition diagrams of natural disease histories
a) hypovolemia
b) respiratory acidosis treated with ventilator therapy

Both examples suggest that the current state of the transducer can be regarded as its output. The output function ε characteristic of transducers is therefore omitted throughout the following, and the remainder is addressed by the more general terms *automaton* and *finite state machine*, which are used as synonyms here.

Practical applicability of automata in clinical monitoring is hindered by the following dilemma: if an automaton consists of only few states and transitions, it is easy to handle, yet a poor model of the natural disease history, as abrupt transitions from one state to another (denoting an entirely different condition of the patient) rarely occur; if the automaton has many states to model finer nuances of a disease and its development, the resulting number of transitions required to model all possible changes in state soon becomes so vast that it escapes manageability.

Just as before, fuzzification is a promising remedy: while maintaining an original small number of significant states and transitions modelling the natural disease history on a rather abstract

level, fuzziness allowing for gradual state transitions can blow up the number of states the automaton can take on without adding complexity.

3.9.2 Fuzzy Automata

Considering the fact that systems such as fuzzy discriminators and trend detectors yield fuzzy output, the extension of state machines to process fuzzy input seems a natural consequence. Applying the extension principle from Section 1.2.1 to (3.29) yields

$$\tilde{q}[t_n] = \tilde{\delta}\left(\tilde{q}[t_{n-1}], \tilde{i}\,[t_n]\right),$$

where

$$\mu_{\tilde{q}[t_n]}(q) = \begin{cases} 0 & \text{if } \neg\exists q', i : q = \delta(q', i) \\ \max\limits_{q', i: q=\delta(q', i)} \min\left(\mu_{\tilde{q}[t_{n-1}]}(q'), \mu_{\tilde{i}\,[t_n]}(i)\right) & \text{else} \end{cases} \tag{3.30}$$

Unfortunately, application of (3.30) is not as unproblematic as it seems, as the following will demonstrate.

The Problem of Membership Depletion

The extension of the transition function, (3.30) implies that full set membership of any one state reflected in

$$\mathrm{hgt}\,(\tilde{q}[t_n]) = 1, \tag{3.31}$$

once lost, cannot be regained. More specifically, from (3.30) it follows that

$$\mathrm{hgt}\,(\tilde{q}[t_n]) \le \mathrm{hgt}\,(\tilde{q}[t_{n-1}]) \;\wedge\; \mathrm{hgt}\,(\tilde{q}[t_n]) \le \mathrm{hgt}(\tilde{i}\,[t_n]) \quad \text{for all } t_n,$$

which means that $\langle \mathrm{hgt}\,(\tilde{q}[t_n])\rangle$ is a monotonically decreasing sequence.

Example

Consider the automaton of Figure 3.23 a) which may be considered to be a part of a more complex one, particularly as no initial state is specified. Given a current state

$$\tilde{q}[t_n] = \{(q_1, 0), (q_2, 1), (q_3, 0)\}$$

and next input

$$\tilde{i}\,[t_{n+1}] = \{(i_1, 1), (i_2, 0.1)\}\,,$$

the next state is derived as

$$\tilde{q}[t_{n+1}] = \{(q_1, 0), (q_2, 0), (q_3, 0.1)\}\,.$$

◊

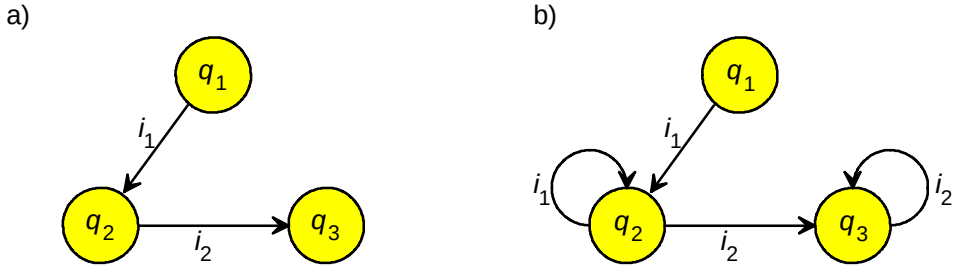


Figure 3.23: Simple automaton serving to explain the problem of membership depletion

Clearly, this is not an acceptable property of an automaton modelling a natural disease history, as any model that cannot assign a specific state to the patient is obviously poorly designed. In particular, note that the condition i_1 that initially led to the state q_2 still fully holds, so that

there is no semantic justification that its degree of membership should drop to zero on the next transition. A first remedy is an extension in the definition of the automaton that induces a "peak hold" [Steimann 94a]. Briefly, if it is guaranteed that every state has a transition onto itself on every input that leads to that state, i.e., if

$$\delta(q', i) = q \;\rightarrow\; \delta(q, i) = q \quad \text{for all } q, q' \in Q, i \in I, \tag{3.32}$$

the membership of q_2 in the current state of the automaton remains high as long as the membership of i_1 remains high.

Example

Revisiting the previous example, check that employment of the automaton of Figure 3.23 b) satisfies (3.32) and, given the same current state and input, yields

$$\tilde{q}[t_{n+1}] = \{(q_1, 0), (q_2, 1), (q_3, 0.1)\}.$$

◊

A positive side effect of (3.32) is that the automaton does not oscillate (i.e. change state in a periodic fashion) when repeatedly fed with constant input, a property that would clearly not be acceptable in the clinical context, as stable (i.e. not changing) input should be reflected in stable output. Instead, the following theorem holds:

Theorem 3.8 (stability)
The current state $\tilde{q}[t_n]$ of a fuzzy automaton with peak hold always becomes stable after a finite number of repeats of the same input $\tilde{i}$.
■

Proof
The theorem is proved by showing that there is a step *r* such that

$$\tilde{q}[t_n] \subseteq \tilde{q}[t_{n+1}] \subseteq \ldots \subseteq \tilde{q}[t_{n+r}] = \tilde{q}[t_{n+r+1}] = \ldots, \tag{3.33}$$

where the fuzzy set inclusion of two fuzzy subsets $\tilde{A}, \tilde{B}$ of X is defined as

$$\tilde{A} \subseteq \tilde{B} \quad \Leftrightarrow \quad \mu_{\tilde{A}}(x) \le \mu_{\tilde{B}}(x) \quad \text{for all } x \in X,$$

in two steps:

1.) $< \tilde{q}[t_n] >$ is increasing, i.e., $\tilde{q}[t_n] \subseteq \tilde{q}[t_{n+1}] \subseteq \ldots$
2.) $\exists r : \tilde{\delta}(\tilde{q}[t_{n+r}], \tilde{i}) = \tilde{q}[t_{n+r}]$. For all subsequent states (3.33) follows from $\tilde{\delta}$ being a function.

1.) For every fuzzy state $\tilde{q}[t_n]$ following the initial state $\tilde{q}[t_0]$ and every constant input $\tilde{i}$ (3.30) implies:

For every state q and sampling time t_n there is a transition that determines q's membership value in $\tilde{q}[t_n]$, i.e.

$$\forall q, \delta^{-1}(q) \neq \varnothing : \exists q', i : \delta(q', i) = q \wedge \mu_{\tilde{q}[t_n]}(q) = \min(\mu_{\tilde{q}[t_{n-1}]}(q'), \mu_{\tilde{i}}(i)).$$

Thus, $\mu_{\tilde{q}[t_n]}(q) \le \mu_{\tilde{i}}(i)$ and following $\min(\mu_{\tilde{q}[t_n]}(q), \mu_{\tilde{i}}(i)) = \mu_{\tilde{q}[t_n]}(q)$. Repeated input of $\tilde{i}$ and $\delta(q, i) = q$ then implies

$$\mu_{\tilde{q}[t_{n+1}]}(q) = \max(\mu_{\tilde{q}[t_n]}(q), \max_{\delta(q', i) = q} \min(\mu_{\tilde{q}[t_n]}(q'), \mu_{\tilde{i}}(i)))$$

$$\mu_{\tilde{q}[t_{n+1}]}(q) \ge \mu_{\tilde{q}[t_n]}(q),$$

which is justifying the term *peak hold,* and consequently

$$\tilde{q}[t_{n+1}] \supseteq \tilde{q}[t_n].$$

2.) (indirect) $\tilde{i}$ being constant and (3.30) imply that there is no infinite sequence of fuzzy states $< \tilde{q}[t_n] >$ such that $\tilde{q}[t_{n+1}] = \tilde{\delta}(\tilde{q}[t_n], \tilde{i}) \wedge \tilde{q}[t_{n+1}] \supseteq \tilde{q}[t_n]$. Therefore, there must be a

step r after which $\tilde{q}[t_{n+r}] \subseteq \tilde{q}[t_{n+r-1}]$. Because $< \tilde{q}[t_n] >$ is increasing, $\tilde{q}[t_{n+r}]$ must equal $\tilde{q}[t_{n+r-1}]$.

■

The response of the automaton of Figure 3.23 b) with "peak hold" to continuously decreasing membership of i_1 and concurrently increasing membership of i_2 is shown in Table 3.2. Note how the membership of q_2 to follows i_1's while q_3 picks up the membership of q_2. However, the rise of q_3 is limited by the crossover point where q_2 falls below q_3. Eventually, this means that, given the input of Table 3.2, q_3 can never take over full membership, suggesting that the patient never fully makes the transition to q_3. Considering the given input, this is somewhat counterintuitive and still not the behaviour expected of an adequate model, as it eventually leads to membership depletion.

A simple solution is to normalize $\tilde{q}[t_n]$ after each transition, i.e., to adjust its membership function so that (3.31) holds. Two straightforward normalization rules are

$$\mu_{\tilde{q}_{norm}[t_n]}(q) = \frac{\mu_{\tilde{q}[t_n]}(q)}{\mathrm{hgt}\,(\tilde{q}[t_n])} \tag{3.34}$$

and

$$\mu_{\tilde{q}_{norm}[t_n]}(q) = \left\{ \begin{array}{ll} 1 & \text{if } \mu_{\tilde{q}[t_n]}(q) = \mathrm{hgt}\,(\tilde{q}[t_n]) \\ \mu_{\tilde{q}[t_n]}(q) & \text{else} \end{array} \right\} \text{for all } q \in Q. \tag{3.35}$$

Another possibility is to operate the fuzzy automaton and its crisp counterpart in parallel and to define

$$\mu_{\tilde{q}_{norm}[t_n]}(q) = \left\{ \begin{array}{ll} 1 & \text{if } q = q[t_n] \\ \mu_{\tilde{q}[t_n]}(q) & \text{else} \end{array} \right\} \text{for all } q \in Q \tag{3.36}$$

where $q[t_n]$ is the current state of the crisp automaton. (The crisp automaton transitions only on $\mu_{\tilde{i}\,[t_n]}(i) = 1$). This latter normalization, which is functionally equivalent to the introduction of active states as suggested in [Steimann 94a], has been used in the experiments conducted in [Steimann 94c] and Chapter 5.

Table 3.2: Gradual transition of the automaton of Figure 3.23 b) from q_2 to q_3, triggered by the input sequence $\langle \tilde{i}\,[t_n] \rangle$; representation is a continuous-time approximation of a dense discrete-time sampling sequence.

INPUT	STATES			
	no normalization	normalization (3.34)	normalization (3.35)	normalization (3.36)
$\mu_{\tilde{i}\,[t_n]}(i_1)$	$\mu_{\tilde{q}[t_n]}(q_2)$			
$\mu_{\tilde{i}\,[t_n]}(i_2)$	$\mu_{\tilde{q}[t_n]}(q_3)$			

Finally, membership depletion can be counteracted by adjusting the state membership function after each transition so that

$$\sum_{q \in Q} \mu_{\tilde{q}[t_n]}(q) = 1, \tag{3.37}$$

meaning that a constant "amount of membership" is distributed over all states.

Interpretation of the Fuzzy State

The extension principle is mathematically motivated—its application to automata does not guarantee a meaningful extension of the model of a natural disease history. An interpretation of the fuzzy current state therefore has to be sought.

Combining the characteristics of crisp and fuzzy automata, the automaton should not be in more than one state at one time, and it should allow for gradual transition and consequently for transitional states, i.e. the being between states. This suggests that the fuzzy state is interpreted as a position in the state space. Deviating from fuzzy monotonic discrimination, however, the space is not one-dimensional—rather, each state has its own dimension, as shown in Figure 3.24 a).

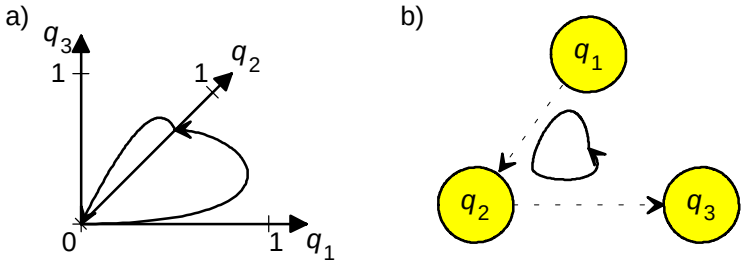


Figure 3.24: Interpretation of a fuzzy state
a) trace of position in an *n*-dimensional state space
b) two-dimensional model based on variable attraction of states ("gravity")

Transferred to the example of Figure 3.22 b), Figure 3.24 a) could be interpreted as a moderate respiratory acidosis that is being corrected and slightly overcorrected before the acid-base balance goes back to normal (a state not included in the diagram).

Alternatively, degrees of membership could be interpreted as attracting the fuzzy current state to the positions of the crisp states in the two-dimensional space of the state transition diagram, as suggested in Figure 3.24 b). This representation relies on a meaningful arrangement of states in space so that any current position can be assigned an unequivocal interpretation. Note that for this latter representation (3.37) is particularly well-suited.

3.9.3 Stabilizing Effects

All previously presented systems tend to propagate changes in their input to their output. Although data smoothing techniques, discriminators, and trend detectors can reduce the energy in the output, their responsiveness to small changes in input is still too high to result in acceptably stable temporal abstractions. Ideally, the output of a monitor changes only if something truly meaningful has occurred. If it is to indicate a slowly progressing change of state, then this change should be monotonic. If the transition from one state to another is at all reversible, then this reversal can be triggered by different findings. Inertia introduced by automata leads to reduction of changes and adds to achieving stable temporal abstractions.

Normalization contributes its share to the fact that once an automaton has made the (fuzzy) transition to a state, it remains in that state until enough evidence has been collected for it to change to a successor state. This inertia reflects a natural decision making behaviour: once a decision has been made, it is usually pursued rather uncritically until enough evidence has been gathered to arrive at a new conclusion. In particular, a diagnosis is not retracted as soon as its symptoms begin to vanish; rather, a significantly improved condition is to be seen before the therapy is ended and the patient is released.

If properly designed, fuzzy automata are practical tools to achieve stability (monotonicity and constancy) in the output of a monitor. This is impressively supported by the experiments presented in Chapter 5.

3.9.4 Design Considerations

Earlier experiments I conducted with fuzzy automata were implemented largely independent of the monitoring framework and modelled transitions between rather heterogeneous states [Steimann 94c]. In their current, integrated employment, the design of automata is influenced by the fact that all information is tied to variables each reflecting a certain property of the process under observation. Consequently, the output of an automaton is a value of a (fuzzy) symbolic variable whose base set is a set of semantically related symbols—failing to comply with this requirement will result in an ill-designed monitor the output of which can hardly be further processed.

On the other hand, there are lower limits for the number of states an automaton must have. The current state of an automaton is understood to be a sufficiently differentiated abstraction of the history the automaton has encountered. An automaton must therefore have at least as many states as there are cases to differentiate.

Generally, an automaton specifies a symbolic variable by defining possible sequences in which it can take on its values together with the conditions that cause a change of value. This is in contrast to memoryless, monotonic (fuzzy) discriminators, whose output sequences remain unspecified (depend solely on the sequence of input) and whose ordering of symbols is implied by the ordering on its underlying base set.

Automata's advance in expressive power over discriminators is rooted in the fact that their interpretation of current input is history-dependent. Two identical snapshots in time can therefore be interpreted differently, which is a major step towards intelligent behaviour.

3.9.5 Automata in Other Monitoring Projects

VM corroborates its pioneering role by employing a finite state machine to specify possible changes in therapy, namely the change of ventilation modes. However, this employment (generating sequences) is contrary to that in tracking disease histories as presented above (accepting sequences).

Direct diagnostic use of automata has been made in DYNASCENE [Cohn 90] and ICM [Factor 89, 92]. Both systems detect clinically meaningful constellations—called scenes—and interpret the sequence of their occurrence. However, they both, like all other approaches to disease tracking, lack concepts of smooth progression or partial illness as introduced by fuzzy automata.

SBFR [Nelson 94] employs finite state machines for trend detection; see Section 3.7.9 for a discussion. Regular grammars are occasionally used to classify temporal patterns that can be coded as a sequence of symbols (representing discrete events), for example in [Barro 91, Juhola 91]. A two-level approach to real-time diagnosis and control of fermentation processes is taken in [Konstantinov 91]; the upper level, basically a finite state machine, traces the stage of the fermentation process and so determines the choice of applied control strategies. The transititon between stages is triggered by the detection of trends via temporal shape analysis [Konstantinov 92].

Automata undoubtedly contribute to monitoring temporal developments on a symbolic level. However, the temporal component coded in the transitions is very limited: not only can no explicit reference to past values be made, but also the only temporal relation is that of subsequence. YAQ [Uckun 92a] overcomes this deficiency by linking state transitions to explicit history lookups which can query additional temporal information.

Chapter 4

The Design and Implementation of DIAMON-1

DIAMON-1 is not a ready made system, but an implementation of the monitoring framework of Chapter 2 supplemented by a collection of elementary systems presented in Chapter 3. Figuratively speaking, DIAMON-1 provides the building blocks required to construct an individual monitor.

As pointed out in [Meyer 88], design and implementation are basically the same activities, only on different levels of abstraction. Sympathizing with this perspective, I will not treat design and implementation separately, but rather try to provide a general overview of the design of DIAMON-1 and go into detail wherever deemed necessary. But first, I will briefly sketch the development environment.

4.1 Development Environment

Design and implementation are creative processes that are influenced by environmental factors as much as by the problem itself. Available resources such as personnel and software tools contribute to feasibility considerations, precluding some design decisions while favouring others.

In this particular case, the development team consisted of myself only, and the goal was to develop a working prototype that could serve as a proof of concepts rather than to produce a fully-fledged diagnostic monitor ready for routine use. This allowed a rather casual software development process: team conferences were as superfluous as sophisticated version control systems enabling parallel development of interdependent parts of the system.

State-of-the-art software development offers a wide palette of different ontologies to choose from. To mention just a few mainstream ones: logic programming, which is particularly popular for AI applications, competes with functional approaches and, more recently, with object-oriented programming.

The choice may seem arbitrary, yet has its implications: a logic-based approach typically leads to a different solution than a functional approach, and, because requirements are never complete, it also has different properties.

Frankly, the design of DIAMON-1 is object-oriented. As will be seen, this choice is natural, even though the functional[21] specification of variables and systems of the previous chapters suggests a functional design.

4.1.1 The Object-Oriented Paradigm

> *"Born in the ice-blue waters of the festooned Norwegian coast; amplified (by an aberration of world currents, for which marine geographers have yet to find a suitable explanation) along the much grayer range of the Californian Pacific; viewed by some as a typhoon, by some as a tsunami, and by some as a storm in a teacup—a tidal wave is reaching the shores of the computing world."* [Meyer 88]

Undoubtedly, *object-oriented* is one of the buzzwords of contemporary software engineering. However, it is more than a jargon: object-orientedness is a paradigm that pervades all aspects of software development including analysis, design, implementation, and maintenance.

Yet, like many buzzwords, *object-oriented* lacks a generally agreed upon definition. The following exposition therefore invariably has a personal bearing.

Briefly,

object-oriented = abstract data types + inheritance.

This "equation" may appear oversimplified, yet stands up to most objections, and it is indeed the best I have encountered. Nevertheless, it requires some explanation.

In the object-oriented paradigm, the role of abstract data types is taken over by *classes*. Semantically, a class denotes a set of objects characterized by a number of common properties. Properties are in particular functions, the ability of an object to fulfil certain tasks or to perform certain actions. An *instance* is an instantiation of a class and corresponds to an object, to an element of the set the class denotes.

Classes and their interrelations make up the static structure of an object-oriented design and its implementation, whereas instances are the entities found in a running program. Both classes and instances are often comprised under the general term *object*—this explains why the paradigm is called object-oriented.

Just as sets can have subsets, classes have subclasses. Semantically, a subclass denotes a subset of the set its superclass denotes. It is thus only natural that properties (including functions) specified for a class (meant to characterize its instances) also hold for its subclasses—properties are propagated from superclasses to subclasses. This propagation is called *inheritance*.

Inheritance promotes subclassing, practically to save code, but as a side-effect to organize the classes in a class hierarchy, in which more general classes are superclasses, more specialized classes are subclasses, and classes sharing common functionality are arranged as siblings. Because it designates the paths of inheritance, the class hierarchy is also called *inheritance hierarchy*.

Inheritance not only imposes its hierarchy on the static (class) structure of a design and its implementation, it also extends its expressiveness: wherever an instance of any one class is required, instances of all of its subclasses can take its place. This is again implied by the set-subset interpretation of the class-subclass relation: a role that can be played by any element

[21] here meaning "based on functions"

of a set can also be played by any element of all of its subsets. This is what is occasionally called the *polymorphism* of object-orientedness.

However, the main advantage of the object-oriented paradigm over functional design and implementation lies in the naturalness of the approach. Instances model objects that can be identified in the problem domain, and classes specify meaningful categories of objects by listing their properties that are deemed important in the given problem. A solution of a problem is then computed by simulating the processes that are or should be going on in the real world.

4.1.2 An Object-Oriented Perspective of Monitoring

Transferred to the diagnostic monitoring problem, the object-oriented paradigm leads to the following view. Objects that can immediately be identified in the problem domain are the patient, the monitor, and the kinds of systems presented in Chapter 3. Properties of the patient are the variables that can be observed of him; properties of systems are their input and output variables, a possible window-width, and a characterizing function to perform. Closer examination reveals that there must be input and storage devices for sequences of samples, output facilities, a clock, and so on. Once all relevant objects have been identified, the classes and their interrelations can be specified. The result of doing so in the case of DIAMON-1 is presented below.

Quite to the contrary, with a problem as general as that of diagnostic monitoring, the functional decomposition approach runs into difficulties very soon, as it heavily depends on the functionality to be decomposed, which varies from (monitoring) case to case.

4.1.3 Chosen Environment

Current software development environments are more or less loose collections of tools intended to support and facilitate the software development process. Development environments usually favour a target programming language, or, at least, a category of such languages. Consequently, there are environments for logic programming, for functional, and for object-oriented programming.

Among the purely object-oriented (target) programming languages are SMALLTALK [Goldberg 83] and EIFFEL [Meyer 88]; one that has adopted object-oriented characteristics (among many other features) is C++ [Stroustrup 86].

SMALLTALK has the advantage that it is a programming language and development environment at the same time. It is almost completely self-contained and utilizes only very primitive operating system services. SMALLTALK makes its own source code accessible and modifiable: it is its own library, compiler, editor, etc.

A program in SMALLTALK is always an extension of the existing system; rather than developing a stand-alone program, new code is integrated in the environment. Working with SMALLTALK is highly interactive: by inspecting the system, the developer can learn how things are done and continuously improve his own SMALLTALK programming style.

After all this praise it is quite clear that I chose SMALLTALK as the development environment. SMALLTALK's major deficiencies, its lack of type checking and of exception handling, are acceptable in its employment as a prototype development environment and more than outweighed by its homogeneity and ease of use.

4.2 Framework Classes and the Evaluation Cycle

Every monitor constructed with DIAMON-1 consists of three main components: a *data source*, a *blackboard*, and a *system hierarchy*. The monitor itself, the data source, and the blackboard

are specified by respective classes; the system hierarchy is organized as a collection of systems ordered in the sequence of evaluation.

4.2.1 The Class *Monitor*

Every monitor constructed with DIAMON-1 is an instance of class *Monitor*. As suggested by Figure 4.1, a monitor maintains a blackboard and a system hierarchy. Each monitor is also linked to a source interfacing its input devices.

