Real-Time Feature Recognition in Medical Data

Kyle S. Nelson
nelson_kyle@htc.honeywell.com

George D. Hadden
hadden_george@htc.honeywell.com

Honeywell Technolgy Center
3660 Technology Dr.
Minneapolis, MN 55418

1.0 Introduction

Honeywell has developed an approach, State-Based Feature Recognition (SBFR), to automatically recognize trends and predict failures using parametric data. We have applied this technique to the health monitoring of space vehicle systems including the Orbital Maneuvering System/Reaction Control System for the Space Shuttle, the Attitude Determination and Control System for Space Station Freedom, and Electro-Mechanical Actuators to be used on future space vehicles.

Our success in this domain has led us to explore applications in other domains, such as power plant monitoring and building control. Applying SBFR to the medical domain is a recent exploration and this paper represents the first foray into that domain. It will introduce SBFR, review some similar work, describe its current status, and discuss applications of the technology to the medical domain.

2.0 SBFR Description

SBFR is "feature-based" in that each of its state machines operates by recognizing patterns in its input data that it has been designed to look at as a single item, a "feature". Features to be recognized using SBFR are represented as finite state machines. The following refers to Figure 1 which illustrates an example of a generic state machine. The state machine is made up of a set of states, {S1,...,S4} and transitions between those states, {T1...T6}. Each state in the machine represents a stage in the identification of the feature. Each state has associated with it a set of transitions. Each transition may have one or more actions associated with it, to be performed whenever that transition is taken.

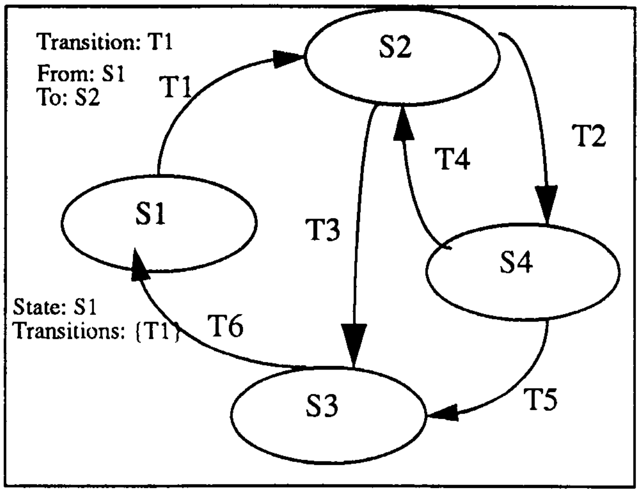

Figure 1 General State Machine

Each transition has a condition which defines when the transition may be traversed. When the machine's current state is the transition's "from" state and the condition for a particular transition is true then the machine traverses the transition.

The general meaning of a feature is captured by the machine's states and transitions, while the specifics of the feature (i.e. the exact data which causes the state machine to move from one state to another) are captured by the conditions. This affords a separation between the general definition of a feature and its real world implementation, allowing a general machine to be instantiated in many different contexts.

3.0 Trend Recognition

One important application of SBFR is the difficult, but important, task of trend recognition. SBFR provides a representation scheme encompassing an intuitive method for describing the trend in human terms and an easy way to translate it into computer terms.

The machines to recognize trends, called trend machines, are built on the principles discussed in section 2.0. The initial state is the default starting state of the machine and represents no trend being detected, i.e. no evidence of the trend has been seen. As the machine enters different states, information can be stored in local variables and in the status registers. When the machine detects the trend, the pertinent information, along with notification of the trend, is written to globally accessible status registers, essentially writing information to a distributed blackboard. Any number of trend machines can be associated with a parameter and any number of parameters can be associated with a single machine.