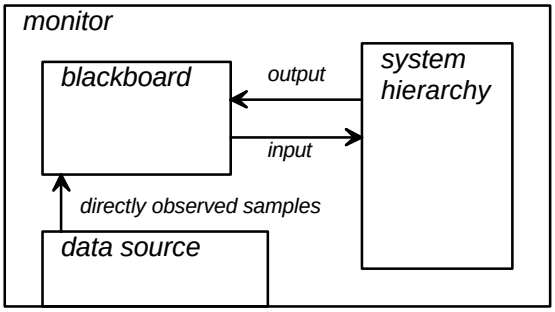


Figure 4.1: Components and information flow of a monitor

The monitor rules the information and control flow between the data source, the blackboard, and the systems of the system hierarchy. It is responsible for synchronization and therefore maintains an internal clock holding the current sampling time.

A monitor operates in evaluation cycles. Depending on its synchronization mode, a new evaluation cycle is either triggered by the availability of new data from the source (data-driven) or by the internal clock, polling the source at regular intervals (clock-driven).

In its off-line employment, the source is a database in which all sequences of samples are already recorded. Data is accessed from the database in chronologically ordered packages of samples with identical sampling time. Evaluation is triggered every time a new package is read, and the monitor's internal clock is set to the sampling time of the package. Evaluation is thus, albeit simulated, data-driven.

4.2.2 The *Source* Classes

All source classes provide a common access protocol for reading the next sample from the source. A sample is represented by a quadruple specifying the patient, the sampling time, the label of the variable, and its value.

Currently, only classes that read database sources are implemented. Source classes interfacing on-line monitoring devices would have to integrate different device drivers and user dialog facilities to acquire supplementary information entered manually.

Three types of databases can currently be accessed: plain ASCII tables, dbase III format database files, and files in a self-specified medical record format called MRF. The former two are tuple-oriented, i.e., each row is associated with one sampling time (specified in two special columns *date* and *time*) and one patient (named in the special row *patient*). The proper variable label can be assigned to each column by using a map file associating database column headers (which are usually syntactically constrained) with the monitor's variables' names (which may be arbitrary strings).

The MRF is based on single samples. Briefly, it is organized as follows. Each entry is of the form

<*label*> **of** <*patient*> **at** <*time*> **=** <*value*>

To reduce the required space, two standard variables *patient* and *time* are available that can be set by

patient = <*patient*>

and

time = <*time*>,

respectively. A reduced entry of the form

<*label*> = <*value*>

then assumes the current values of *patient* and *time* for the omitted identifiers.

4.2.3 The Class *Blackboard*

Once read from the source, the data is written on the blackboard. Each variable has its own row on the blackboard which is labelled with the variable's name. Vertically, the blackboard is divided into two sections: one contains the current values of all variables (the *now* section), and the other contains the variables' individual histories (the *history* section, see Figure 4.2). Individual means that the history of each variable is based on its own sampling sequence of which all unknown values have been removed—this is in contrast to the *now* section of the blackboard, where missing values are explicitly entered as *unknown*.

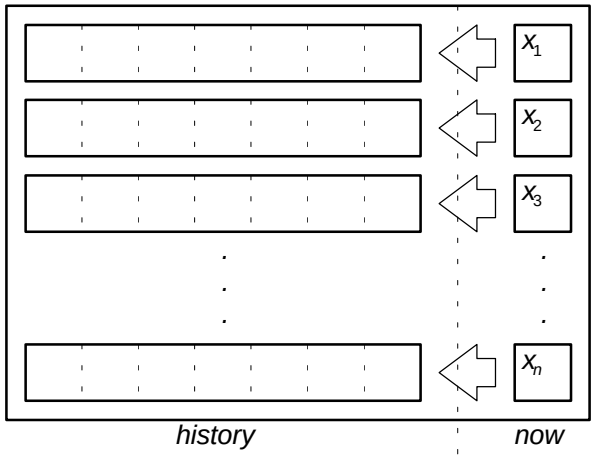


Figure 4.2: The blackboard of a DIAMON-1 monitor

The history section of the blackboard provides shared memory for all systems deriving their output from a limited past of their input variables. Access is restricted to read only for the *history* section. The blackboard has no knowledge of the extent of the history each system requires—all systems accessing the common memory must release the entries they do not need anymore.

At the beginning of each evaluation cycle, the blackboard is "wiped". "Wiping" in this case means that all values (excluding the *unknown* entries) from the *now* section are transferred to the *history* section of the blackboard. After wiping, new data can be written in the *now* section of the respective variables; variables for which no new value is available are marked as *unknown*.

4.2.4 The Class *System*

The class *System* is an abstract class that specifies the features provided by all elementary systems of DIAMON-1. It attributes all instances with two collections holding input and output variables, respectively, and a display facility capable of presenting the system's output.

All systems must respond to the request for evaluation, which makes them inspect the blackboard, derive a new output, write it on the blackboard, and update their internal state regarding the current time of the monitor. Window-based systems also release the entries from the *history* section of the blackboard that have dropped out of their window.

With the monitor's input available, the execution of the system hierarchy can begin. The systems are evaluated in a bottom-up sequence implied by the hierarchy—systems that require data from the source only are evaluated first, and systems consuming the output of other systems are evaluated subsequently. Each system writes the values of its output variables into the *now* section of the blackboard, making it accessible to all subsequent systems. The fact that the monitor is organized as an acyclic hierarchy guarantees that no deadlock can occur, i.e., that at any time at least one system finds all its current input on the blackboard so that it can execute. Independent of this, the input may still be underspecified: if some of a system's input is unknown and the system itself cannot compensate for missing values, it produces *unknown* as output.

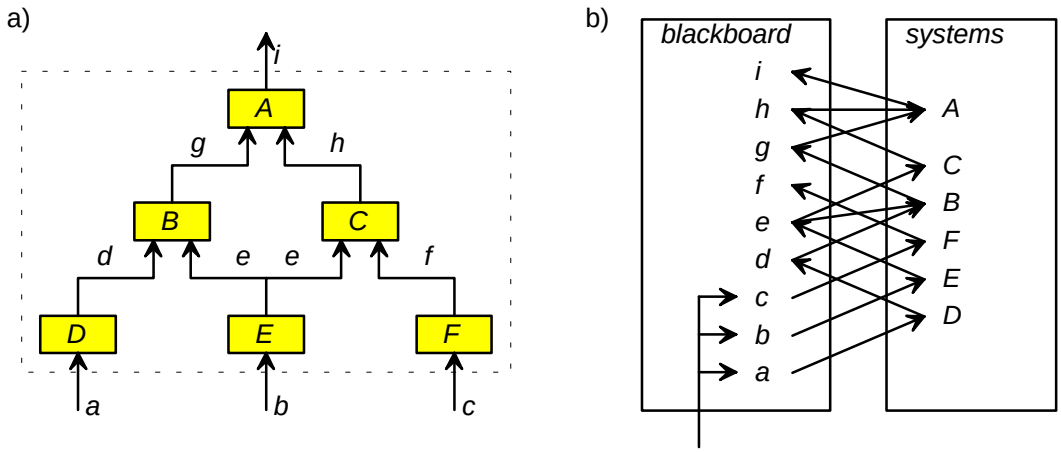


Figure 4.3: Information and control flow
a) flow specified by common input and output variables of systems of the system hierarchy
b) actual information flow: reading and writing of systems from and to the blackboard

The implications of the blackboard architecture on the evaluation process of the monitor are negligible—the reader may verify that other mechanisms, for example passing variables from one system to the next directly as suggested by the block diagram of the monitor, do not improve the order of complexity (which is in fact O(*n*), in either case, where *n* is the number of systems) but require more complicated communication protocols.

Some window-based systems produce a new output even if their current input is missing. This is the case in data smoothing when early samples drop out of the window, and in trend detection when the time of onset is adjusted to reflect the new current time. The request to evaluate can therefore also be understood as a request to synchronize the internal state or memory to the current time.[22]

Last but not least, there are systems whose only purpose it is to provide an output for times at which their input is unknown: the class of interpolators. They serve to synchronize variables sampled at different sampling times by making sure that (interpolated) values are available for every evaluation cycle.

[22] The same considerations give rise to scepticism about whether data-driven synchronization is at all adequate: in order to keep the output of such systems up to date, a least synchronization rate should be guaranteed by the monitor.

The evaluation cycle is completed after the last system of the hierarchy has been evaluated—all derivable information has then been derived and can be found in the *now* section of the blackboard.

4.3 Display and Explanatory Capabilities

A diagnostic monitor that does not present its outcome is useless. Obviously, mere inspection of the blackboard makes no good presentation: current variable values or even excerpts from the history alone will hardly be sufficient to communicate the result of monitoring to the user in an understandable form. More problem-oriented display and explanatory capabilities seem therefore necessary.

As already pointed out in [Factor 90], the system hierarchy of a well-designed monitor reflects the nature of the monitored problem. The presentation of the monitor and the display of its output can therefore be guided by its structure.

Quite clearly, if the main goal of a diagnostic monitor is to reduce the information content of all sampled variables and to present an interpretation in the form of an easily conceivable, high-level abstraction, then only the highest level of systems of the monitor must present its output to the user. This output can even be instantaneous—recalling that the current state of an automaton comprises the history of all variables of its input, there is no need to present the history of its output.

Example

> The instantaneous output “overcorrected respiratory acidosis” tells the observer that the patient has gone through episodes of respiratory and corrected respiratory acidosis and therefore implicitly communicates the history of the patient.

◊

The observer may, however, be inquisitive about the way this interpretation was derived. Statically, this is delineated by the system hierarchy. Dynamically, however, a satisfactory explanation involves the values and courses of all involved variables, together with an explicit reference to the contribution each system has made to the final result. Each system should therefore be capable of presenting its input and output on demand.

Ideally, a diagnostic monitor would present itself to the observer as a graphical depiction of the monitoring hierarchy with only selected instantaneous values shown. Upon request, the observer can then have the monitor explain its outcome by descending in the hierarchy down to the root of the observer’s concern, at each level offering a full graphical history of input and output of the system and the possibility to add the respective display to a window dedicated to the persistent presentation of important variables. Note that this procedure roughly conforms to the interface design based on user models as put forward in [Coiera 91]. A possible implementational framework is provided in the form of a time line browser in [Cousin 91]. The paving stone interface presented in [Factor 90] is a well-devised presentation of the monitoring hierarchy drawing attention only to those systems whose output has reached a level of criticality.

Currently, however, only static displays are implemented.

4.3.1 The *Display* Classes

Not only because medical staff are used to it, but also because our graphical capabilities of perception usually outperform our textual ones, all chronological information is presented in the form of charts called *trend lines*[23]. Trend lines can be combined to form displays of

[23] *Trend* is used here in the colloquial sense of displaying a course, not in the formal sense of trend

individual type and character, for example by displaying input and output of a system together using different colours, or on parallel trend lines.

Each trend line can present its samples in different forms: although discrete-time sequences of samples are usually visualized as dots, the display classes can connect consecutive samples through straight lines thereby suggesting a continuous-time course of the variable. In addition, for the presentation of degree-type output, each display class can fill the area under the continuous-time trend line. One must, however, always bear in mind that continuous display assumes adequacy of linear interpolation, which is not always the case.

DIAMON-1 provides three types of displays: a *steady trend line*, a *moving trend line*, and a *stored trend line*. The former two display their contents on the screen and have a fixed length corresponding to a fixed temporal extent. Their behaviour differs only in the way they present the information. The first starts recording the data at its left and proceeds with time to the right border. Once it has reached its limit, it returns to the left margin and begins to overwrite its previous recordings with the new ones. The second starts at the right margin and rolls its content to the left, where it drops off the display.

The stored trend line writes its output in left-to-right order to a file that can be used to produce hard copies, i.e., paper printouts. The resolution and extent can be adjusted to meet the output device's requirements. All charts presented in the following chapter have been produced by DIAMON-1's stored trend lines.

4.3.2 Employment of Displays

Displays are linked to systems. The way displays are used varies from system to system. Each systems specifies a display format that assembles its own, characteristic display out of trend lines.

Each trend line is labelled with the name of the variable whose values it displays. If input and output of a system are displayed together, its label is that of the variable designated as the output of the employing system.

Trend detectors divide the display into two sections, the upper to show the course of the variable in which the trend is to be detected, and the lower to display the degree of compatibility of the sequence with the trend.

The output of automata currently presents itself as a collection of trend lines each presenting the degree of membership of one state. More intuitive representations of the fuzzy state as suggested in Section 3.9.2 would require display classes specially designed to depict the output of fuzzy automata.

All other systems currently display their output as a single trend line. Examples of all system displays follow in the next chapter.

4.4 Selected Systems

The implementation of most systems is straightforward and deserves no special attention. The following presents a few design issues and features deemed worth recognition.

Frequently used primitive systems that have a single input and a single output variable provide an automatic naming service of their output variable: the label of the output variable is obtained by appending a modifier indicating the system's effect to the input variable label.

detection.

Example

The variable *temperature* expanded by a latch and filtered through a median filter first becomes *temperature, latched* and then *temperature, latched, median filtered*.

◊

4.4.1 The Class *MedianFilter*

Median filters are window-based; they access the *history* section of the blackboard. To avoid re-sorting of the whole window each time a new sample is added (or an old one drops out of the window, see above), each median filter maintains a private sorted collection of all samples included in the window. New samples are then added to this collection whenever they arrive, and old ones (determined from the blackboard history) are removed.

The current implementation of median filters is not weighting, meaning that the temporal distance between successive samples is not taken into consideration. However, there is a minimum window extent specified falling short of which renders the output unknown.

4.4.2 The Trend Detection Classes

Trend detection as defined in Section 3.1 is a rather general task. The core problem is that of deciding whether a given sequence of samples can be explained by a course that complies with the definition of a particular trend. Trend detection has a general and a specific component, the latter depending on the chosen specification of trend.

Currently, only four different types of trends are provided: *constant*, *linear*, *step change* and *bounded linear*. All four are defined via fuzzy courses. Constant and linear fuzzy courses are specified by four baselines, an upper and a lower *zero* baseline and an upper and a lower *one* baseline. The courses have a trapezoidal appearance—all vertices are straight lines. Most of the fuzzy courses employed in Chapter 5 are of this type.

A step change fuzzy course is a concatenation of two constant courses; Section 5.3 contains examples. Bounded linear fuzzy courses resemble linear fuzzy courses, except that their baselines are bounded by some limit. Consequently, they have a “crushed” appearance: a rising (or falling) course, once it has reached its limit, changes into a constant one. An example of this can be found in Section 5.2.1.

A trend detector can operate in three different modes: *absolute*, *floating level* and *derivative*. Accordingly, it uses three different services of the fuzzy courses: membership of a sample relative to a time of onset, distribution of offsets making a sample match the course relative to a time of onset, and membership of a sample in the integrated fuzzy derivative relative to its predecessor and a time of onset (compare respective definitions and theorems of Section 3.7).

The degree of compatibility of a whole sequence of samples is derived in all three modes using the same iteration: the degree of membership is derived sample by sample and combined using the minimum operator (cf. Section 3.7).

4.4.3 The Class *Automaton*

Automata are made up of states and transitions. The transitions name input variables that determine when a transition takes place; the variables are queried each time the automaton is requested to evaluate. The input variables must be of type degree; discriminators and trend detectors are predestined systems to deliver the input for an automaton.

The peak hold transitions (3.30) are added automatically to the definition of an automaton. The normalization currently employed is that of active states (3.33). If a selection of normalization methods is desired, a choice can be implemented in respective subclasses of *Automaton*.

4.5 The Class Hierarchy of DIAMON-1

Like all experimental systems, DIAMON-1 is in flux. Figure 4.4 is hence merely a snapshot of its current implementation and certainly subject to future change (including improvement).

Not all classes found in Figure 4.4 have been addressed in this chapter; some are auxiliary and deserve no further mention, others are somewhat outdated and should be replaced with the next design revision. Vice versa, not all systems presented in Chapter 3 have been implemented; the design of DIAMON-1, however, guarantees straightforward integration of additional systems in the form of respective subclasses.

```
Monitor
DataSource
  ASCIITable
  DBaseIII
  MedRecFile
TimeStampedValue
  BlackboardEntry
System
  MemorylessSystem
    Discriminator
    Mute
  MemoryBasedSystem
    Automaton
    HistorySensitiveSystem
      Alias
      Delay
      MeanSlope
      MedianFilter
      MovingAverage
      NetChange
    Latch
    TrendDetector
Display
  TrendLine
    SteadyTrendLine
    MovingTrendLine
    StoredTrendLine
Scale
  LinearScale
    TimeScale
FuzzyRelation
  FuzzyCourse
    BoundedLinearCourse
    ConstantCourse
    LinearCourse
    StepChange
FuzzySet
  ConvexFuzzySet
  EmptyFuzzySet
State
Transition
TimeObject
  AbsoluteTime
  RelativeTime
Parser
```

Figure 4.4: Class hierarchy of DIAMON-1 (subclasses are indented)

The class *Blackboard* is what is sometimes called a *design class*: although it is an essential part of the design, it is not found in the implementation. Admittedly, design classes may be viewed as bad practice, as they disrupt the continuous transition from design to implementation. In this case, they are a consequence of the fact that the design of DIAMON-1 is newer than its implementation, which has naturally grown from earlier versions that did not provide common memory for systems. In the current implementation, the blackboard is substituted by a collection of *blackboard entries* held in the monitor.

4.6 Recommendations for the Construction of a Monitor

The construction of a specific monitor resembles very much a software development process. When done on a larger scale, it also involves problem specification, analysis, design, and implementation phases. Badly prepared experimenting with different systems and system hierarchies in the hope of finding a good solution, like code-and-fix software development, only too quickly results in “spaghetti” monitors.

Specific design recommendations for a generic task are hard to give. Certainly, most monitors could be organized in conceptual layers: the ones mentioned in Section 2.8.2 are a good basis. However, the individual structure of a monitor depends too much on the respective problem to be forced to take on certain standards.

Synchronization should be a prime concern of low monitor levels. If a data source is known to be asynchronous, i.e., if variables tend to be sampled at different sampling times so that the *now* section of the blackboard would only be sparsely filled, interpolating systems such as latches should be used to synchronize, i.e. to bring together the data from the source.

Also, median filters should be used at low levels of the monitor—their potential for removing artefacts combines nicely with their data smoothing properties.

All monitors should carefully take implicit delay times of window-based systems such as moving averages and median filters into account: relating side-effectively delayed variables with timely others may lead to erroneous results. Explicit delay of the timely variables is then a remedy.

Chapter 5

Application to Intensive Care

Data acquisition in intensive care units and operating rooms is typically characterized by

- dense, automated on-line sampling of all variables of interest for which devices exist,
- laboratory test results obtained from aperiodically drawn samples to verify and supplement automatically acquired data, and
- data from flowsheets and medical imagery.

Preferably, *all* significant variables would be sampled continuously. Unfortunately, automated acquisition of data is still dominated by what can be done rather than by what would be desirable to do. However, in the years to come more and more variables will become available on-line and already existing devices will become increasingly reliable. Automated, comprehensive and continuous patient monitoring is therefore likely to be more than a vision.

In the meantime one will have to make do with a heterogeneous conglomerate of data: density, reliability, and diagnostic value of each variable are qualities that vary depending on its technical source and on the situation being monitored. Although realizing that integration of such different information is a worthwhile, yet nontrivial problem, I decided to work on densely sampled on-line variables only. Even if this approach appears currently impractical, it is a proof of concepts, which is justified by the anticipation that the development of fully-fledged diagnostic monitors may well coincide with the on-line availability of *all* required variables.

To demonstrate the practicability of my approach I selected two situations from intensive care: the first is a tightly monitored episode of adult respiratory distress syndrome (ARDS), while the second consists of data routinely collected by a patient data management system (PDMS) from eight patients weaned from ventilator support after cardiac surgery [Hiesmayr 93]. The former demonstrates the expressive power of DIAMON-1, while the latter points out its practicability in clinical routine.

5.1 Evaluation in Simulated On-Line Operation

Real-time on-line employment is the ultimate probation of every diagnostic monitor. However, it is also the most delicate stage of its development. Shortcomings that may become apparent interrupt or even abort its evaluation and require the development process to be re-iterated. This is the more the case as each monitor must be individually designed to suit a particular task, and a number of test runs is usually necessary to find out which design is the best. The interest of clinical personnel is valuable and, once lost, hard to regain. Therefore, real on-line employment during the experimental stage of development is not advisable—rather, a diagnostic monitor should be tested off-line by operating on data recorded in a file.

For an off-line evaluation of a monitor to be realistic, a few constraints must be met: firstly, no use of the retrospective nature of an off-line employment must be made, which basically means that no look-ahead in the data stream is acceptable. Secondly, all recorded data must be explicitly time-stamped and the monitor must synchronize its internal clock to the respective sampling times in order to process the data correctly. Thirdly, if the monitor is to operate in real-time, then the actual time consumed to process data from one sampling time should generally not exceed the temporal distance to the next sampling time (occasional exceptions may be tolerated as long as the monitor catches up with the sampling sequence). If a diagnostic monitor complies with these requirements, then its off-line operation on recorded data can be called *simulated on-line*.[24]

All monitors described in the following were implemented for and tested in simulated on-line operation. The monitors present their output in the form of charts. All output reproduced in the following was generated by the involved systems' display capabilities.

5.2 Monitoring a Case with Adult Respiratory Distress Syndrome

The adult respiratory distress syndrome (ARDS) is a form of acute respiratory insufficiency. It is characterized by bilateral diffuse interstitial and alveolar lung infiltrates, resulting in progressive hypoxemia which is refractory to oxygen therapy. ARDS usually occurs as a complication of a basic disease and may develop as one of a multiple organ failure. Mortality of patients with ARDS is high (ranging from 50 to 90% depending on its stage and severity) not only due to the ARDS itself, but also to the underlying disease and coexisting complications. [Sittig 89, Niemer 92]

Unfortunately, the global directive for recovery, to let the lung rest, is at odds with primary life-support which often requires aggressive ventilator therapy with high pressures (both peak inspiratory and positive end-expiratory pressure) and high fractions of inspired oxygen (F_IO_2). Although various measures can be taken to improve oxygenation and ventilation, extracorporeal membrane oxygenation (ECMO) and CO_2-removal ($ECCO_2$-R) are often administered as a last resort to save the life of a patient. [Morris 92]

The experiments of this section are based on densely sampled data from a twelve-hour episode of ARDS in an eight-month-old female.[25] Among other variables, the data set contains F_IO_2, arterial oxygen saturation (S_aO_2), heart rate (HR), and mean arterial pressure (MAP) sampled approximately every 15 seconds. The data is stored in a compressed form such that only changes in the variable's values are reported; prior to further processing, data can be expanded by a latch to restore its original format. The courses of expanded F_IO_2 (range: 50–100%), S_aO_2 (range: 80–100%), HR (range: 150–190 bpm), and MAP (range: 50–90 mm Hg) are

[24] Note that the conditions for simulated on-line operation in part reflect the requirements of Section 2.1; the real-time constraint, however, has been added to emphasize the feasibility of the approach.

[25] The data is used by courtesy of Boston Children's Hospital, MA, USA. It was made accessible to serve as a common basis for the participants of the 1994 Stanford Spring Symposium on Artificial Intelligence in Medicine.

presented in Figure 5.1. Note that HR and MAP contain obvious artifacts—in fact, one MAP value read 32000 mm Hg. The dashed vertical bars indicate hand-bagging, periods during which the patient is disconnected from the ventilator and ventilated manually by a hand bag. The purpose of these sessions during which F_IO_2 is increased to 100% is to deliver a nebulized bronchodilator.

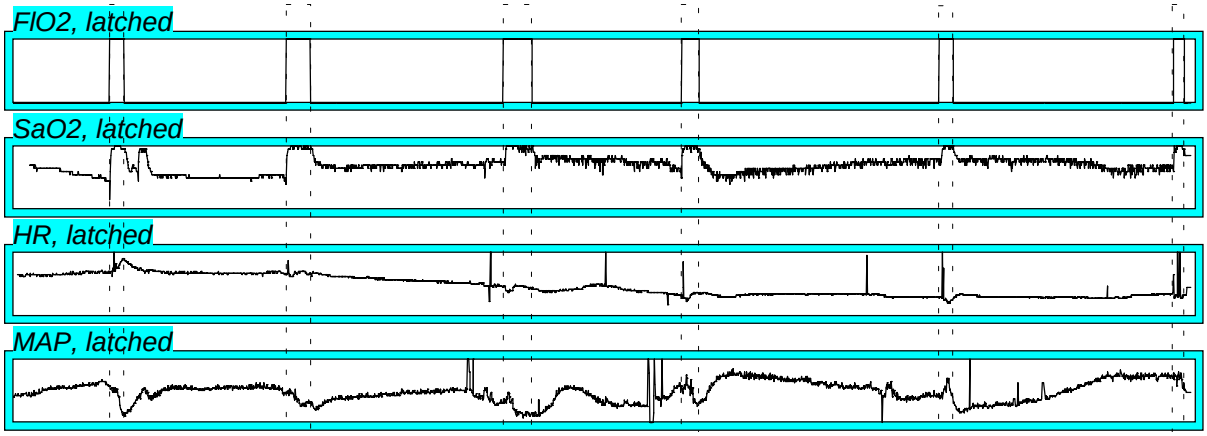


Figure 5.1: Variables of an ARDS patient sampled over a twelve-hour period (expanded by a latch)

The experiments of this section concentrate on the peculiarities of the presented case. Hand-bagging sessions interfere with the global development of the patient as they lead to a temporary increase in oxygenation and expectedly to a (temporary) improvement of the patient's general condition. Haimowitz and co-workers also detected a temporary decrease in blood pressure followed by an increase in heart rate [Haimowitz 94], which may be explained by increased intrathoracic pressure resulting from vigorous hand-bagging.

In the following subsections I describe the design of monitors analysing the patient's response to 100% F_IO_2, the general development of oxygenation, and the effects of hand-bagging on this development. A cardiovascular monitor that produces results comparable to those of [Haimowitz 94] concludes the experiments on this case.

5.2.1 Oxygenation Test

In patients suffering from ARDS alveolar gas exchange is compromised by extravascular lung water, interstitial and intra-alveolar fluid that handicaps diffusion. If the reactivity of oxygenation to high F_IO_2 is in question, an oxygenation test may provide an answer: a sudden increase in F_IO_2 from lower levels to 100% normally results in a sharp increase in arterial oxygen partial pressure (P_aO_2) to values well above normal. The presented case experiences such increases—even though for a different purpose, the effect should be the same.

Unfortunately, P_aO_2 is not continuously available for the given case so that S_aO_2 must be used instead. This is indeed a restriction, as S_aO_2 should always remain above 90% and cannot rise beyond 100%, so that its potential for change is limited. Nevertheless, if saturation is at a comparatively low level, it may be expected to rise sharply to values close to 100%. A fuzzy course representing this expectation is shown in Figure 5.2 a).

The oxygenation test proceeds as follows. It is initiated by an increase of F_IO_2 to 100%. Within 1 minute after this increase saturation is expected to rise sharply, indicating a *positive* effect. If it fails to do so, the outcome is *negative*. After F_IO_2 has gone back to normal, the test can be repeated and is considered *not done*.

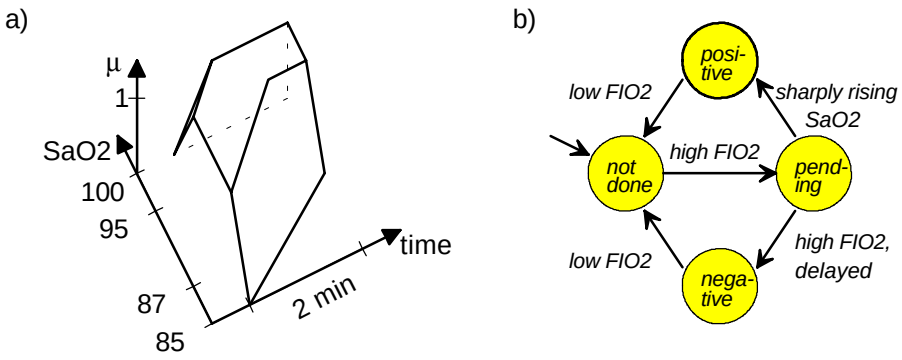


Figure 5.2: Oxygenation test
a) fuzzy course *sharply rising SaO2*
b) state transition diagram of automaton interpreting the test

The state transition diagram in Figure 5.2 b) specifies an automaton relating all states the oxygenation test can take on. The state *pending* is associated with the period during which the automaton waits, either for the oxygenation to increase or for a time-out. A time-out can be realized by delaying the state's entering condition; in this case, *high FIO2* is delayed by 1 minute (tolerance of 21 seconds, see Section 3.1). Figure 5.3 depicts the specification of the entire monitor.

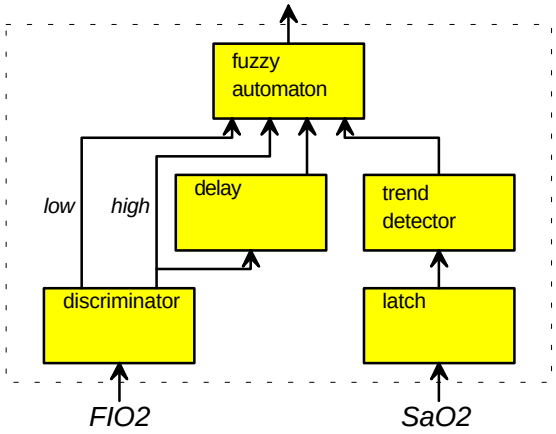


Figure 5.3: Monitor of oxygenation test

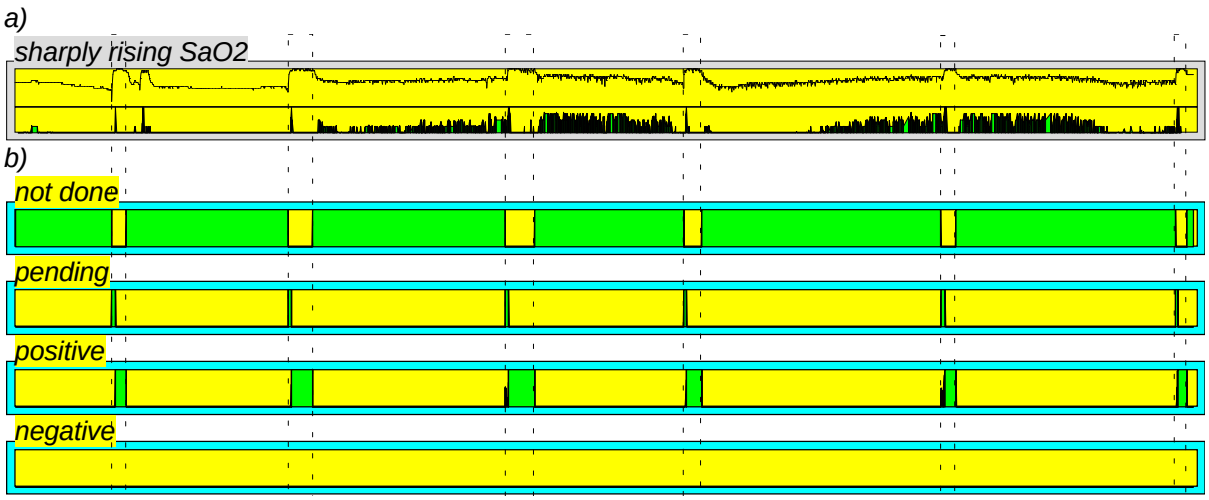


Figure 5.4: Oxygenation test
a) detection of *sharply rising SaO2*
b) state transitions over time

Trend detection based on the fuzzy course of Figure 5.3 a) fully recognizes sharply rising S_aO_2 seven times, six times at the beginning of a hand-bagging session triggering the transition from *pending* to *positive* in Figure 5.4 b) and once during mechanical ventilation (Figure 5.4 a), upper frame shows S_aO_2, lower frame shows degree of compatibility with fuzzy course). Note that the latter does not affect the output of the automaton, as it does not occur while in state *pending*. Also note that floating level trend detection or trend detection relying on the derivative (as defined in Section 3.7) would not be appropriate because the trend to be detected is tied to full saturation (S_aO_2 = 100%) as a landmark value.

5.2.2 General Development

Because oxygenation is the main concern in life-support of an ARDS patient, the general development is assessed here in terms of S_aO_2 only. Alternatively, a score for the well-being of the patient could be derived from a combination of variables, and the systems described in the following could be applied accordingly.

The general development of oxygenation should be reflected in a longer-term trend in S_aO_2. However, to assess the overall development of a densely sampled variable over a long period, direct employment of fuzzy courses is problematic: a step change or a slow ramp in the signal may be equally acceptable to indicate improvement or deterioration, so that a general development is reflected in a net change rather than a characteristic course.

Besides, valuation of oxygenation should always regard the effort, namely its context in the form of ventilator settings. However, the context-dependency of S_aO_2 is not easily taken account of, basically because S_aO_2 is a bounded variable, meaning that its dependency on ventilator settings is not reversible. Hand-bagging is thus an intervention the effect of which has to be cut out for its dependent variable to indicate a general improvement.

To monitor the general development of oxygenation, the decompressed S_aO_2 is therefore passed through a mute triggered by *high FIO2* and smoothed by a median filter with a window width of 1 hour, introducing an (in this case tolerable) delay of 30 minutes. Oxygenation is then classified *rising* if the smoothed signal's mean slope determined over 30 minutes persists to range above 1 percent point (0 percent points) per hour for 15 minutes, *falling* if it ranges below -1 percent point (0 percent points) per hour, and stable if the slope remains between -0.5 percent points (-2 percent points) and 0.5 percent points (2 percent points) per hour (percent points in parentheses denote the lower thresholds for fuzzy classification). The classification process is implemented by trend detection based on the fuzzy courses of Figure 5.5 a), the output of trend detection feeds the fuzzy automaton of Figure 5.5 b).

The complete monitor to derive the general development is specified in Figure 5.5 c). It involves more subsystems than the previous monitor, basically because S_aO_2 requires more preprocessing before it can be passed through trend detection. The outcome of the monitor is depicted in Figure 5.6 a) and b). Note that the output of the monitor is fairly stable, contains far less insignificant information than the input (almost all noise has been eliminated), and is easily processed by subsequent symbolic reasoning. Its main advantage over its input, however, becomes clear when looking at instantaneous values: the output *improved* for example communicates that the patient has gone through an improving phase and then stabilized, which may be far more meaningful than the statement, say, S_aO_2 = 95%.

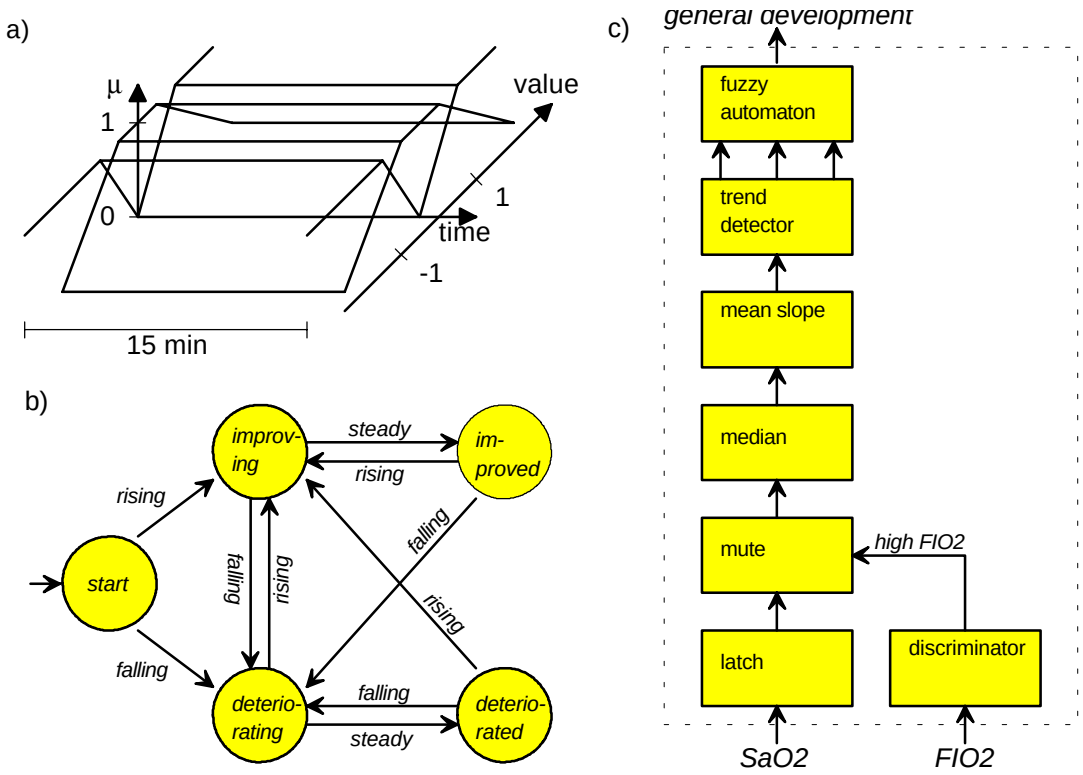


Figure 5.5: a) Fuzzy courses *rising*, *steady* and *falling* to classify the mean slope
b) interpretation by a fuzzy automaton
c) design of a monitor tracking the general development of a patient

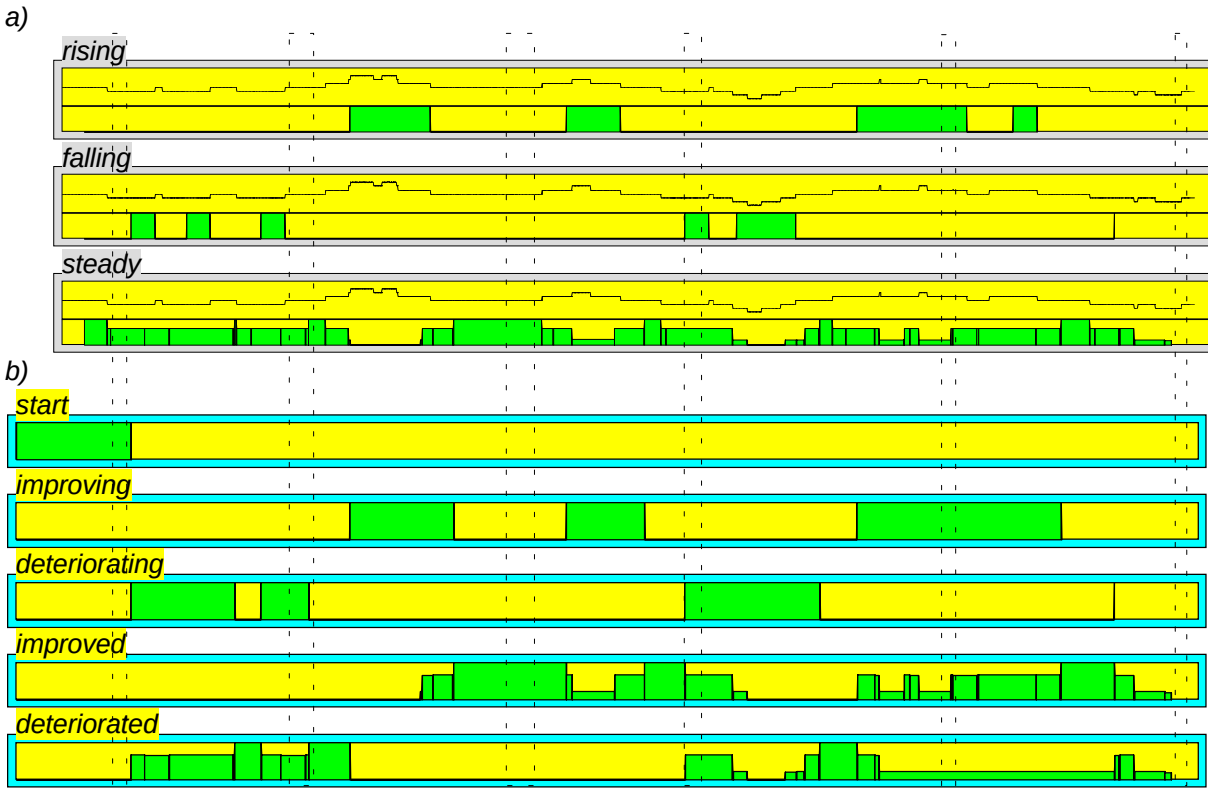


Figure 5.6: General development of the patient as judged by oxygenation
a) trend detection based on the mean slope over 30 minutes, classified by the fuzzy courses of Figure 5.5 a)
b) state transitions triggered by the detected trends; note the delay of 30 minutes induced by the median filter

5.2.3 Effects of Hand-Bagging

In the presented case, the purpose of hand-bagging is to deliver a bronchodilator, which is expected to improve ventilation by reducing resistance of the airways and thus to support gas exchange. The immediate effect of hand-bagging, the sharp increase in oxygenation, must be attributed to high F_IO_2. However, if higher pressures exerted during hand-bagging help recruit previously occluded alveolar spaces, then hand-bagging should have a positive effect on the general development of oxygenation of the patient, and a trained observer should be able to recognize this effect.

Initially, i.e. before the first period of hand-bagging, an effect cannot be observed. Nevertheless, a relative level of the patient's development can be established: if her general development is improving or improved, then a relatively high level may be assumed, whereas if it is deteriorating or deteriorated, the respective relative level is low.

After hand-bagging has ended, a lasting improvement reflected in improving or improved development may be considered a success if the previous level was low, while a deterioration from a formerly high level suggests adverse effects of hand-bagging. Improvements starting from a high level and deterioration from a low level are indicative of no effect—the patient then remains at her respective level of development.

The complete automaton modelling the behaviour of a variable assigned to the effects of hand-bagging is described in Figure 5.7 a). Certainly, more sophisticated models can be devised, however, the one described here is an example designed to demonstrates the feasibility of the approach.

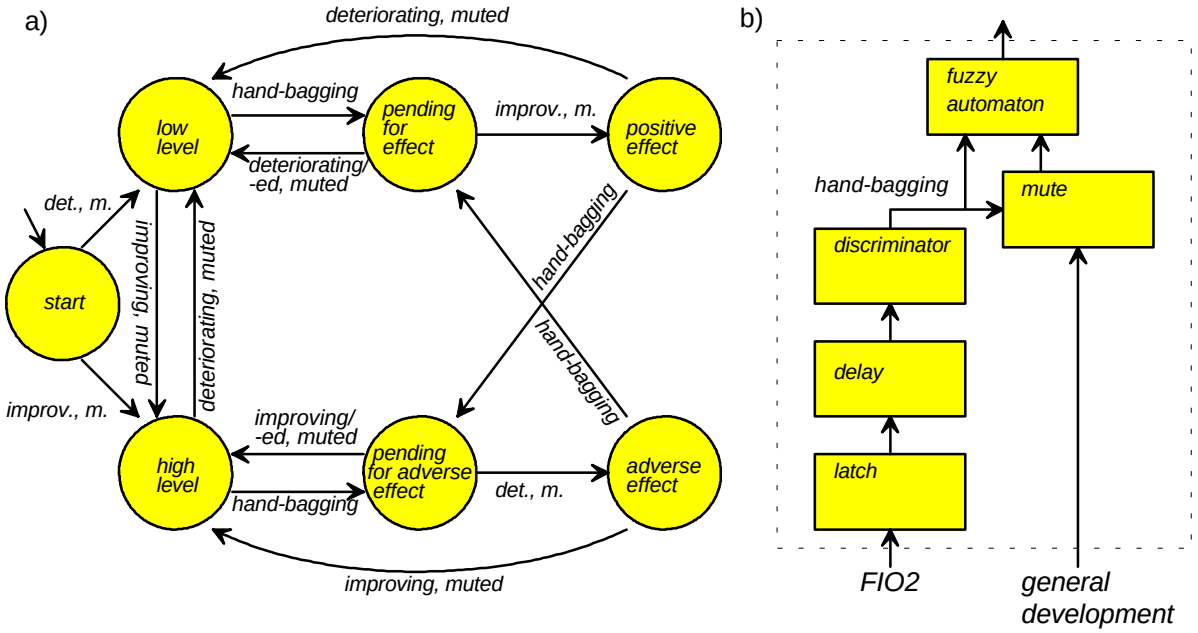


Figure 5.7: a) state transition diagram and b) monitor to trace the effects of hand-bagging

A complete design of a monitor dedicated to the effects of hand-bagging is specified in Figure 5.7 b). Note that *improving*, *improved*, *deteriorating* and *deteriorated* are output of the monitor *general development* described in the previous section. The output of *general development* is muted so as not to bring about a decision before hand-bagging has ended. Note that F_IO_2 is delayed by 30 minutes to take the implicit delay of *general development* caused by the one-hour median filter (see above) into account.

The output of the automaton is shown in Figure 5.8 b). A positive effect is detected only after the second hand-bagging session, which seems in accordance with the course of S_aO_2 repeated

in Figure 5.8 a). Note that the deterioration after session number four is not classified as an adverse effect, because S_aO_2 starts decreasing before the onset of hand-bagging. Its posterior decrease is hence a continuation and no proof of an adverse effect.

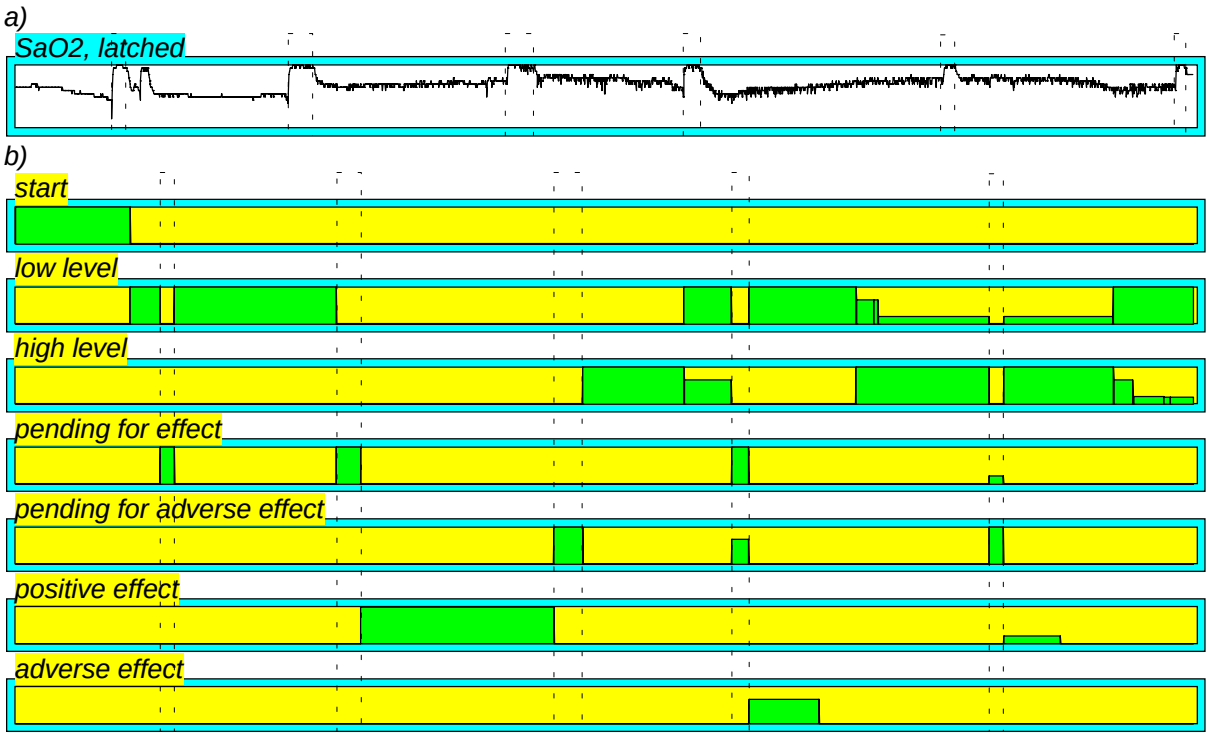


Figure 5.8: The effects of hand-bagging
a) S_aO_2 (repeated from Figure 5.1)
b) assessment of effect through the automaton of Figure 5.7 a) (note again the delay of 30 minutes)

5.2.4 Cardiovascular Response

During pressure-limited mechanical ventilation, airway pressures are dependent on and controlled by ventilator settings. Sudden changes in arterial blood pressure can hence be no direct consequence of artificial respiration—reason for suspicious blood pressure fluctuations during mechanical ventilation of a relaxed patient must be sought elsewhere. The situation is different, however, during hand-bagging: there, pressure is exerted to the lung manually and thus subject to arbitrary alterations.

Careful examination of Figure 5.1 reveals temporary drops in MAP during hand-bagging. Concurrently, a rise in heart rate (if only subtle) can be observed, which is explainable as a compensatory mechanism maintaining cardiac output.

If reduced MAP is attributed to reduced venous return and this in turn to increased intrathoracic pressure (cf. example of Section 3.4), then this finding could hint at the fact that hand-bagging is a little vigorous. A monitor that detects this situation is defined in Figure 5.9 a) and b). A drop in blood pressure (*MAP falling*) is defined as a net change of more than -5 (-1) mm Hg over the last 3 minutes, an increase in heart rate (*HR rising*) as a change of more than 2 (0) bpm over the last 2 minutes.

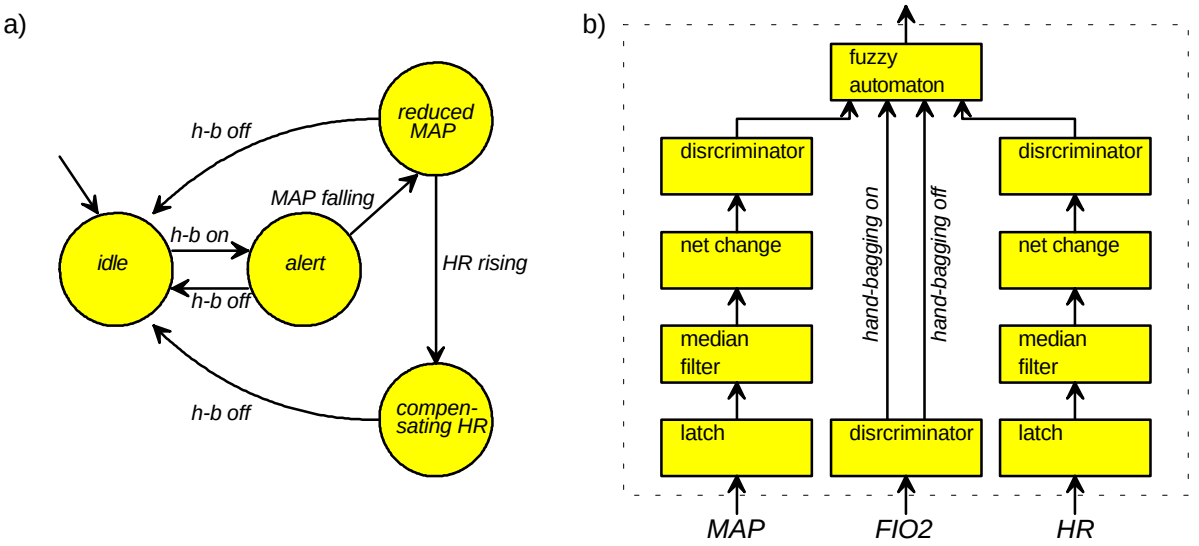


Figure 5.9: Monitoring the cardiovascular response to hand-bagging
a) state transition diagram
b) specification of the monitor

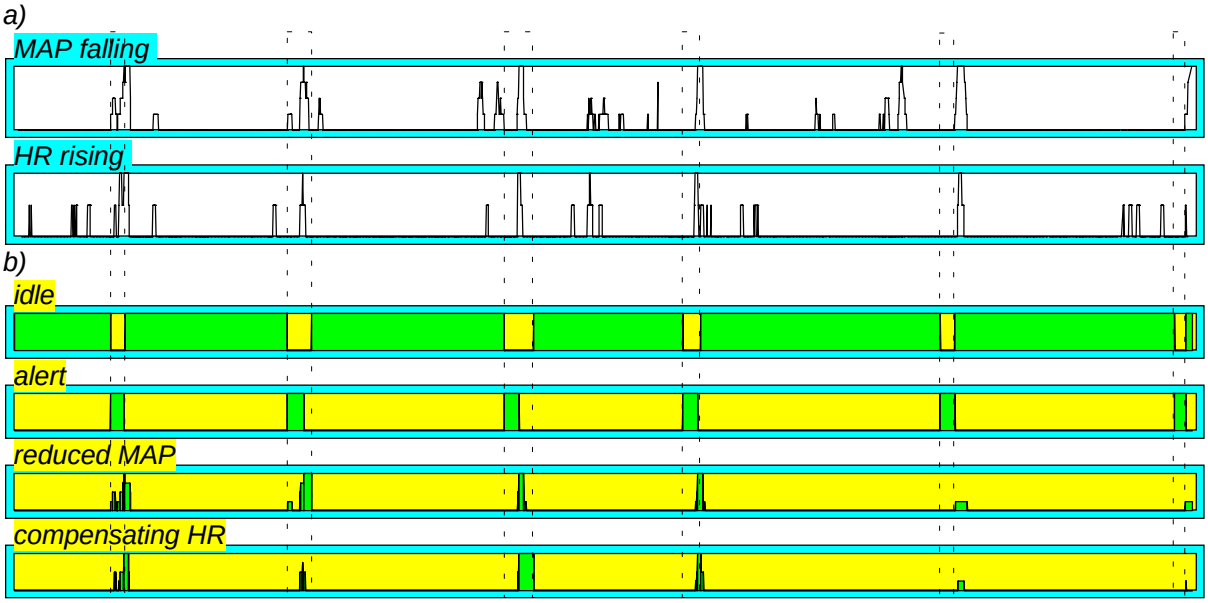


Figure 5.10: Output of monitor watching over the effects of hand-bagging on circulation

Note that the increase in heart rate is several times not diagnosed as compensating, once because it is detected simultaneously with the decrease of blood pressure (second hand-bagging episode), and twice because both blood pressure decrease and heart rate increase occur after hand-bagging has ended (fifth and sixth hand-bagging episode).

Although all involved systems were designed after theoretical considerations, one may suspect that the monitors are tuned to suit the presented case and would produce inferior results if applied to different cases. Indeed, the above experiments do not even prove that the same monitor will work equally well for similar cases. Therefore, the following experiments go to show that DIAMON-1 can contribute to routine patient surveillance.

5.3 Routine Employment

The previous sections presented a highly specialized monitoring task. Consequently, the employed monitors are specially adapted to suit the given case. However, in a routine setting, extensive customization of monitors is rarely feasible, so that practical advances in medical care are primarily to be expected from standardized monitors designed for routine use. In this spirit, the following goes to show how floating level trend detection can contribute to the creation of such monitors.

Weaning patients from respiratory support after cardiac surgery is a typical case requiring routine patient surveillance. Patients are monitored for stable breathing patterns and sufficient ventilation before mechanical assistance is withdrawn. During that time, respiratory data and ventilator parameters are continuously sampled and displayed to the clinical staff. A patient data management system (PDMS) can additionally store the collected information and prepare it for further use, for example by compiling and printing 24-hour charts.

The following experiments operate on compressed data sets as compiled by a PDMS. Each data set presents an excerpt from a patient history. Episodes are not annotated—in particular, it is not clear whether and when weaning was actually initiated. Consequently, verification of automated decision support (indicating, for example, when weaning should be attempted) based on the given data would be difficult. I therefore decided to focus on trend detection in isolation and not to further interpret the findings.

5.3.1 Data Preparation

The PDMS reads data from its input devices at adjustable rates. In the given case, each input channel is compressed to report the average of 10 minutes sampling, and the result is time stamped and stored in a file for further processing.

The data material underlying my experiments is left as it was—containing frequent artefacts[26] and sometimes long intervening periods of missing values, partly affecting one variable, partly affecting all. In a way, the chosen scenario is close to the worst case, the positive thing about it being that it is very "real".

Following Dojat's and his coworker's closed-loop weaning project GANESH [Dojat 92a], four variables are monitored: end tidal CO_2 ($ETCO_2$), respiratory rate (RR), tidal volume (VT), and pressure support, in this case defined as peak inspiratory pressure (PIP). All variables are smoothed by a median filter of 35 minutes with 11 minutes tolerance (covering, under normal conditions, three samples, thus being able to filter out single artefacts).

5.3.2 Specified Trends

Figure 5.11 depicts the fuzzy courses underlying trend detection: stability of $ETCO_2$, RR, and VT is assessed based on constant trapezoidal fuzzy courses, while reduced pressure support is modelled as a persistent step change of at least one unit.

[26] Note that averaging as performed by the PDMS was obviously not able to filter out the artefacts.

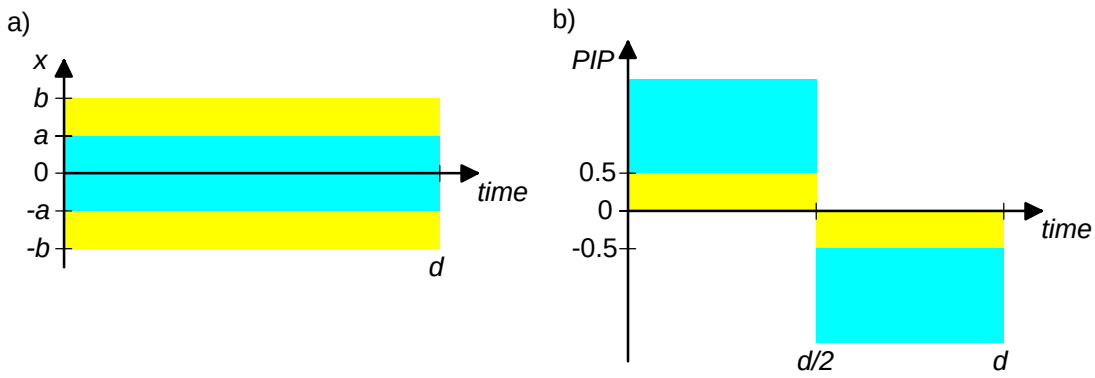


Figure 5.11: Fuzzy courses defining trends
a) *a* = 2 mm Hg, *b* = 5 mm Hg for *stable ETCO2*; *a* = 1 1/min, *b* = 3 1/min for *stable RR* ;
a = 30 ml, *b* = 10 ml for *stable VT*;
b) *reduced pressure support* (defined on PIP in mbar);
d = 2 hrs for 60-hour trend line and *d* = 6 hrs for 14-day trend line

The duration *d* of the trends varies with the period over which the patients are observed. For better comparability, the patients are divided into two groups, one including five patients monitored for up to 60 hours, and one with three patients monitored over up to fourteen days. For the former, trends extend for two hours, while for the latter they extend for six; different extents are chosen merely to demonstrate the applicability of trend detection to relatively long sequences of samples (approximately 360 in the latter case).

The design of the monitor is depicted in Figure 5.12. It is very regular and requires no further explanation.

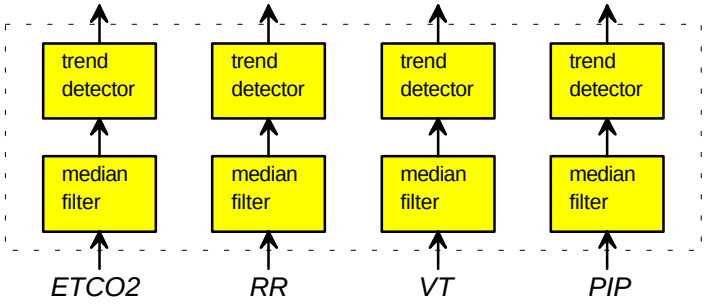


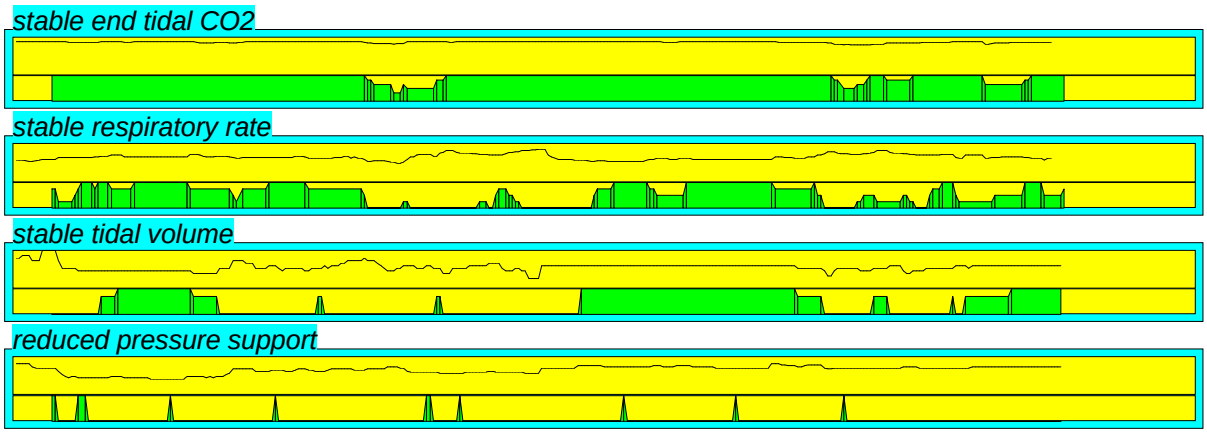
Figure 5.12: Design of a monitor for routine monitoring of weaning candidates

5.3.3 Results

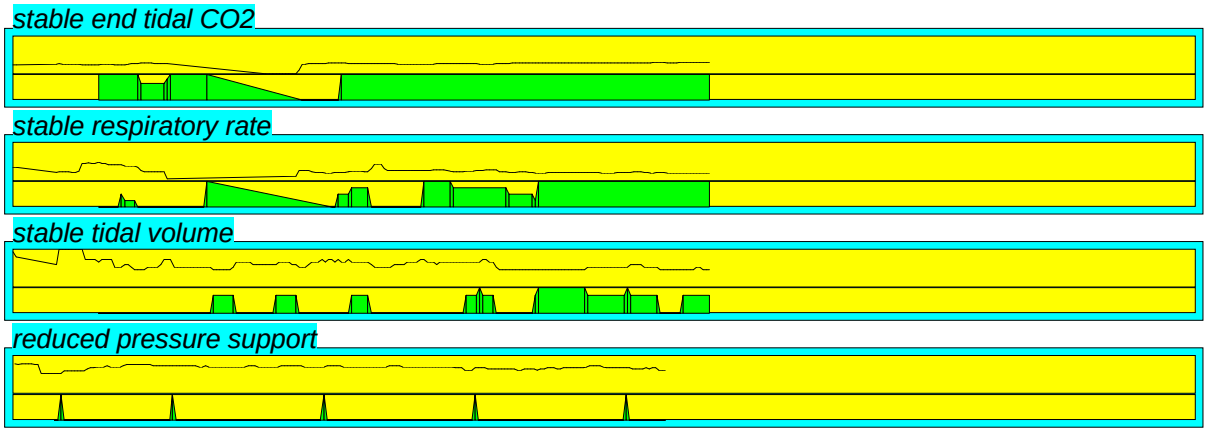
The results of applying the monitor to the given patient data are presented in Table 5.1 and Table 5.2. The ranges and units for the trend lines are defined as follows: 0–110 mm Hg for $ETCO_2$, 0–50 1/min for RR, 0–1.5 l for VT, and 0–35 mbar for PIP.

Table 5.1: Five patients monitored over up to 60 hours

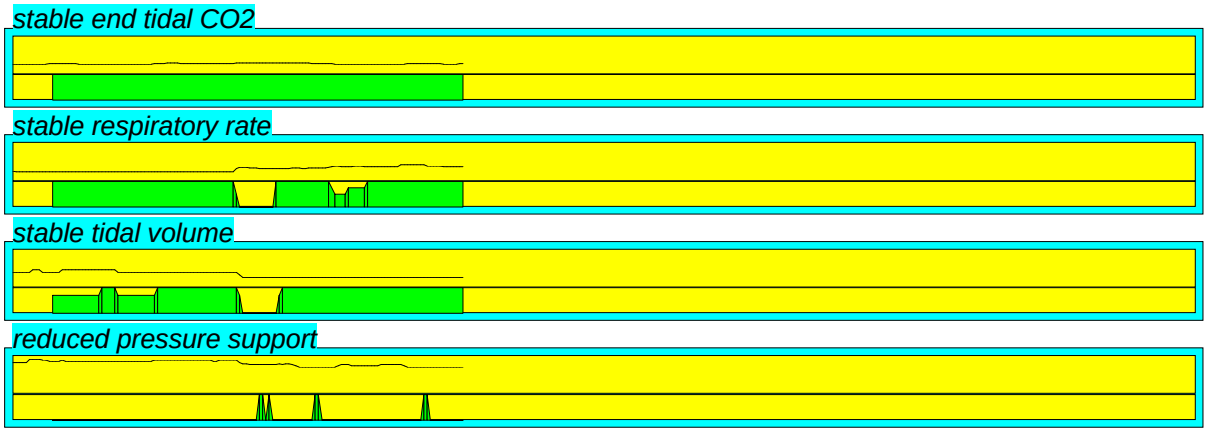
PATIENT 1



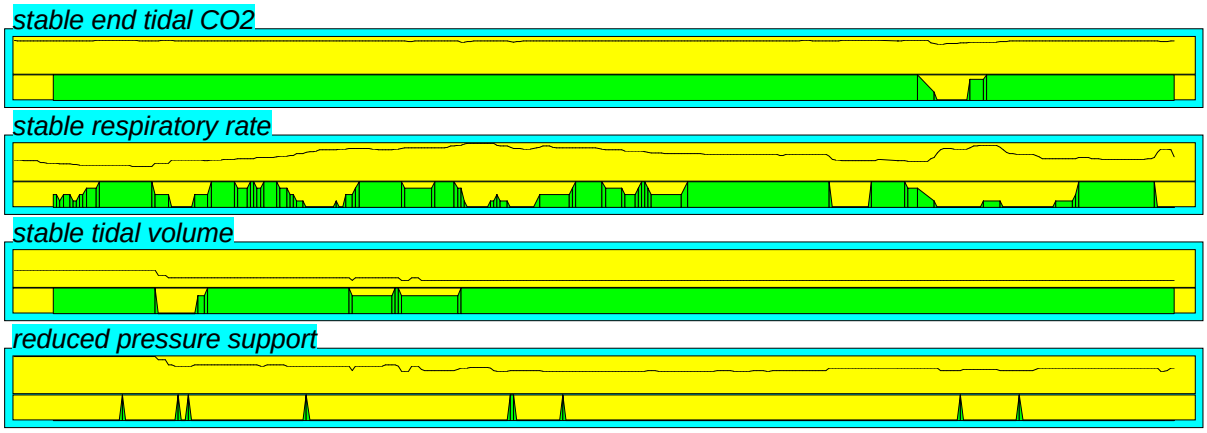
PATIENT 2



PATIENT 3



PATIENT 4



PATIENT 5

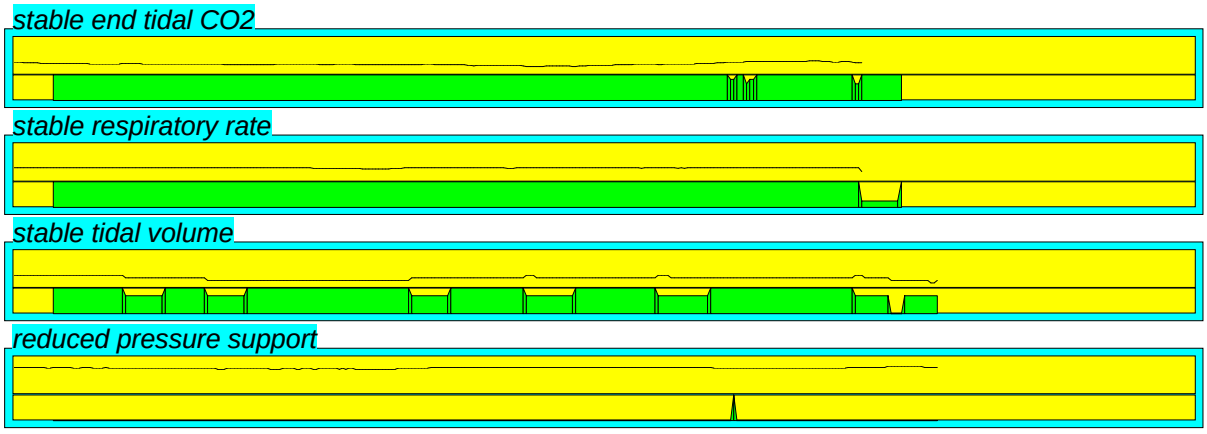
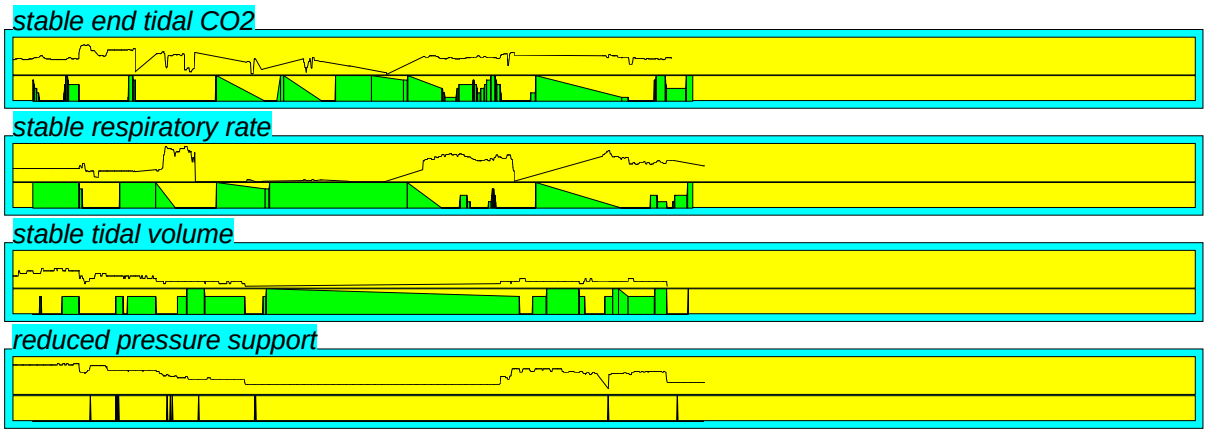
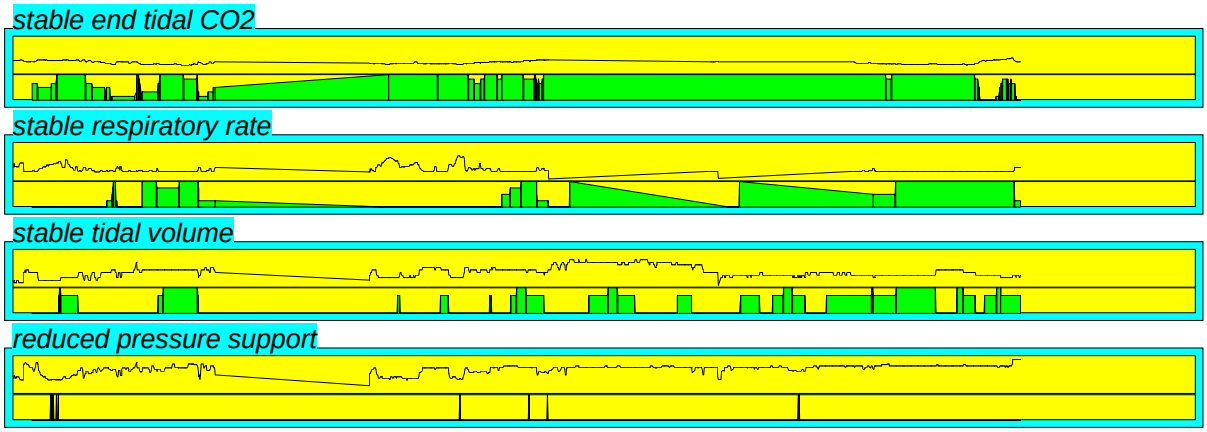


Table 5.2: Three patients monitored over up to 14 days

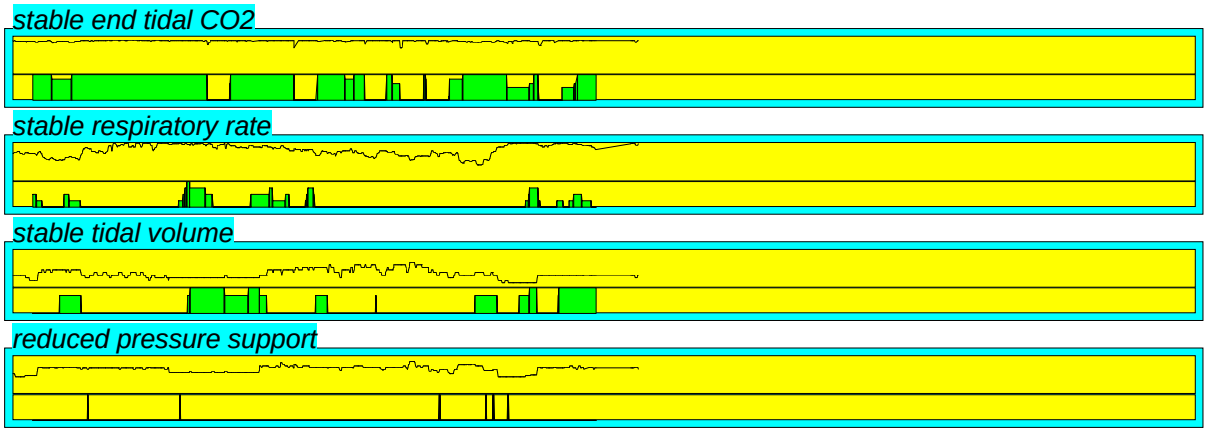
PATIENT 6



PATIENT 7



PATIENT 8



The ramps found in both input and output trend lines of Patients 2, 6 and 7 are due to interpolation over longer intervals of missing values. With the chosen presentation (data of up to 14 days, sampled every 10 minutes), a non-interpolating display, although better revealing these gaps, would have made the diagrams more difficult to read.

The examples show that trend detection based on floating level fuzzy courses is capable of detecting long-term stability of variables. Also, the detection of step changes, where the observed value persistently crosses an arbitrary threshold, presents no problem (as demonstrated

by the reliable detection of *reduced pressure support*). Again, trend detection based on fuzzy courses proves to be intuitive and straightforward, thus making it easy to formulate and apply.

One may wonder why stability is not defined in terms of the first derivative: a stable course, independent of its level, should be characterized by a derivative remaining close to zero. However, as Figure 5.13 demonstrates, detecting longer-term trends (involving, say, ten or more samples) via the first derivative poses nontrivial problems: minor fluctuations in a signal (that cannot always be filtered out by data smoothing) require a certain laxity in the constraints which, on the other side, allows a persistent drift in one direction, which in turn is at conflict with the concept of stability.

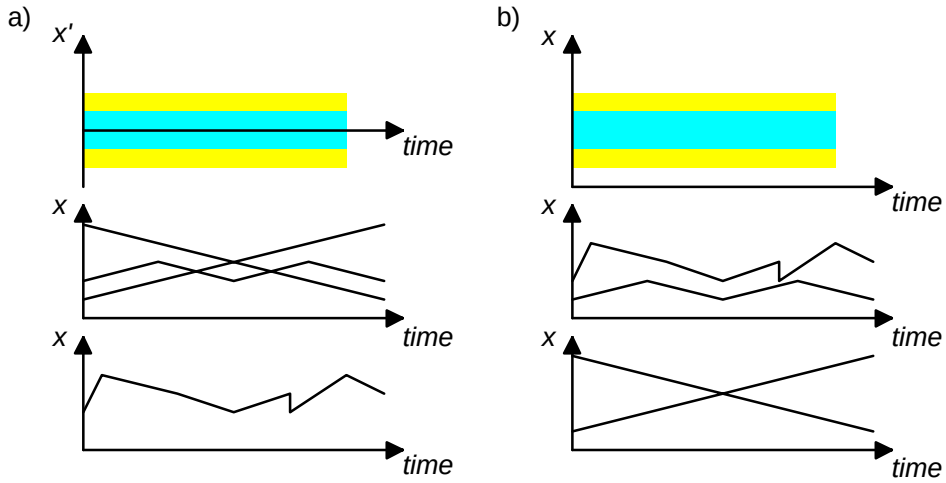


Figure 5.13: Different properties of trend detectors based on
a) fuzzy first derivatives and b) floating level courses;
top row defines trends, middle row shows courses compatible with trends, and bottom rows show courses incompatible with trends

The admissible drift in floating level trend detection based on (constant) absolute fuzzy courses, on the other hand, is limited by $2a/d$ (see Figure 5.11 for lettering) for full degree of compatibility and thus the smaller the longer the trend to be detected is. Nevertheless, the slope of an explaining course between two subsequent samples may reach much higher values; in fact, the instantaneous slope is not constrained at all, as Figure 5.13 b) exemplifies. Note that properties are reversed if trends are short: a short floating level trend may well be compatible with a sequence of samples exhibiting a significant drift, while one based on the first derivative may be better suited to filter out such stray courses.

Summarizing, one should be aware that the choice between first derivative and floating level trend detection is not arbitrary, but has its implications. Which one is better suited depends on various factors, including the duration of the trend and the properties of the signal. In any case, the floating level trend, operating on absolute courses, is easier to specify, particularly for persons with poor mathematical imagination, and is therefore likely to become more broadly accepted.

Chapter 6

Application to Serodiagnosis

The examples of the previous chapter have demonstrated the practicability of DIAMON-1 and its components in densely sampled domains. Interpretation of sparse samples, on the other hand, is a different matter, and applicability of a method in a dense data domain has little implications for the sparse data domain. In the following sections I will therefore show how trend detection of Section 3.7 can be applied to the problem of screening pregnant women for infection with *Toxoplasma gondii*, which is a typical representative of the sparsely sampled domain.

6.1 Outpatient Monitoring

As compared to intensive care unit or operation room monitoring, outpatient monitoring is characterized by a rather loose surveillance of patients linked to occasional (or, less frequently, regular) doctor visits. A diagnosis, if only preliminary, is made on each visit, and future visits are scheduled on demand. If the patient can perform the required measurements himself (as, for example, blood glucose testing in case of diabetes), the attending physician can rely on a fairly dense sequence of recordings. More frequently, however, measurements are made only during the visit, and the sampling sequence is determined by practical circumstances as much as by medical needs. It is therefore almost invariably sparse.

Throughout the following sections I will describe my approach to automated diagnostic support for outpatient monitoring employing the same methods used for trend detection in intensive care.

6.2 Screening for Infection with Toxoplasma Gondii

By definition, screening is the routine examination of selected populations. Screening for infection with Toxoplasma gondii involves only pregnant women, because only these, as will be seen below, carry a certain risk.

6.2.1 Aetiology and Pathogenesis

Billions of humans and animals all over the world are infected with the parasite Toxoplasma gondii. Its main mode of transmission is by ingestion of oocysts, toxoplasma cells excreted by cats and spread around by flies and other insects, and by consumption of raw meat of infected animals.

Once acquired, the pathogen spreads throughout the body via lymphatics and the bloodstream and infects tissue cells, mostly of the central nervous system, heart, and muscle. After this acute phase of infection, cell-mediated and humoral immune response remove the parasite from the bloodstream, leaving viable organisms to persist encysted in the host tissue. This latter *chronic* or *latent* stage of infection is assumed to last for the whole lifetime of the host. [McCabe 83, Desmonts 85]

Toxoplasmosis is the clinical disease associated with an infection with Toxoplasma gondii. It is not an obligatory consequence of acute infection—only in about 10% of the population postnatal infection is symptomatic. Frequent signs of *postnatal* toxoplasmosis are those of mononucleosis, namely malaise, fever, headache and swollen lymph nodes. Because it is generally well controlled by the immune system, toxoplasmosis is not life-threatening for the immunocompetent person. More recently, however, postnatal toxoplasmosis has gained in significance, as the number of immunocompromised (such as transplant recipients and other immunosuppressed patients) and immunodeficient (AIDS!) individuals increases.

Prenatal toxoplasmosis, on the other hand, has longer been a matter of concern: transplacentally transmitted pathogens causing congenital toxoplasmosis can destroy the foetus's developing tissue and cause irreparable damage. This fact gives rise to toxoplasma screening; more on this below.

6.2.2 Serological Tests

Diagnosis of toxoplasma infection can be based on serological tests detecting specific toxoplasma antibodies. These tests vary in the type of antibody and in the quality they respond to. Most frequently used are tests for the determination of IgG and IgM antibody concentrations. More recently, tests to measure IgG antigen-binding avidity [Lappalainen 93] and IgA [Bessièrs 92] have come to support serodiagnosis.

The laboratory where I conducted my studies, the Toxoplasmosis Laborartory of the University Children's Hospital in Vienna, Austria, currently uses three tests:

1. Sabin-Feldman dye test (DT), the World Health Organization reference test measuring mostly specific IgG antibodies,
2. immunosorbent agglutination assay (ISAGA) for the detection of IgM antibodies, and
3. solid-phase enzyme immunoassay for the determination of IgG antibody avidity.

All test results are obtained and further processed in their quantitative form, i.e., the DT as a titer in steps of fourfold dilution, IgM-ISAGA as an index in the range of 0–12, and IgG avidity as a percentage.

The DT is highly sensitive and specific, i.e., positive titers prove toxoplasma infection while negative exclude it. Because it requires living parasites, it can only be performed at few reference laboratories. Unfortunately, its capability of discriminating acute from chronic infections based on a single serum is limited, as individual immune response to toxoplasma antigen varies

considerably in both strength and speed so that reliable assessment of acuteness can only be based on relative changes. The IgM ISAGA is used to rule out acute infection: while high indices can occur with both acute and latent infection, a negative IgM is never found with acute infections. The IgG avidity test is a new technique for the measurement of the antigen-binding avidity (functional affinity) of IgG distinguishing low-affinity antibodies at an early stage of infection from those with a higher binding affinity reflecting pre-existing immunity [Lappalainen 93].

6.2.3 Courses of Infection

The serological findings in a patient depend on the age of infection: a very recent (acute) infection typically presents a different picture than an old (latent) one. One can thus speak of typical *courses of infection*.

In response to primary contact with antigens, first specific IgM and then IgG antibodies are produced by the immune system. Both quality and quantity of antibodies take typical courses: Figure 6.1 is an adaptation of courses published in [Desmonts 85, Bessières 92, Lappalainen 93]. Unfortunately, actual courses vary considerably from patient to patient: with some patients, IgM remains high for years, while with others the IgG response is so moderate that an acute finding can easily be mistaken for a latent one.

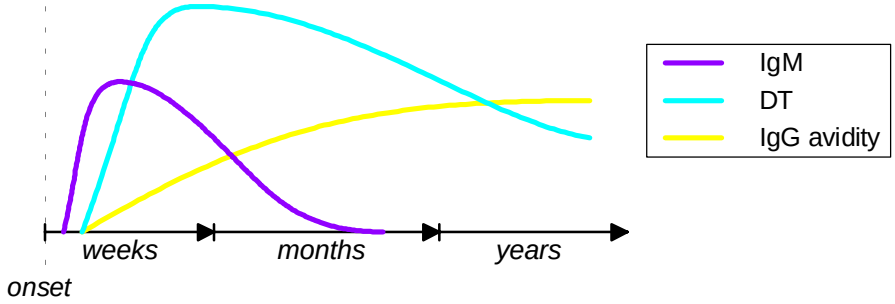


Figure 6.1: Typical course of antibody production after primary infection with Toxoplasma gondii as observable through various tests (adapted from [Desmonts 85, Bessières 92, Lappalainen 93])

Specific antibodies usually persist for a whole lifetime, although in varying concentrations. In general, repeated contact with antigen leads to a secondary immune response, which is usually faster and stronger than the primary. However, typical courses of reinfection with Toxoplasma gondii have not been published—aparently, repeated contact with the parasite leaves the antibody concentration unaltered. This is maybe because cysts leak and so emit antigens to the bloodstream, thus keeping immune activity on a certain level. Whatever the true reason is, a significant increase in antibody concentration appears to be symptomatic of acute infection and acute infection only. This fact is heavily relied on in serodiagnosis.

6.2.4 Congenital Toxoplasmosis

Transplacental transmission of the parasite from an infected mother to the foetus is possible only under certain circumstances: practically, only cases of primary infection acquired during pregnancy (so-called *postconceptional* or *gestational* primoinfections) present a risk[27]. If, on the other hand, infection is acquired before conception (so-called *preconceptional infection*), no practical risk of transmission exists [McCabe 83]. Figure 6.2 visualizes the terminology.

[27] which further depends on the gestational age

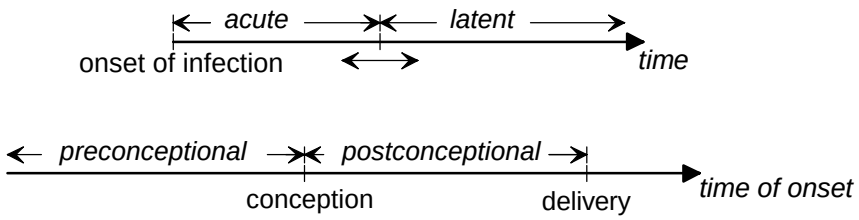


Figure 6.2: Naming of infections

In the context of congenital toxoplasmosis it is thus not the acuteness that matters, but the time of onset relative to the date of conception. It is important that acute and postconceptional infection are not confused, nor latent and preconceptional infection.

Example

A woman with evident toxoplasmic lymphadenopathy (diseased lymph nodes diagnosed by, e.g., biopsy, as due to acute toxoplasma infection) who becomes pregnant is acutely, yet preconceptionally, infected and therefore bears no risk of transplacental transmission.

◊

Congenital toxoplasmosis is known to cause severe damage ranging from foetal death and stillbirth through hydrocephalus to clinically healthy newborns with an 80% chance of developing occular toxoplasmosis and blindness in adulthood. It is therefore to be avoided.

6.2.5 Screening of Pregnant Women

Current screening for toxoplasma infection as implemented in France and Austria aims at preventing congenital toxoplasmosis [Aspöck 92]. To achieve prevention, all pregnant women with postconceptionally acquired infection must be identified and treated.

Usually, a pregnant woman is attended by an obstetrician who draws serum samples at predetermined intervals and has them examined by a laboratory. The laboratory returns the results, possibly attributed with interpretation and therapy suggestions, to the obstetrician who administers further examinations and treatment if necessary. In difficult cases, specimen can be sent to a reference laboratory performing additional tests with the aim of further clarification.

The efficacy of toxoplasma screening has been shown in several studies, for example [Aspöck 92, Lappalainen 92]. With early diagnosis and treatment, the incidence of congenital toxoplasmosis can be reduced from 2.4–7‰[28] to less than 0.1‰. These figures imply that only few cases of postconceptional infection are overlooked, which means that the diagnostic procedure currently employed in screening is highly sensitive. Figures from the toxoplasmosis laboratory I worked with, however, suggest that it is not very specific, as of approximately 180 treatments per 10,000 births 110–156 cases (61%–87%) are treated in excess of what would have been necessary.

6.2.6 Problems of Serodiagnosis

Because diagnosis of postconceptional infection requires time-dependent interpretation of all findings, it is a nontrivial task that can benefit from analytic support. However, even if mathematical methods are employed, diagnosis based on a single serum is not likely to be both sensitive and specific [Ades 91]. In fact, more often than admitted diagnosis of postconceptional primoinfection with Toxoplasma gondii is a dilemma.

Uncertainty in serodiagnosis is basically due to the wide variability of individual courses of infection: certain samples typically found in preconceptional infections can somtimes also be found in a postconceptional infection. In clinical practice this means that a therapy suggestion cannot always be based on irrefutable evidence, leaving the clinician with a difficult decision:

[28] wide spread partly due to different decades and regions from which figures stem

in the face of risk of foetal infection, waiting for follow-up serology (and thus potential clarification) cannot be afforded, whereas subsequent revision of a diagnosis that induced treatment is awkward for both mother and attending obstetrician, the latter the more as it reflects upon the quality and credibility of the obstetrician's and laboratory's work.

Serodiagnosis based on sequences of samples, although potentially more decisive, is a complex problem. The significance of time as the crucial factor in differentiating pre- from postconceptional infections boosts complexity to orders of magnitude far beyond human comprehension. Roughly estimated, if n is the number of sera drawn, c is the number of possible test result combinations obtainable by the tests performed on one serum, time is discrete with a temporal grid width of one week, and a pregnancy lasts 40 weeks, then the number of theoretically possible combinations is

$$c^n \left(\begin{array}{c} 40 \\ n \end{array} \right),$$

which amounts to 15,823,936,440 for three samples, nine possible DT titers, and 13 possible ISAGA indices (not regarding IgG avidity). This explains why test results are usually abstracted to a small number of qualitative terms such as *low*, *borderline* and *high*, even though this does not affect the complexity due to time (compare [Steimann 95]).

Of these two problems of current serodiagnosis, the diagnostic dilemma and complexity, only the second can be alleviated by computer support. Nevertheless, a statement from the computer that diagnosis of a certain case is uncertain can reassure the obstetrician in as much as it shows that uncertainty is a matter of fact and not due to his lack of expertise.

6.3 The ONSET Program

Driven by the specific problem of toxoplasma serodiagnosis we developed a computer program called ONSET that derives possible times of onset of infection from serological findings and specific knowledge about natural courses of infection. Although developed independently, it relies on concepts very similar to those of trend detection described in Section 3.7, and it is the concern of the next subsection to present the functioning of ONSET as an alternative employment of trend detection rather than an application-specific solution.

6.3.1 Redefinition of the Trend Detection Problem

Clearly, consecutive findings of toxoplasma serology present a sequence of samples comparable to those found in inpatient monitoring. Interpretation of such sequences requires understanding of possible developments in a variable and their meanings—this is independent of the application domain. However, whereas interest in intensive care or operation room monitoring is naturally in developments lasting in the order of seconds to days, screening for gestational infections studies developments occurring over weeks, months, or even years. In addition, toxoplasma screening has only one complication (the acute infection) with one typical course to be watched out for, and the goal of diagnosis is to determine the onset of this course rather than to differentiate different underlying causes. Nevertheless, the basic problem is the same: a sequence of findings must be matched with a set of courses that make up a physiological trend. In fact, if all serological knowledge about the course of toxoplasma infection as observable through a test x is encoded in the trend $\widetilde{C}_x$, the degree of compatibility

$$\gamma(\langle x[t_n] \rangle, \widetilde{C}_x, t_\Omega)$$

denotes the degree to which infection at time t_Ω (the onset of infection) explains the serological findings $\langle x[t_n] \rangle$. Deviating from the on-line monitoring problem, however, t_Ω is not linked

to the current time; rather, it is the independent variable of trend detection implemented as a system

$$y[t_\Omega] = \gamma(\langle x[t_n]\rangle, \widetilde{C}_x, t_\Omega) \tag{6.1}$$

which is not causal because $t_n > t_\Omega$ for some t_n and t_Ω. Input and output of such a system applied to a sequence of two samples are depicted in Figure 6.3; note that the output (solid black line) is a continuous distribution of compatibility over time.

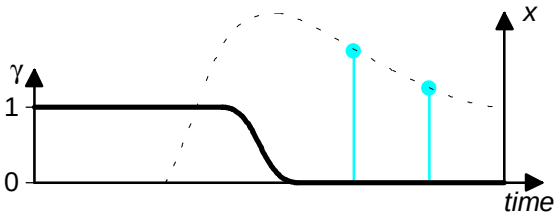


Figure 6.3: Input and output of a system determining possible times of onset of infection from samples of a test x; the output (solid line) is a distribution of compatibility γ over time, the dashed line is an explaining course

Despite the noncausality of (6.1), routine serodiagnosis is an inherently prospective task, as at any time t_n at which a diagnosis is made only samples up to t_n, i.e., $\langle x[t_m] | t_m \le t_n \rangle$, are available. It is the nature of the problem that

- not only compatibility with an onset of infection at t_n, but also at all previous and subsequent times is of interest, and
- previously assessed times of onset must be revised as new information becomes available.

Integrated in the monitoring framework of Chapter 2, determination of the time of onset is a causal system that consumes a sequence of findings $x[t_1]$ up to $x[t_n]$ and produces a distribution of compatibility over time at every t_n. Although this interpretation is theoretically sound, it is not very practical as it requires its own special data type, namely a distribution of degrees over time, and it is not at all clear how subsequent systems should further process this output. In the following I will therefore refrain from embedding determination of onsets in the general monitoring framework and describe serodiagnosis as an independent process.

6.3.2 Serodiagnosis as an Incremental Process

Serodiagnosis of toxoplasma infection very much relies on the combination of evidence that taken alone is very weak. Characteristically, possible onsets are determined by means of exclusion: with each new sample, potential times of onset can be excluded, the previous diagnosis can be refined (become more specific), and the degree of indecision can be reduced.

The following examples schematically demonstrate the derivation of diagnoses from typical findings. Note that trend detection based on the fuzzy courses actually employed by ONSET leads to slightly different results.

Example (postconceptional seroconversion)

Initially, when no information is available, onset of infection seems equally possible at all past and future times. The corresponding distribution of compatible times of onset is shown in Figure 6.4 a). With the first DT sample, however, the picture changes drastically: if it is negative, the patient is not yet infected so that only future infection remains possible, a fact that is reflected in the distribution of Figure 6.4 b). A subsequent positive finding excludes later onset of infection, so that possible onsets are restricted to times between the samples, as shown in Figure 6.4 c).

◊

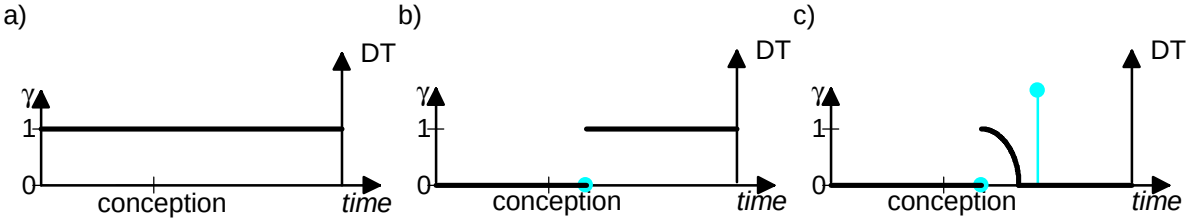


Figure 6.4: Distribution of compatibility of times of onset a) when nothing is known, b) after a negative sample, and c) after a seroconversion

Example (preconceptional infection)

If the first DT performed during pregnancy is positive, infection is evident, yet may have equally possibly been acquired at times before and after conception, as shown in Figure 6.5 a). Note that the titer height alone is not indicative of the recency of infection—both high and low titers can be found in acute and latent infections. An ISAGA test performed on the same sample may provide a different picture: as Figure 6.5 b) demonstrates, if the sample is negative, the patient is either seronegative or infection is latent. Only in combination with the DT of Figure 6.5 a) can the infection be dated back to times before conception, as shown in Figure 6.5 c).

◊

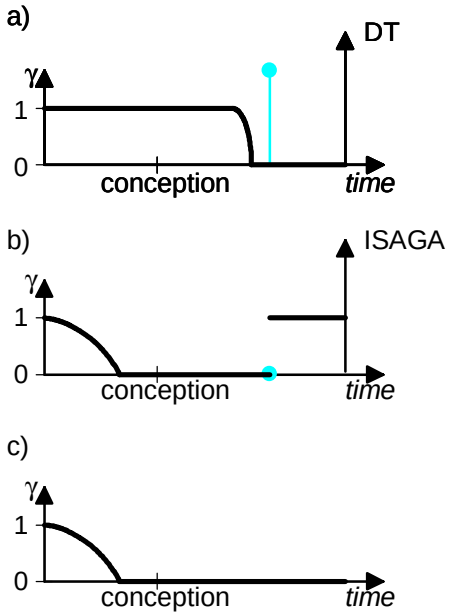


Figure 6.5: Same as Figure 6.4 for
a) a positive DT,
b) a negative IgM ISAGA, and
c) the combination of both

Example (pre- or postconceptional infection: diagnosis uncertain)

If a first positive DT is complemented by a positive IgM ISAGA, a postconceptional infection cannot be excluded. On the other hand, high IgM does not prove recent infection, as high titers can persist for years, so that the distribution of posible onsets after the first sample resembles the one of Figure 6.6 a). In this case, a second sample is required. If this second sample produces the same results, diagnosis may still be uncertain; Figure 6.6 b) presents such a case. Note that the diagnosis might have been different if the DT

titer had risen significantly—this would be suggestive of a recent (but not necessarily postconceptional) infection.

◊

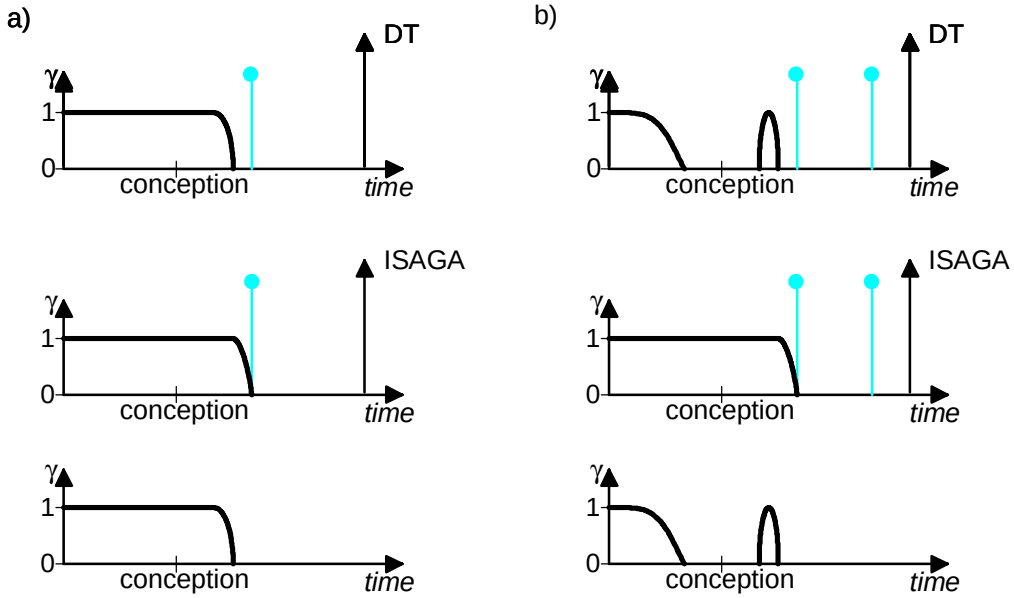


Figure 6.6: A case that remains undecided
a) positive DT and ISAGA may occur with recent and latent infections
b) two consecutive high DTs are compatible with acute (peak titer between samples) and latent infection

6.3.3 Combination of Evidence

Combination of evidence obtained by repeated evaluation of the *same* test on consecutive samples is implicit in the trend detection method itself: the explaining course necessary to derive compatibility must pass through all findings. Evidence obtained from *different* tests is combined in a conjunctive fashion: if one test excludes infection at a certain time, then it overrules other tests even if they suggest a possible infection at that time. The examples illustrated in Figure 6.5 and Figure 6.6 combine evidence this way.

6.4 Acquisition of Fuzzy Courses of Infection

Although typical courses of infection are described in the literature, practical employment of onset determination requires a reliable assessment of *all* practically occurring courses. For this purpose I decided to derive the courses from actual data of evident acute infections documented at the laboratory.

Briefly, the fuzzy courses are constructed by reversing the trend detection process. A clinician was asked to assess possible times of onset of infection by defining a trapezoidal fuzzy set $\tilde{\Omega}_{p_n}$ on the time scale for each of 42 acutely infected cases p_n (with the findings $\langle x_{p_n}[t_m]\rangle$, as shown in Figure 6.7 a). From these assessments a partial fuzzy course $\tilde{x}_{p_n}(t)$ is generated for each case by defining

$$\mu_{\tilde{x}_{p_n}(t)}(x_{p_n}[t_m]) := \mu_{\tilde{\Omega}_{p_n}}(t_m - t) \quad \text{for all } t_m$$

as depicted in Figure 6.7 b). By superimposing all 42 partial definitions, i.e., by letting

$$\mu_{\tilde{x}(t)}(x) := \begin{cases} \max\limits_{p_n} \mu_{\tilde{x}_{p_n}(t)}(x) & \text{for all } x \in V_x \text{ found in at least one acute case} \\ 0 & \text{else} \end{cases}$$

the individual assessments are combined into one general, as indicated in Figure 6.7 c). Note that derivation of the time of onset of any case p_n based on this so-defined $\tilde{x}(t)$ results in a distribution of compatibility over time that includes $\tilde{\Omega}_{p_n}$.

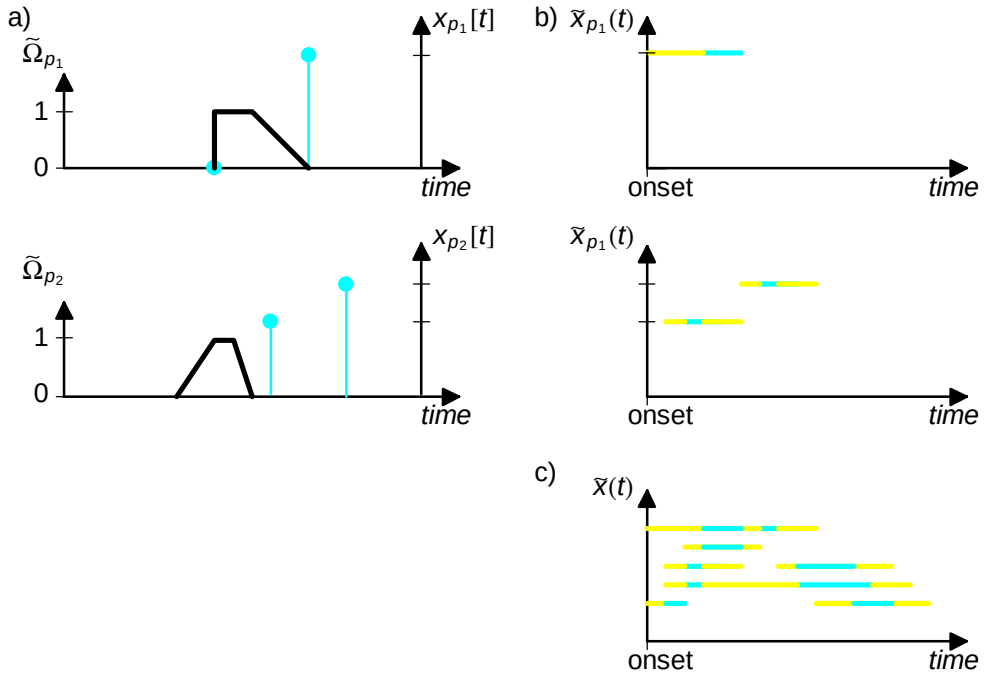


Figure 6.7: Generation of the fuzzy course of infection from findings of a test *x*
a) assessments of possible times of onset for evident acute infections by an expert
b) translation of these assesments to partial fuzzy courses
c) superposition of partial courses

The process of generating a fuzzy course general enough to cover all known courses of infection is completed by replacing $\tilde{x}(t)$ with a fuzzy envelope that presents a continuous specification complemented with serological knowledge that could not be extracted from the learning cases.

Applied to findings from the DT and IgM ISAGA this procedure led to the fuzzy courses shown in Figure 6.8 and Figure 6.9. It must be emphasized that the courses have a subjective component, as they reflect the clinician's conception of possible courses on which he bases his routine diagnosis—because the exact time of onset of infection is only known in very few cases, it is possible that these courses do not comply with reality.[29]

[29] The fact that neither of the courses takes the delayed onset of symptoms (including measurable immune response) into account hints at such a discrepancy.

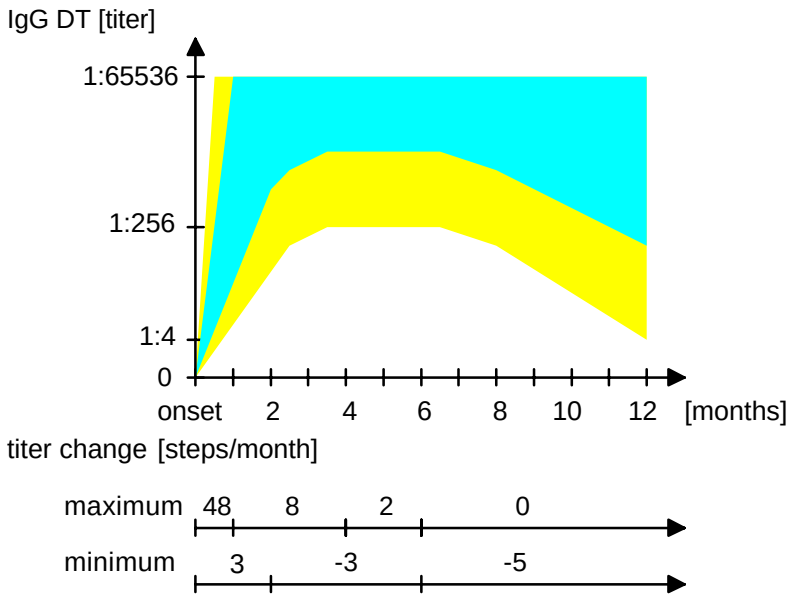


Figure 6.8: Fuzzy course of acute primary infection as observable through IgG DT; the slope of explaining courses is additionally restricted by minimum and maximum titer changes

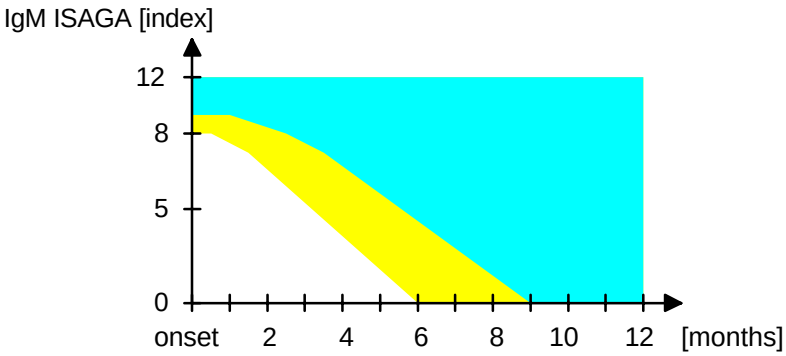


Figure 6.9: Same as Figure 6.8 for IgM ISAGA; the slope is not constrained;
note that the fuzzy course suggests the presence of high IgM immediately after onset of infection

Because of the fuzzy courses' wide spread, sequences of findings suggestive of a falling development are potentially compatible with the rising phase of the course (and vice versa), a problem that is visualized in Figure 6.10. This theoretical consideration was confirmed by a first retrospective evaluation of ONSET which showed that this problem caused 86% of the total misclassifications [Steimann 94d]. ONSET was therefore extended to perform trend detection via restriction of the first derivative of explaining courses. This (non-fuzzy) restriction is also included in Figure 6.8; the determination of possible times of onset based on DT findings thus relies on the joint specification of absolute course and relative change. The results of this combined approach are presented below.

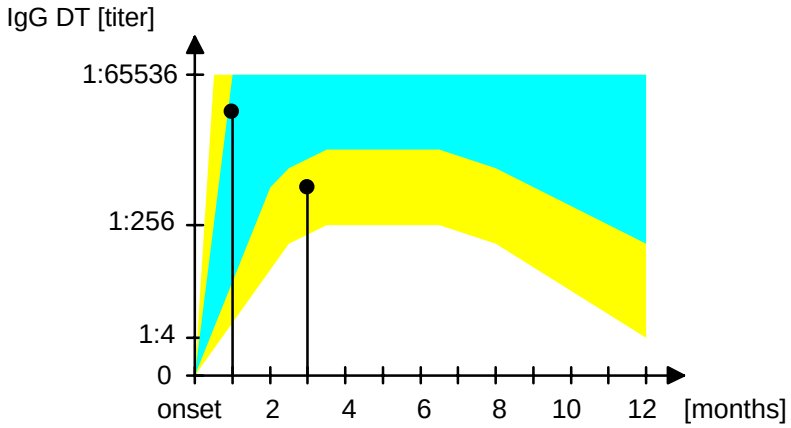


Figure 6.10: Possible misinterpretation of findings due to the wide spread of fuzzy courses

Although an analogous construction of the course of IgG avidity was also included in an earlier version of ONSET, it now uses a modified[30] form of the non-fuzzy criteria published in [Lappalainen 92] as reproduced in Table 6.1. Graphically, these criteria translate to Figure 6.11.

Table 6.1: Interpretation of IgG avidity results

IgG Avidity	Interpretation
≤ 15%	infection acquired within last three months
16–29%	no statement possible
≥ 30%	infection more than six months ago

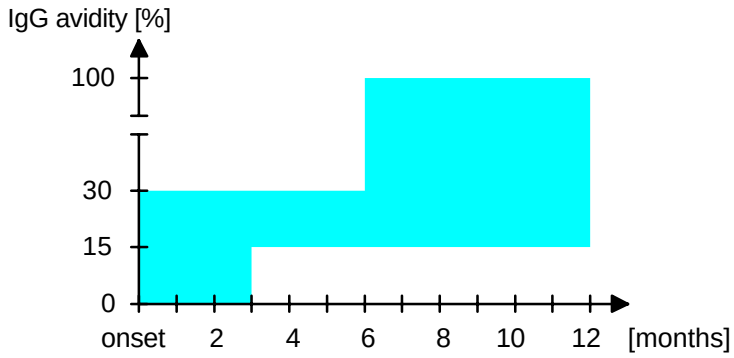


Figure 6.11: Graphic representation of Table 6.1

6.5 Evaluation

Compared to intensive care or operation room monitoring, evaluation of a system for supporting serodiagnosis poses few problems. It must be noted, however, that even if a retrospective evaluation produces promissing results, a true evaluation of its usefulness can only be obtained prospectively.

6.5.1 Retrospective Evaluation

To get a first idea of its diagnostic performance, ONSET was tested retrospectively on 1000 pregnancies with follow-up serology randomly chosen from a toxoplasma database. All cases were diagnosed by ONSET, and the diagnosis was subjected to evaluation. The lack of a readily available objective gold standard leaves only a clinician's judgement to compare ONSET's

[30] Modifications are minor and partly reflect deviating criteria published by the same authors.

diagnoses with. Comparison, however, is not straightforward, as ONSET's diagnosis is an explicit distribution of compatibility of times of onset with the findings, while the clinician's routine diagnosis is usually geared to therapy decision which is inherently binary. As a common denominator, five diagnostic classes are defined, namely *postconceptional*, *undecided*, *preconceptional*, *seronegative*, and *inconsistent data*. ONSET's distributions are translated to these classes according to Table 6.2; findings are regarded to be explainable by a time of onset if the degree of compatibility of these findings with the fuzzy course relative to that time is non-zero.

Table 6.2: Translation of possible times of onset to diagnostic classes

FINDINGS[1] EXPLAINABLE BY TIMES OF ONSET		DIAGNOSIS
BEFORE CONCEPTION	AFTER CONCEPTION	
no	*yes*	*postconceptional*
yes	*yes*	*undecided*
yes	*no*	*preconceptional*
no	*no*	*inconsistent data*
ALL FINDINGS NEGATIVE		*seronegative*

[1] including at least one positive finding

The same cases were then presented to a clinician who was asked to state his diagnosis based on the given findings. He was urged not to question the correctness of individual findings and declare all cases of which he felt that data was erroneous as *inconsistent data*. The only modification that both ONSET and the clinician were allowed to make was to regard an increase in DT by one step (which may be due to measurement error, see below) as constant.

Table 6.3: Performance of ONSET contrasted with that of a clinician

ONSET	CLINICIAN					
	POSTCONCEPTIONAL	UNDECIDED	PRECONCEPTIONAL	SERONEGATIVE	INCONSISTENT DATA	TOTAL
POSTCONCEPTIONAL	3	1	–	–	–	4
UNDECIDED	1	30	3	–	–	34
PRECONCEPTIONAL	–	7	335	–	2	344
SERONEGATIVE	–	–	–	606	–	606
INCONSISTENT DATA	1	–	3	–	8	12
TOTAL	5	38	341	606	10	1.000

The overall accuracy of ONSET on the selected cases is 98.2%. Of the 18 deviating diagnoses 13 were judged as being acceptable by the clinician. Of the remaining five,

- all three cases classified *undecided* by ONSET and *preconceptional* by the clinician had low DT titers (1 : 1024 or 1 : 256) at the end of pregnancy which have also been found in evident postconceptional infections and were thus included in the fuzzy courses,
- one postconceptional infection was considered inconsistent by ONSET because a rapid increase (from 1 : 64 to 1 : 65536 within twelve days, corresponding to a slope of more than twelve steps per month) was not followed by higher titers, an observation that is not compatible with the slope restriction of Figure 6.8 which would require a further titer rise (for practical reasons titration usually ends at 1 : 65536 so that higher concentrations, even if present, cannot be observed), and

- one preconceptional infection was classified as inconsistent by ONSET because the DT decayed slightly faster than tolerated by the slope restriction.

Of the critical misclassifications (agreed by the clinician to be acceptable),

- one case classified *postconceptional* by ONSET and *undecided* by the clinician had a significant increase from 1 : 256 (15^{th} week of gestation) to 1 : 4096 within seven weeks (followed by a titer of 1 : 1024), and
- one case classified *undecided* by ONSET and *postconceptional* by the clinician had a primary high titer (1 : 65536 in the eleventh week of gestation) which is suggestive of, but not proving, postconceptional infection.

Contrasted with the diagnostic performance of ONSET published in [Steimann 94d], its current implementation shows significantly increased congruence with the clinician's diagnoses.[31] Most notably, no false classifications are due to the disregard of relative change, which presented a major deficiency of the earlier version. This result is the more impressive as the fuzzy courses were updated to represent a wider range of acute infections, reflecting more general (and hence less specific) serological knowledge. In numbers, employment of the slope restriction improves ONSET's diagnosis as shown in Table 6.4.

Table 6.4: Impact of slope restriction on ONSET's diagnosis

DIAGNOSIS	WITHOUT SLOPE RESTRICTION		WITH SLOPE RESTRICTION
postconceptional	3	1 →	4
undecided	80	← 44 →	34
preconceptional	303		344
seronegative	606		606
inconsistent data	8		12
total	1.000		1.000

6.5.2 Prospective Evaluation

In the retrospective evaluation, both the clinician and ONSET fully exploited the availability of follow-up serology. In clinical practice, however, a diagnosis must be made after each serum, and this diagnosis is subject to confirmation or change after each follow-up. To demonstrate the predictive value of ONSET at every stage of serodiagnosis, a prospective study paralleling clinical routine would be necessary.

However, a prospective study is a critical and expensive undertaking and therefore hard to implement. The following list of impediments is all but complete.

- For the evaluation to be fair, the information given to both the clinician and the program must be identical. This requirement somehow interferes with clinical routine, as clinicians must strive to produce the best possible diagnosis using *all* information available, even if not in a form suitable for computer processing.
- To increase credibility of the study and to eliminate randomness in the clinician's diagnostic performance, a panel of experts is to be involved.
- A prospective study takes a long time (in this case, at least the duration of a pregnancy) and, accompanying clinical routine, consumes valuable resources.

For these reasons a truly prospective evaluation was not undertaken. Rather, I decided to simulate the prospective diagnostic performance of ONSET in analogy to the simulated on-line evaluation of Chapter 5 by diagnosing the same 1000 cases incrementally, i.e., sample by sample.

[31] The clinician's diagnoses are identical in both evaluations.

Prospective serodiagnosis requires a different framework: here, a seronegative sample must be classified as a potentially postconceptional infection, as subsequent samples may indeed turn positive indicating seroconversion. Only if the sample taken at delivery is also negative, the diagnosis *seronegative* is justified. To make the evaluation more transparent, I excluded all cases retrospectively determined as seronegative; the remaining cases with an initial negative titer were initially classified as *undecided*.

To trace the progress of diagnosis, the number of changes from one diagnostic class to another was determined after each sample. The results presented in Table 6.5 show that a high fraction (55.5%) of all preconceptional infections is classified as *undecided* after the first sample. Differentiation is highest with the second sample: half of all diagnoses made at this stage are a refinement of the previous diagnosis.

Table 6.5: Diagnosis after each sample. The first term of each sum counts the number of cases whose diagnoses remain unchanged on follow-up, while the second counts the ones that change. Numbers on arrows state the number of *undecided* diagnoses that are differentiated to *pre-* or *postconceptional* after the next sample, respectively.

DIAGNOSIS	AFTER 1ST SAMPLE	AFTER 2ND SAMPLE	AFTER 3RD SAMPLE	AFTER 4TH SAMPLE	AFTER 5TH SAMPLE
postconcept.	2+0	4+3	2+0	1+0	
undecided	41+195	36+5	10+2	5+0	1+0
preconcept.	152+2	338+0	48+0	11+0	2+0
inconsistent	2+0	8+0	8+0	2+0	
TOTAL	197+197	386+8	68+2	19+0	3+0

5 (*postconcept.* ← *undecided*); 186 (*undecided* → *preconcept.*); 4 (*undecided* → *preconcept.*); 2 (*undecided* → *preconcept.*)

Table 6.5 does not provide any evidence that ONSET could save therapies. Quite to the contrary, if all initially undecided cases were treated, the treatment rate would be more than tenfold.

The high degree of indecision after the first sample is only partly due to the fact that for the slope restriction to work at least two samples ("paired sera") are required. Many of the initially undecided cases would have been classified *preconceptional* if a (mostly negative) IgM had been available. However, considering the fact that a reliable IgM is not always available and positive IgM results allow no statement about the onset of infection, the results impressively justify the serological necessity of follow-up serology after a first positive sample.

On the other hand, Table 6.5 suggests that 220 (45.3%) of all follow-up examinations of seropositive mothers could have been saved if ONSET's diagnosis were trusted. In particular, the prospective evaluation shows that while a second sample makes a difference in 50% of all cases, the third and all further samples are only seldom useful.

Interestingly, the fraction of *undecided* cases increases with the number of samples taken, which reflects the laboratory's search for evidence that would differentiate these cases.

6.6 Further Improvements

Although the accuracy of ONSET is already very high, theoretical considerations suggest further improvements that allow the system to adopt the reasoning behaviour of a clinician when confronted with nonstandard situations.

6.6.1 Integration of Measurement Error

All measurements are subject to error. In case of the DT, a major source of error is the test method itself, as it is usually performed in steps of fourfold dilution, which implies a rather crude discretization. Small, random fluctuations in an otherwise constant antibody titer for

example can lead to two consecutive measurements deviating by a factor of four, which may be falsely interpreted as a titer change. Maybe because of this inherent randomness, conventional diagnosis relies on what is termed a *significant titer rise*, which refers to a change of two or more steps. The test method inherent discretization error is added to by measurement itself, that is in this case by counting dyed cells through the lenses of a microscope.

Adhering to the fuzzy approach, a simple, yet reasonable, solution to this problem is to replace each measurement by a fuzzy number that represents the range of actual (i.e., non-discretized, true) titers compatible with the discrete observation. An example of such a number is shown in Figure 6.12. It basically allows deviation by factor two from the named value and suggests a continuously decreasing compatibility for values approaching a fourfold deviation, which is the center of the next fuzzy number. Note that these fuzzy sets are different from those employed in fuzzy discrimination in Section 3.6.2: here, fuzzy numbers represent findings, while there they represent categories or classes of findings that are to be treated equally.

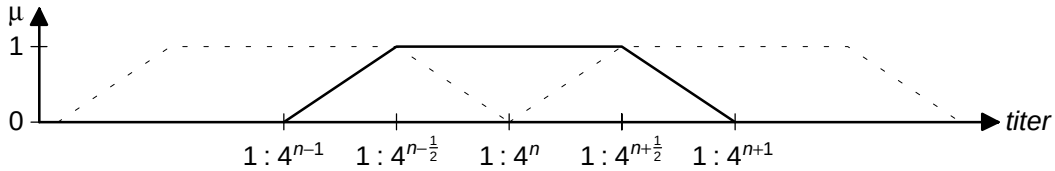


Figure 6.12: Fuzzy number representing an observed titer $\tilde{x}[t_n] = 1 : 4^n$

With the representation of findings the definition of compatibility (Definition 3.3) needs to be extended to sequences of fuzzy samples. Briefly, compatibility is then defined as

$$\gamma(\langle\tilde{x}[t_n]\rangle, \tilde{C}_x, t_\Omega) = \sup_{x_{\tilde{e}}(t)} \min\left(\omega(\tilde{C}_x, x_{\tilde{e}}(t), t_\Omega), \min_{t_n} \mu_{\tilde{x}[t_n]}(x_{\tilde{e}}(t_n))\right) \quad ,$$

where $x_{\tilde{e}}(t)$ is a (crisp) course fuzzily explaining $\langle\tilde{x}[t_n]\rangle$, i.e., going through the fuzzy points $\tilde{x}[t_n]$.

Example

The fuzzy findings $\tilde{x}[t_1] = 1 :1024$ and $\tilde{x}[t_2] = 1 : 4096$ where the fuzzy numbers are defined as in Figure 6.12 are fuzzily explained by the course $x_{\tilde{e}}(t) = 1 : 2048$ with the degree of explanation

$$\min(\mu_{\tilde{x}[t_1]}(1 : 2048), \mu_{\tilde{x}[t_2]}(1 : 2048)) = 1.$$

$x_{\tilde{e}}(t) = 1 : 1024$ cannot explain these findings because $\mu_{\tilde{x}[t_1]}(1 : 1024) = 0$.

◊

6.6.2 Integration of External Findings

Employed test methods vary from laboratory to laboratory. Because different tests measure different (properties of) antibodies, results can generally not be transferred. However, if post-conceptional infection is suspected, all information available should be exploited to make the diagnosis more specific.

The least assumption that can be made is that an external positive sample implies seropositivity of the DT, even if the actual titer remains unknown. In this case, the fuzzy number depicted in Figure 6.13 can represent a general positive finding that proves existing infection. Carefully taking all sources of variability of the external test into account, translation tables can be devised that allow the mapping of external test results to internal fuzzy findings that are easily integrated in the diagnostic process without importing a systematic error.

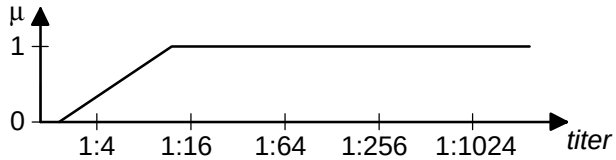


Figure 6.13: Seropositivity expressed by a fuzzy number

6.6.3 Alternative Courses

Evident acute infections are seldom caught in routine screening so that data describing the course of acute infection is very rare. Although I have no evidence for the existence of different courses of infection, inspection of a large toxoplasma screening database showed that acute infection is commonly accompanied by a DT titer of $1:65536$. There are, however, cases of evident acute infections whose measured titer does not exceed $1:1024$. Even if the peak titer is higher (which it probably is), these cases differ significantly from the more frequent ones and it must be suspected that closer examination (i.e., higher sampling rates) will not reveal more similarity. Wide individual variations are also observable in other tests: while IgM usually decays soon after infection, it remains on medium and even high levels for years in some patients. Even if the immunologic reasons to these variations are unclear, they are evident and can thus be reasoned with.

Comprising all observable courses of infection under one fuzzy course yields a very unspecific description and hence unspecific diagnoses. Indeed, as can be shown, reducing the spread in the fuzzy course reduces the number of diagnoses reading *undecided*. However, an increase in specificity should not be traded against sensitivity—if there are different types of courses, then diagnosis should regard them all.

A simple solution to this problem is to derive different diagnoses for different fuzzy courses representing different types of immune response. These diagnoses are then to be regarded as alternatives, and their results can be joined disjunctively to produce the total diagnosis.

Example

Assuming a normal course of infection, the two findings $x[t_1] = 1:1024$ and $x[t_2] = 1:1024$ are diagnosed *preconceptional*. However, a weak immune response assumed, infection may as well be *postconceptional*. Consequently, combined diagnosis must read *undecided*.

◊

While this approach may not seem truly advantageous over the diagnosis based on generalized courses, it can be used to differentiate different types and so assign probabilities (based on relative frequencies of the types) and other features to the diagnosis. Theoretically, it is also capable of excluding courses that develop between a normal and a weak immune response, which makes sense if such courses do not naturally occur. Note that the definition of fuzzy course (Definition 3.4) prevents single courses from encoding disjoint alternative paths.

Chapter 7

Discussion

The presented work approaches the problem of diagnostic monitoring from a signal processing perspective, yet it does not rely on conventional digital filtering theory, but strives to integrate information processing methods from very different disciplines, including artificial intelligence (AI), fuzzy set theory, time series analysis and automata theory. Time is regarded as *the* independent variable and all other observations (including high-level abstractions derived by a monitor) are considered to be functions of time. This perspective strongly influences the elements of monitoring: interval-based temporal reasoning for example as put forward by the AI community is less natural to integrate into this framework than, say, Kalman filtering.

7.1 Fulfilment of Requirements

The requirement specification of Section 2.1 is informal, the fulfilment of requirements thus hard to prove. Nevertheless, the following paragraphs revisit the requirements one by one and check them against the properties of the monitors presented in Chapters 5 and 6.

A valuation of the *generality* of DIAMON-1 would require a formal investigation of its expressive power, which is not pursued in this work. The experiments described in Chapter 5 and 6, however, cover quite different cases of clinical monitoring, so that some generality cannot be denied. *Customizability* of monitors, a property complementary to generality, has been shown in the ARDS monitoring case of Section 5.2.

As pointed out before, the output of monitors presented in Sections 5.2 and 5.3 changes far less frequently than their input. The *temporal information reduction* is primarily taken over by the employed fuzzy automata, which derive stable (defined as changing less frequently than the input) temporal abstractions. Data smoothing, discrimination and trend detection can also lead to information reduction, however, their effect very much depends on properties of the data (compare discussion in Sections 3.4.4 and 3.6.4).

Apart from offering standard interpolation methods for the reconstruction of *missing values*, trend detection methods as defined in Section 3.7 take explicit account of missing values by reasoning with (continuous) courses explaining the findings. The trend detection of DIAMON-1, which is designed for intensive care employment, is opportunistic in that it always assumes the explaining course best matching the trend being sought for, which is purely hypothetical if sampling is sparse. In the sparsely sampled domain of outpatient monitoring, the ONSET program derives a complete set of hypotheses (attributed with a degree reflecting compatibility with the findings) by regarding all possible values to fill the gaps left by missing values.

DIAMON-1 currently offers four *data types*: numeric, degree, symbolic and fuzzy set. Its system's input and output are not restricted to particular data types or combinations thereof: in principle, a system can transform any data type into any other. All systems (and, consequently, all data types) can occur anywhere in the monitoring hierarchy.

Several approaches to *integrating different sampling times* are offered: the one used in the experiments of Chapter 5 synthesizes a sampling sequence as the "least common denominator", i.e., by executing the whole monitoring hierarchy every time a new (set of) samples arrives.

The required *on-line operability* was more than fulfilled by the monitors presented in Chapter 5: all monitors access databases strictly sequentially and produce their output in far less time than granted by the recorded sampling sequence. However, general real-time response is not proven.

The theoretical *openness* is guaranteed by composing a monitor out of elementary systems. The technical openness of DIAMON-1 is granted by its object-oriented design and implementation (all systems are implemented as subclasses of class *System* and respond to the same set of methods, i.e., provide a *uniform interface*). The systems listed as subclasses of class *System* in Figure 4.4 make up the currently implemented *tool box* of DIAMON-1.

The design of a monitor is completely freed of *control flow* issues, the execution of the monitoring hierarchy is totally controlled by the class *Monitor* (Section 4.2). All systems provided with DIAMON-1 have a standard *self-display capability* (requiring no further programming), which is currently restricted to presenting a trend line. A more sophisticated concept of self-explanation is outlined in Section 4.3.

Finally, the *abstraction hierarchy* is provided for, yet cannot be guaranteed, by the framework: although the acyclic coupling of systems necessarily results in a (multiple) hierarchy, the hierarchy need not be one of abstraction. This responsibility is left to the designer of a monitor.

7.2 Other Approaches to Diagnostic Monitoring

Quite obviously, the approach to monitoring taken in this work is not the only possible; other authors have approached the problem from quite different backgrounds. The following sections are an attempt to classify existing monitoring projects according to their main concepts.

7.2.1 Qualitative Simulation and Model-Based Monitoring

Hierarchical monitors such as the ones constructed with DIAMON-1, if properly designed, can be thought of as models of the monitored domain [Factor 90b]: respective monitoring may thus be regarded as model-based. However, model-based is a very general term, and there are other approaches that make more explicit use of models and thus may more justifiably be called model-based.

Typical model-based monitoring systems are characterized by employing the following strategy:

1. predict next measurement based on the current model of the monitored process;
2. read next measurement and compare it with the prediction;
3. if necessary, adjust current model;
4. continue with 1.

Model adjustment may involve tuning of model parameters, or, if a number of applicable models compete in explaining the findings, preference of different hypotheses.

Model-based monitoring is a very old idea: basically the same strategy is pursued in Kalman filtering [Kalman 60] and related approaches [Spall 88]. More recently, numerical and Bayesian process models have been replaced by qualitative ones; their application to patient monitoring is discussed in the following.

Although some monitoring projects employ precise quantitative models (for example, TOPAZ [Kahn 91b] and VENTPLAN [Rutledge 89]), it is generally recognized that such models are hard to specify, particularly if they are to hold for different patients in (slightly) different situations. Several theories of qualitative system descriptions have therefore been devised [Bobrow 84]. These theories can roughly be classified into device- and process-centred: while the former start with a qualitative description of the behaviour of devices as the sole functional objects connected to form a model out of parts, the latter make processes the active parts of a model that determine the appearance of involved objects.

A third, mathematically motivated approach is taken by QSIM [Kuipers 86]. In this model, process-specifying ordinary differential equations are replaced by qualitative ones determining the values of variables over time. QSIM has been made the basis of, for example, MIMIC and NEOANEMIA. (MIMIC is a model-based monitoring system dedicated to the tracking of the observed process's current state while diagnosing possible faults based on explicit fault models [Dvorak 89]. NEOANEMIA [Ironi 90] is a system to diagnose hematologic disorders causing anemia.)

Coiera picks up the QSIM approach, applies it to diagnostic patient monitoring (in particular, to monitoring acid-base disorders), and extends it by a theory of identifying interacting diseases [Coiera 89, 90]. He exploits the fact that the solution of the qualitative differential equations derived by QSIM from initial conditions are qualitative functions over time that can be interpreted as sequences of distinct states representing natural disease histories.

YAQ [Uckun 92a, 92b, 93a], the ontology of the patient model of SIMON, is a process-centred approach to integrated qualitative/quantitative simulation based on Qualitative Process theory [Forbus 84]. It extends other qualitative process formalisms by allowing exogenous quantities, by a hybrid qualitative/quantitative algebra, and by introducing a framework for hypothesized states. Because exogenous quantities (modelling therapeutic actions and other interventions not controlled by YAQ) can alter the model at any time, YAQ is not designed for long-term simulation (as would be required in autonomous therapy planning and control), but for diagnostic monitoring.

In YAQ, diagnoses are derived from associative knowledge, and, alternatively, from a model-based hypothesize-and-test-like reasoning employing short term simulation (prediction) as described above. Surprisingly, the tracking of disease histories is not covered by model-based diagnosis, but assigned to the associative knowledge and reasoning complex, which is inherently unaware of past states. Consequently, explicit history lookups implemented by user-defined functions external to YAQ are required to trigger state transitions [Uckun 93a].

It is frequently stressed that model-based monitoring has significant advantages over other approaches. However, the prediction aspect of model-based monitoring should not be overemphasized[32]: the hypothesize-and-test procedure is only one solution to the general

[32] This is not to say that prediction is not an important issue (for example in therapy planning), but that

problem of identifying (out of a set of possible process models) the ones that (best) explain the current and past findings. Pattern matching and grammar-based approaches to model identification are alternatives to this.

Seen in this light, the claimed hypothesize-and-test character [Dvorak 89, Coiera 90, Uckun 92a] of model based monitoring is more of an implementational issue than the principal nature of the problem. For example, trend detection as presented in Section 3.7 could also be implemented in a hypothesize-and-test manner: starting with an hypothetical time of onset, all subsequent findings are matched with fuzzy ranges predicted for the next finding by all hypothesized trends. Whenever the next sample lies outside a predicted range, the respective trend (a candidate model) is either updated or discarded.

Note that grammar based approaches could also be implemented in a hypothesize-and-test fashion (yet for good reasons are not), and, again, the Kalman filtering approach to trend detection may in fact be viewed as that of a hypothesize-and-test system. A comprehensive review of model-based reasoning in medicine may be found in [Uckun 92c].

7.2.2 Iterative One-Shot Systems and Their Ad Hoc Extensions

Iterative one-shot monitors are systems whose output depends on contemporaneous input only, i.e., which are in essence memoryless. Their behaviour is totally independent of the history they have encountered, meaning that any permutation of the input sequences will lead to an identical permutation of the output sequence. In other words: no information whatsoever is drawn out of sequence.

Iterative one-shot approaches generally exploit the availability of reasoning strategies developed for (time-ignorant) consultation systems. Indeed, such iterative systems may implement the full "vertical" range of desirable operations, yet completely fail to address "horizontal"[33] relations and implications. The temporal accounting of such systems may indeed be viewed as a step back in development since VM [Fagan 84], one of the pioneers in the field extending the rule-based formalism of MYCIN by parameter classifications over time and an explicit notion of state transitions. Examples of such systems are AES-2 [Schecke 91, 92], a knowledge-based approach to intelligent alarms in anaesthesia, and RESAC, a similar system dedicated to the determination of depth of anaesthesia [Greenhow 92].

Such systems frequently avoid disturbing fluctuations in their output (as a direct consequence of fluctuating input and time-ignorance) by averaging input over a period deemed the minimum acceptable change rate of output and evaluating only at the end of each such period.

The iterative one-shot approach has been extended to adopt some minimum history sensitivity by maintaining internal variables whose values are changed in reaction to specified events. While the border to fully developed history sensitivity is certainly blurred, extended iterative one-shot systems are typically characterized by a non-formalized, ad hoc accounting for past observations. In particular, such systems may provide time-valued variables (variables that have a time as their value) and one special variable *now* holding the current time. Characteristically, these systems treat time as a dependent variable (rather than *the* independent variable) and thus as an add-on feature extending time-ignorant systems.

One example of such a system is VIE-VENT [Miksch 93, 94], which, while basically adhering to the one-shot approach, maintains a state variable ruling over the issuing of therapy recommendations. The variable is set to suppress recommendations after therapeutic actions have

prediction is not necessary to identify valid models and thus to diagnose.

[33] The terms *horizontal* and *vertical*,r relate to the traditional orientation of time as the independent variable along the horizontal axis, while contemporaneous observations and their implications are arranged vertically on top of each other.

been taken by the staff, and reset after some predetermined time has elapsed [Miksch 93]. The incentive to do so is to prevent VIE-VENT from rash criticism.

One step further go GANESH and its object-oriented reimplementation NÉOGANESH [Dojat 92a, 92b], two rule-based systems dedicated to weaning patients from ventilator therapy. In accomplishing their tasks, they derive the time for which the patient has been stable, initiate a trial decrease in respiratory support for a predetermined period of time, and assess the patient as weanable once a lower limit of respiratory support has been reached. For this purpose, both systems maintain state and time variables of the above kind. Again, temporal reasoning is implemented in an ad hoc fashion and not founded on some temporal theory; nevertheless, GANESH is one of the very few therapy systems that has been evaluated in closed-loop operation [Dojat 92a], and this with recognizable results.

Admittedly, the introduction of state variables is all that is required to advance from memoryless to history-sensitive systems. Indeed, the introduction of one state variable *q* suffices to extend the time-ignorant tables and rules of Section 3.8 to the finite state machines of Section 3.9, constructs capable of tracking diseases in a history-sensitive manner. Also, all other window-based systems presented in Chapter 3 make do with a finite number of state variables. Seen from this standpoint, monitors constructed with DIAMON-1 may also be regarded as extended one-shot systems—yet, the extension is theoretically founded, and time is not treated as *primus inter pares*, but as *the* independent variable.

7.2.3 Data-Driven Temporal Abstraction and Diagnostic Monitoring

Temporal data abstraction methods are primarily concerned with comprising time-ordered data in the form of homogenous episodes. They typically employ interval-based symbolic data models, i.e., associate temporal intervals with symbolic descriptions.

Temporal data abstraction methods are particularly well-suited to serve as mediators between temporal databases and symbolic temporal reasoners. Because information content preserving abstraction of raw data requires some interpretation, respective systems are sometimes considered diagnostic.

TOPAZ [Kahn 91b] employs a time-oriented data model named ETNET [Kahn 91a] to create its interval-based symbolic abstractions. The abstraction primitives of ETNET are active parts in that their creation can cause creation of other abstractions and the extraction of features characterizing the abstracted interval. TOPAZ maintains an explicit numerical model of the monitored disease, discrepancy of whose predictions with the actual findings initiates the creation of temporal abstractions. The interval-based temporal abstractions derived by TOPAZ are used to generate a natural language summary of the patient history.

RÉSUMÉ [Shahar 92a, 92b] is another system dedicated to deriving abstractions from temporal databases. It defines *point temporal abstraction*, the abstraction of data sampled at (roughly) the same time, *temporal inference* pertaining to the merge and segmentation of intervals, and, most significantly, *temporal interpolation*, the bridging of gaps based on adjacent abstractions and a domain specific maximum-gap function.

RÉSUMÉ comes with a truth maintenance system capable of updating its abstractions when data arrives out of chronological order, namely past its time of occurrence, and when past interpretations are rendered void by new data (compare preceding work in [Russ 86, 90]) [Shahar 92b].

Temporal abstraction methods suffer from their generative nature: given a finite set of abstraction rules (or other mechanisms, the realization does not make a difference) and a potentially infinite number of possible input combinations, the methods can generate infinitely many abstractions. Although, when presented with one particular case, the abstraction will indeed be a

reduction of the offered information, abstractions may vary widely among patients, even if presenting similar findings, and thus require subsequent interpretation. This is in contrast to other approaches to diagnostic monitoring starting with some predetermined notions of temporal patterns (including models) associated with diagnoses and trying to identify these in the input stream. It seems that applying temporal abstraction methods is one step towards diagnostic monitoring, just like feature extraction can be one step to trend detection (compare Section 3.7.1). To arrive at a full-scale diagnosis, however, further processing is necessary.

7.3 Costs and Benefits of Fuzziness

Throughout this thesis, employment of fuzziness has not been presented as a major innovation in the field of diagnostic monitoring, but rather as a natural extension based on the replacement of ordinary with fuzzy sets. Nevertheless, a critical reflection of the adequacy of doing so seems appropriate.

Chapter 2 suggests that fuzziness comes in two flavours: as a degree (replacing the binary truth values *true* and *false* of variables with propositional character) and as a fuzzy set (a compound of elements with degree-valued membership). Both manifestations are employed as means of graduation rather than uncertainty: a patient being in a certain state only to a certain degree means that he is not quite in that state (and this with certainty) rather than that he is possibly (to that degree) (fully) in that state.

Severity indices and scores employed in diverse classification schemes of illness (for example, Murray and Morel score in the definition of the ARDS [Niemer 92]) suggest that fuzziness reflecting graduation is a worthwhile contribution [Kuncheva 92a]. The ongoing discussion of the interpretation of fuzziness as a possibilistic alternative to probability theory does therefore not apply. Independent of this, a possibilistic interpretation of fuzzy numbers in the form of fuzzy courses (Section 3.7.6) and fuzzy findings (Section 6.6.1) is possible, yet not further pursued here.

I maintain that specification of fuzzy sets as employed in fuzzy classification is no more demanding a task than the specification of (non-fuzzy) intervals: agreement on a sharp border in an unsharp physiological domain can in fact be more difficult than the specification of continuous membership functions, where small inadequacies in the specification have only minor impact on the result of classification. Certainly, given contemporary computing power, computational issues of determining (fuzzy) set membership are entirely negligible.

However, it appears that the benefit of fuzziness in binary environments is rather limited, particularly if employed in a history-sensitive consultation system such as ONSET. The following will give rise to this conjecture.

In the context of congenital toxoplasmosis, therapy decisions are inherently binary (to treat or not to treat). Any fuzzy advice giving system must therefore defuzzify its output to yield a binary answer.

A practical defuzzification method in the given context is to require the crossing of a certain threshold α. However, doing so makes fuzziness ineffective, as the same result can be obtained by replacing fuzzy sets with their respective α-cuts and using ordinary set operations [Dubois 80].[34] The role of fuzziness is then reduced to providing parametrical set specifications with a single tuning parameter α. As a matter of fact, the evaluation of ONSET in Section 6.5 was based on the support of fuzzy sets, which is equivalent to α being very close to zero.

[34] Note that this does not hold if other than the standard fuzzy connectives, min and max, are used; particularly not if the operators are compensating.

The favourable outcome of the evaluation can therefore by no means be attributed to the fuzziness of the approach.

By contrast, fuzziness can contribute to highly repetitive environments such as inpatient monitoring, even if the conclusion drawn from its fuzzy assessment of the state of affairs is eventually binary. Much of this contribution is related to the time-varying nature of the problem: while conventional, non-fuzzy abstractions can only change abruptly and thus always come with an element of surprise, the smooth alterations induced by fuzziness comprise early warning and dynamic severity assessment in a natural way.

Admittedly, fuzziness causes more frequent changes in the output of a monitor. This seems particularly at odds with the requirement for stable temporal abstraction in high-frequency sampling domains such as intensive care. Yet it is indicative of tendency and therefore invaluable in domains where early indication of and reaction to developments improves patient care.

The latter opens up the discussion for the steps that follow after diagnostic monitoring: therapy advice (open loop) and therapy control (closed loop).

In fact, fuzzy set theory owes much of its recognition to its success in fuzzy control. Practical problems such as stabilizing an inverted pendulum that are hard or even currently impossible to solve using conventional controllers are astonishingly easily solved by use of fuzzy controllers [Mamdani 93, Ying 94].

Transferred to the medical domain, fuzzy control would result in continuous, gradual therapy suggestion without relying on an explicit numerical model describing the impact of the quantity of a therapeutic action on the controlled variable.

Example

Suppose that positive end-expiratory pressure (PEEP) would normally be altered in steps of 2.5 cm H_2O . A therapy recommendation

$$increasePEEP[t_n] = 0.4$$

could then be interpreted as suggesting an increase by 1 cm H_2O (assuming linear defuzzification).

◊

A prerequisite of fuzzy control is the fuzzy preparation of process variables. In domains as complex as the medical, this involves not only instantaneous fuzzy classifications (as performed by fuzzy discriminators), but also assessments of the temporal development regarding substantial time lags between cause and effect. I therefore suspect that fuzzy trend detection and disease histories will find their place in fuzzy control of clinical therapy problems.

Summarizing, fuzziness contributes to diagnostic monitoring by allowing the assessment of a degree of illness together with (near-)continuous transition of the patient's state indicating speed and direction of change even on a symbolic level. The price being paid for this contribution is negligible.

7.4 Outlook

While taking uncertainty related to vagueness and graduation of medical concepts and their interrelations into account, the presented work leaves all other aspects of uncertainty unaddressed.[35] In particular, the uncertainty arising from imprecise measurement and sparse sampling needs to be taken care of. Although much work has been done by other authors in

[35] This is actually not quite true: ONSET computes all possible times of onset of infection and produces a distribution of compatibility, which may be interpreted as a possibility distribution. However, ONSET is not fully integrated into the monitoring framework, and its approach to dealing with uncertainty arising from sparse sampling is not further discussed here.

the field, their work remains rather isolated, and it is not at all clear how different sources of uncertainty are to be integrated. For example, the degree of compatibility of an imprecise measurement (e.g. represented by a probability distribution) with a linguistic concept (e.g. represented by a fuzzy set) is nontrivial to determine. Although the necessary theoretic considerations are largely independent of the representation of uncertainty, the combination of fuzziness with probability imposes additional complexity. Before additional systems taking account of other sources of uncertainty are introduced to DIAMON-1, this uncertainty needs to be soundly incorporated in the monitoring framework.

Trend detection in its current form is very basic: in most applications, it requires careful preprocessing to remove outliers and disturbing fluctuations. Intuitively, one would suspect that the performance of trend detection could be improved by using compensating fuzzy operators to combine the evidence obtained from different samples. Doing so would allow occasional greater deviations from the fuzzy course to be compensated by a majority of good fits, which would probably be closer to human argumentation in classifying and interpreting courses of variables.

The modelling capabilities of DIAMON-1 monitors are currently rather weakly developed. Fuzzy courses employed for trend detection represent patterns rather than models (although the distinction may be viewed as arbitrary, see above), and finite state machines modelling natural disease histories are at the low end of the scale of model-based reasoning. Work in the field of qualitative simulation applied to monitoring problems (for example, [Coiera 89, Uckun 92a]), but also Kalman filtering and adaptive forecasting techniques based on more complex physiological models show the way ahead. Considering the fact that number- and signal-to-symbol conversion (necessary for qualitative simulation) within DIAMON-1 are fuzzy, fuzzy simulation seems a natural extension that could overcome some of the deficiencies of qualitative simulation. Accordingly, the potential role of fuzzy Kalman filtering in diagnostic monitoring tasks seems worth investigation.

Chapter 8

Conclusion

With DIAMON-1 and ONSET, two programs have been presented that perform diagnostic interpretation of clinical time series. Although operating in different application domains with different characteristics (DIAMON-1 interprets densely sampled data from intensive care, while ONSET is dedicated to the interpretation of sparse sequences of serological findings), they both are embedded in the same theoretical framework of clinical monitoring and, in particular, they both share the same approach to trend detection.

While aspects of medical vagueness and graduation have naturally been integrated into the monitoring framework by means of fuzzy set theory, much work needs to be done to incorporate uncertainty arising from imprecise measurement and partial ignorance as resulting from sparse sampling.

Diagnostic monitoring of clinical time series is a highly general task. Any practical solution to a particular monitoring problem, if using the framework and the components presented in this thesis, requires the adaptation of a multitude of parameters. Because of clinical factors, the configuration of an individual monitor for a particular monitoring case is currently impractical, the on-line employment of monitors of the described kind thus still a vision. Higher utilization can be expected from areas where monitoring follows standard procedures, as in (uncomplicated) weaning from the ventilator and mass screening programs.

References

[Ades 91] AE Ades (1991) "Evaluating the sensitivity and predictive value of tests of recent infection: toxoplasmosis in pregnancy" *Epidemiol Infect* 107, 527–535

[Adlassnig 82] KP Adlassnig: "A survey on medical diagnosis and fuzzy subsets", in: M Gupta, E Sanchez (eds.) *Fuzzy Information and Decision Processes*, North-Holland, Amsterdam 1982, 203–217

[Adlassnig 86] KP Adlassnig, G Kolarz (1986) "Representation and semiautomatic acquisition of medical knowledge in CADIAG-1 and CADIAG-2", *Computers and Biomedical Research* 19, 63–79

[Adlassnig 88] KP Adlassnig: "Uniform representation of vagueness and imprecision in patient's medical findings using fuzzy sets" in: *Proc. EMCSR 88*, Kluwer Academic Publishers, Dordrecht 1988, 685–692

[Aho 86] AV Aho, R Sethi, JD Ullman: *Compilers: Principles, Techniques, and Tools*, Adisson Wesley, Reading 1986

[Albridge 88] KM Albridge, J Standish, JF Fries (1988) "Hierarchical time-oriented approaches to missing data inference", *Computers and Biomedical Research* 21, 349–366

[Allen 83] R Allen (1983) "Time series methods in the monitoring of intracranial pressure part I: problems, suggestions for a monitoring scheme and review of appropriate techniques", *J Biomed Eng* 5, 5–17

[Allen 84] JF Allen (1984) "Towards a general theory of action and time", *Artificial Intelligence* 23, 123–154

[Avent 87] RK Avent, JD Charlton, HT Nagle, RN Johnson (1987) "A Monte Carlo technique for signal level detection in implanted intracranial pressure monitoring", *Annals of Biomedical Enginnering* 15, 79–89

[Avent 90] RK Avent, JD Charlton (1990) "A critical review of trend-detection methodologies for biomedical monitoring systems", *Critical Reviews in Biomedical Engineering* 17(6), 621–659

[Aspöck 92] H Aspöck, A Pollak (1992) "Prevention of toxoplasmosis by serological screening of pregnant women in Austria", *Scandinavian Journal of Infectious Diseases: Supplement* 84, 32–38

[Barro 90] S Barro, R Ruiz, J Mira (1990) "Fuzzy beat labeling for intelligent arrythmia monitoring", *Comp Biomed Res* 23, 240–258

[Barro 91] S Barro, R Ruiz, J Presedo, J Mira (1991) "Grammatic representation of beat sequences for fuzzy arrythmia diagnosis", *Int J Biomed Comput* 27, 245–259

[Bentley 88] JL Bentley: *More Programming Pearls*, Adisson Wesley 1988

[Berzuini 92] C Berzuini, R Bellazi, S Quaglini, DJ Spiegelhalter (1992) "Bayesian networks for patient monitoring", *Artificial Intelligence in Medicine* 4, 243–260

[Blom 85] JA Blom, JAF de Ruyter, N Saranummi, JEW Beneken: "Detection of trends in monitored variables", in: Carson ER, Cramp DG (eds.) *Computers and Control in Clinical Medicine*, Plenum Press, New York 1985, 153–174

[Bobrow 84] DG Bobrow (1984) "Qualitative reasoning about physical systems: an introduction", *Artificial Intelligence* 24, 1–5

[Challis 90] RE Challis, RI Kitney (1990) "Biomedical signal processing (in four parts). Part 1: time domain methods", *Medical & Biological Engineering and Computing* 28, 509–524

[Charniak 91] E Charniak (1991) "Bayesian networks without tears", *The AI Magazine* 12(4), 50–63

[Cheung 90] JTY Cheung, G Stephanopoulos (1990) "Representation of process trends—part II. The problem of scale and qualitative scaling", *Computers Chem Eng* 14, 511–539

[Cohn 90] AI Cohn, S Rosenbaum, M Factor, PL Miller (1990) "DYNASCENE: an approach to computer-based intelligent cardiovascular monitoring using sequential clinical 'Scenes'", *Methods of Information in Medicine* 29, 122–131

[Cohn 91] AI Cohn: "Automated patient-specific data interpretation in an intelligent cardiovascular monitor", *Annual International Conference of the IEEE EMBS* 13(3), 1991, 1312–1314

[Coiera 89] EW Coiera: *Reasoning with Qualitative Disease Histories for Diagnostic Patient Monitoring*, PhD Thesis, School of Electrical Engineering and Computer Science, University of New South Wales 1989

[Coiera 90] EW Coiera (1990) "Monitoring diseases with empirical and model-generated histories", *Artificial Intelligence in Medicine* 2, 135–147

[Coiera 91] EW Coiera (1991) "Incorporating user and dialog models into the interface design of an intelligent patient monitor", *Medical Informatics* 16(4), 331–346

[Coiera 93] E Coiera (1993) "Editorial: Intelligent monitoring and control of physiological systems", *Artificial Intelligence in Medicine* 5(1), 1–8

[Cousins 91] SB Cousins, MG Kahn (1991) "The visual display of temporal information", *Artificial Intelligence in Medicine* 3(6), 341–357

[Dawant 91] BM Dawant, BH Jansen (1991) "Coupling numerical and symbolic methods for signal interpretation", *IEEE Transactions on Systems, Man, and Cybernetics* 21(1), 115–124

[Dawant 94] BM Dawant, EJ Manders, DP Lindstrom: "Model-based data processing, validation, and abstraction for intelligent real-time patient monitoring and control" in: *Working Papers of the AAAI 1994 Spring Symposium on Artificial Intelligence in Medicine*, AAAI Technical Report, AAAI, Menlo Park 1994

[Dean 88] T Dean, K Kanazawa (1988) "Probabilistic temporal reasoning", in: *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-1988*, 524–528

[Desmonts 85] G Desmonts, J Couvreur: "Toxoplasmosis" in: RB Conn (ed): *Current Diagnosis* 7, W.B. Saunders Company, Philadelphia 1985

[Dojat 92a] M Dojat, L Brochard, F Lemaire, A Harf (1992) "A knowledge-based system for assisted ventilation of patients in intensive care units", *International Journal of Clinical Monitoring and Computing* 9, 239–250

[Dojat 92b] M Dojat, F Pachet: "Representation of medical expertise using the Smalltalk environment. Putting a prototype to work.", in: G. Heeg et al. (eds) *Technology of Object-Oriented Languages and Systems TOOLS 7*, Prentice Hall 1992, 379–389

[Dougherty 88] ER Dougherty, CR Giardina: *Mathematical Methods for AI and Autonomous Systems*, Prentice Hall, Englewood Cliffs 1988

[Doyle 89] RJ Doyle, SM Sellers, DJ Atkinson: "A focused, context-sensitive approach to monitoring", in: *Proceedings of the 11th International Joint Conference on Artificial Intelligence IJCAI-89*, Morgan Kaufmann Publishers 1989, 1231–1237

[Doyle 94] RJ Doyle: "Towards combined attention focusing and anomaly detection in medical monitoring", in: *Working Papers of the AAAI 1994 Spring Symposium on Artificial Intelligence in Medicine*, AAAI Technical Report, AAAI, Menlo Park 1994

[Drakopoulos 93] J Drakopoulos, B Hayes-Roth: *A Sigmoidal Based Fuzzy Pattern Recognition System for Multi-Variate Time-Dependent Patterns*, Knowledge Systems Laboratory, Stanford University, Stanford 1993 (current status unknown)

[Dubois 80] D Dubois, H Prade: *Fuzzy Sets and Systems*, Academic Press, New York 1980

[Dubois 93] D Dubois, H Prade, RR Yager, R.R. (eds.): *Readings in Fuzzy Sets for Intelligent Systems*, Morgan Kaufmann, San Mateo 1993

[Dvorak 89] D Dvorak, B Kuipers: "Model-based monitoring of dynamic systems", in: *Proceedings of the 11th International Joint Conference on Artificial Intelligence IJCAI-89*, Morgan Kaufmann Publishers 1989, 1238–1243

[Factor 89] M Factor, DF Sittig, AI Cohn, DH Gelernter, PL Miller, S Rosenbaum: "A parallel software architecture for building intelligent medical monitors", *Proceedings of the Annual Symposium on Computer Application in Medical Care SCAMC-89*, Washington 1989, 11–16

[Factor 90] M Factor: *The Process Trellis Software Architecture for Parallel, Real-Time Monitors*, PhD Thesis, Department of Computer Science, Yale University 1990

[Factor 92] M Factor, DH Gelernter, DF Sittig (1992) "The multi-trellis software architecture and the Intelligent Cardiovascular Monitor", *Methods of Information in Medicine* 31, 44–55

[Fagan 84] LM Fagan, JC Kunz, EA Feigenbaum, JJ Osborn: "Extensions to the rule-based formalism for a monitoring task", in: BG Buchanan, EH Shortliffe (eds.) *Rule-Based Expert Systems*, Adisson-Wesley, Reading, MA 1984, 397–423

[Forbus 84] KD Forbus (1984) "Qualitative process theory", *Artificial Intelligence* 24, 85–168

[Fröberg 85] CE Fröberg: *Numerical Mathematics—Theory and Computer Applications*, Benjamin/Cummings, Menlo Park 1985

[Gallagher 81] NC Gallagher, GL Wise (1981) "A theoretical analysis of the properties of median filters", *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29(6), 1136–1141

[Goldberg 83] A Goldberg, D Robson: *Smalltalk-80: The Language and its Implementation*, Addison-Wesley, Reading 1983

[Gordon 86] K Gordon (1986) "The multi-state Kalman Filter in medical monitoring", *Computer Methods and Programs in Biomedicine* 23, 147–154

[Gordon 88] K Gordon, AFM Smith: "Modelling and monitoring biomedical time series", in: JC Spall (ed) *Bayesian Analysis of Time Series and Dynamic Models*, MArcel Dekker, New York 1988

[Greenhow 92] SG Greenhow, DA Linkens, AJ Asbury (1992) "Development of an Expert System Advisor for Anaesthetic Control", *Computer Methods and Programs in Biomedicine* 37, 215–229

[Hämäläinen 94] M Hämäläinen, T Lipping, Y Neuvo (1994) "Trends in nonlinear signal processing", *Methods of Information in Medicine* 33, 4–9

[Haimowitz 93a] IJ Haimowitz, IS Kohane: "Automated trend detection with alternate temporal hypotheses", in: *Proceedings of the 13th International Joint Conference on Artificial Intelligence IJCAI-93*, Chamberey 1993, 146–151

[Haimowitz 93b] IJ Haimowitz, IS Kohane: "An epistemology for clinically significant trends", in: *Proceedings of the 11th National Conference on Artificial Intelligence AAAI-93*, Washington D.C. 1993, 176–181

[Haimowitz 94] IJ Haimowitz: *Knowledge-Based Trend Detection and Diagnosis*, PhD Thesis, Massachusetts Institute of Technology, Cambridge 1994

[Hayes-Roth 89] B Hayes-Roth, R Washington, R Hewett, M Hewett, A Seiver "Intelligent monitoring and control", *Proceedings of the International Joint Conference on Artificial Intelligenc IJCAI-89*, 1989, 243–249

[Hayes-Roth 92] B Hayes-Roth, R Washington, D Ash, M Hewett, A Collinot, A Vina, A Seiver (1992) "Guardian: a prototype intelligent agent for intensive-care monitoring", *Artificial Intelligence in Medicine* 4(2), 165–185

[Heckerman 92] DE Heckerman, EH Shortliffe (1992) "From certainty factors to believe networks", *Artificial Intelligence in Medicine* 4(1), 35–52

[Ironi 90] L Ironi, M Stefanelli, G Lanzola (1990) "Qualitative models in medical diagnosis", *Artificial Intelligence in Medicine* 2(2), 85–101

[Juhola 91] M Juhola, T Grönfors (1991) "A scheme of inference of regular grammars for the syntactic pattern recognition of saccadic eye movements", *Artificial Intelligence in Medicine* 3(2), 87–94

[Kahn 91a] MG Kahn (1991) "Extensions to the time-oriented database model to support temporal reasoning in medical expert systems", *Methods of Information in Medicine* 30(1), 4–14

[Kahn 91b] MG Kahn (1991) "Combining physiological models and symbolic methods to interpret time-varying patient data", *Methods of Information in Medicine* 30(3), 167–178

[Kahn 91c] MG Kahn (1991) "Modelling time in medical decision support programs", *Medical Decision Making* 11, 249–264

[Kalman 60] RE Kalman (1960) "A new approach to linear filtering and prediction problems", *Journal of Basic Engineering* 82(1), 35–45

[Komaroff 79] AL Komaroff (1979) "The variability and inaccuarcy of medical data", *Proceedings of the IEEE* 67(9), 1196–1207

[Konstantinov 91] KB Konstantinov, T Yoshida (1991) "A knowledge-based pattern recognition approach for real-time diagnosis and control of fermentation processes as variable structure plants", *IEEE Transactions on Systems, Man, and Cybernetics* 21(4), 908–914

[Konstantinov 92] KB Konstantinov, T Yoshida (1992) "Real-time qualitative analysis of the temporal shapes of (bio)process variables", *AIChE Journal* 38(11), 1703–1715

[Koski 92] EMJ Koski, A Mäkivirta, T Sukuvaara, A Kari (1992) "Development of an expert system for haemodynamic monitoring: computerized symbolization of on-line monitoring data", *International Journal of Clinical Monitoring and Computing* 8, 289–293

[Kuipers 86] B Kuipers (1986) "Qualitative simulation", *Artificial Intelligence* 29(3), 289–388

[Kuncheva 92] LI Kuncheva (1992) "Additional criteria in fuzzy two-level pattern recognition", *Fuzzy Sets and Systems* 47, 281–291

[Lappalainen 92] M Lappalainen, P Koskela, M Koskiniemi, P Ämmälä, V Hiilesmaa, et al. (1993) "Toxoplasmosis acquired during pregnancy: improved serodiagnosis based on avidity of IgG", *Journal of Infectious Diseases* 167, 691–697

[Lee 88] EA Lee, DG Messerschmitt: *Digital Communication*, Kluwer Academic Publishers, Boston 1988

[Maiers 85] JE Maiers: "Fuzzy set theory and medicine: the first twenty years and beyond", in: *Proc of the 9th Ann Symp Computer Appl Med Care* 1985, 325–329

[Mamdani 93] EH Mamdani: "Twenty years of fuzzy control: experiences gained and lessons learnt", in: *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, San Francisco, CA 1993, 339–344

[McCabe 83] RE McCabe, JS Remington (1983) "The diagnosis and treatment of toxoplasmosis", *Eur J Clin Microbiol* 2(2), 95–104

[McDermott 82] DV McDermott (1982) "A temporal logic for reasoning about processes and plans", *Cognitive Science* 6, 101–155

[Miksch 93] S Miksch, W Horn, C Popow, F Paky: *VIE-VENT: Knowledge-Based Monitoring and Therapy Planning of the Artificial Ventilation of Newborn Infants*, Technical Report 93-18, Austrian Research Institute for Artifical Intelligence, Vienna 1993

[Miksch 94] S Miksch, W Horn, C Popow, F Paky: "Intensive care monitoring and therapy planning for newborns", in: *Working Papers of the AAAI 1994 Spring Symposium on Artificial Intelligence in Medicine*, AAAI Technical Report, AAAI, Menlo Park 1994

[Miller 92] AS Miller, BH Blott, TK Hames (1992) "Review of neural network applications in medical imaging and signal processing", *Med & Biol Eng & Comput* 30, 449–464

[Nelson 94] KS Nelson, GD Hadden: "Real-time feature recognition in medical data", in: *Working Papers of the AAAI 1994 Spring Symposium on Artificial Intelligence in Medicine*, AAAI Technical Report, AAAI, Menlo Park 1994

[Nii 82] HP Nii, EA Feigenbaum, JJ Anton, AJ Rockmore (1982) "Signal-to-symbol transformation: HASP/SIAP case study", *The AI Magazine* 3, 23–35

[Nökel 89] K Nökel (1989) "Temporal matching: recognizing dynamic situations from discrete measurements", in: *Proceedings of the 11th International Joint Conference on Artificial Intelligence IJCAI-89*, 1255–1260

[Oppenheim 89] AV Oppenheim, RW Schafer: *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs 1989

[Penczek 87] P Penczek, W Grochulski, J Grzyb, M Kowalczyk (1987) "The use of a multi-chanel kalman filter algorithm in structural analysis of the epileptic EEG", *Int J Bio-Medical Computing* 20, 135–151

[Russ 86] TA Russ: "A system for using time dependent data in patient management", in: *Proceedings of the 5th Conference on Medical Informatics (MEDINFO 86)*, 165–169, North-Holland, Amsterdam 1986

[Russ 90] TA Russ (1990) "Using hindsight in medical decision making", *Computer Methods and Programs in Biomedecine* 32, 81–90

[Rutledge 90a] G Rutledge, G Thomsen, I Beinlich, B Farr, M Kahn, L Sheiner, L Fagan: "VentPlan: an architecture for combining qualitative and quantitative computation", in: *Proc. AAAI Spring Symposium AI in Medicine*, Stanford 1990, 159–160

[Rutledge 90b] GW Rutledge, SK Andersen, JX Polaschek, LM Fagan (1990) "A belief network model for interpretation of ICU data", in: *Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care* 1990, 785–789

[Rutledge 93] GW Rutledge, GE Thomsen, BR Farr, MA Tovar, JX Polascheck, IA Beinlich, LB Sheiner, LM Fagan (1993) "The design and implementation of a ventilator-management advisor", *Artificial Intelligence in Medicine* 5(1), 67–82

[Sadegh-Zadeh 94] K Sadegh-Zadeh (1994) "Fundamentals of clinical methodology: 1. Differential indication", *Artifical Intelligence in Medicine* 6, 83–102

[Schecke 91] T Schecke, G Rau, HJ Popp, H Käsmacher, G Kalff, HJ Zimmermann (1991) "A knowledge-based approach to intelligent alarms in anasthesia", *IEEE Engineering in Medicine and Biology Magazine* 40(4), 38–43

[Schecke 92] T Schecke, M Langen, HJ Popp, G Rau, H Käsmacher, G Kalff (1992) "Knowledge-based decision support for patient monitoring in cardioanesthesia", *International Journal of Clinical Monitoring and Computing* 9, 1–11

[Schmidt 87] RF Schmidt, G Thews, (eds.): *Physiologie des Menschen (Human Physiology)*, Springer, Berlin 1987

[Shahar 92a] Y Shahar, SW Tu, MA Musen (1992) "Knowledge acquisition for temporal abstraction mechanisms", *Knowledge Acquisition* 4, 217–235

[Shahar 92b] Y Shahar, MA Musen: *RÉSUMÉ: A Temporal-Abstraction System for Patient Monitoring*, Report KSL-92-84, Knowledge Systems Laboratory, Stanford 1992

[Shoham 87] Y Shoham (1987) "Temporal logics in AI: semantical and ontological considerations", *Artificial Intelligence* 33, 89–104

[Shortliffe 76] EH Shortliffe: *Computer-Based Medical Consultation: MYCIN*, Elsevier, New York 1976

[Sittig 89] DF Sittig, NL Pace, RM Gardner, E Beck, AH Morris (1989) "Implementation of a computerized patient advice system using the HELP clinical information system", *Computers and Biomedical Research* 22, 474–487

[Sittig 90a] DF Sittig, RM Gardner, AH Morris, CJ Wallace (1990) "Cinical evaluation of computer-based respiratory care algorithms", *International Journal of Clinical Monitoring and Computing* 7, 177–185

[Sittig 90b] DF Sittig, M Factor (1990) "Physiologic trend detection and artifact rejection: a parallel implementation of a multi-state Kalman filtering algorithm", *Computer Methods and Programs in Biomedicine* 31, 1–10

[Sittig 92a] DF Sittig, KH Cheung (1992) "A parallel implementation of a multi-state Kalman filtering algorithm to detect ECG arrhytmias", *International Journal of Clinical Monitoring and Computing* 9 , 13–22

[Sittig 92b] DF Sittig, KH Cheung, L Berman (1992) "Fuzzy classification of hemodynamic trends and artifacts: experiments with the heart rate", *International Journal of Clinical Monitoring and Computing* 9, 251–257

[Spall 88] JC Spall (1988) "An overview of key developments in dynamic modeling and estimation", in JC Spall: *Bayesian Analysis of Time Series and Dynamic Models*, Marcel Dekker, New York 1988, xv-xxvii

[Steimann 94a] F Steimann, KP Adlassnig: "Clinical Monitoring with fuzzy automata", *Fuzzy Sets and Systems* 61(1), 37–42

[Steimann 94b] F Steimann, KP Adlassnig: "A Fuzzy Medical Data Model" in: R Trappl (ed) *Proceedings of the 12th European Meeting on Cybernetics and Systems Research*, World Scientific, Singapore 1994, Vol. 1, 271–278

[Steimann 94c] F Steimann, KP Adlassnig: "Two-stage interpretation of ICU data based on fuzzy sets" in: *Working Papers of the AAAI 1994 Spring Symposium on Artificial Intelligence in Medicine*, AAAI Technical Report, AAAI, Menlo Park 1994

[Steimann 94d] F Steimann, M Hayde, B Panzenböck, KP Adlassnig, A Pollak (1994) "Fuzzy support for serodiagnosis: the onset program", *IEEE Engineering in Medicine and Biology Magazine* 13(5), 705–709

[Steimann 94e] F Steimann: *Diagnostic Monitoring based on Fuzzy State Transitions*, Technical Report 1-94, Department of Medical Computer Sciences, University of Vienna, Austria 1994

[Steimann 95] F Steimann: "A case against logic", in: *Proceedings of the MedInfo ´95*, to appear

[Stroustrup 86] B Stroustrup: *The C++ Programming Language*, Addison-Wesley, Menlo Park 1986

[Uckun 92a] S Uckun: *An Ontology for Model-Based Reasoning in Physiological Domains*, PhD Thesis, Vanderbilt University, Nashville, Tennessee 1992

[Uckun 92b] S Uckun, BM Dawant (1992) "Qualitative modeling as a paradigm for diagnosis and prediction in critical care environments", *Artificial Intelligence in Medicine* 4(2), 127–144

[Uckun 92c] S Uckun (1992) "Model-based reasoning in biomedicine", *CRC Critical Reviews in Biomedical Engineering* 19(4), 261–292

[Uckun 93a] S Uckun, BM Dawant, DP Lindstrom (1993) "Model-based reasoning in intensive care monitoring: the YAQ approach", *Artificial Intelligence in Medicine* 5(1): 31–48

[Uckun 93b] S Uckun: *Intelligent Systems in Patient Monitoring and Therapy Management*, Rep. No. KSL 93-32, Knowledge Systems Laboratory, Stanford University, Stanford, CA 1993

[van der Aa 90] JJ van der Aa: *Intelligent Alarms in Anesthesia, a Real-Time Expert System Application*, Doctoral Thesis, Eindhoven University of Technology, Division of Medical Electrical Engineering, NL 1990

[Ying 94] H Ying, L Sheppard (1994) "Regulating mean arterial pressure in postsurgical cardiac patients", *IEEE Engineering in Medicine and Biology Magazine* 13(5), 671–677

[Zadeh 65] LA Zadeh (1965) "Fuzzy sets", *Information and Control* 8, 338–353

[Zadeh 68] LA Zadeh (1968) "Fuzzy algorithms", *Information and Control* 12, 94–102

[Zadeh 73] LA Zadeh (1973) "Outline of a new approach to the analysis of complex systems and decision processes", *IEEE Transactions on Systems, Man, and Cybernetics* 3(1), 28–44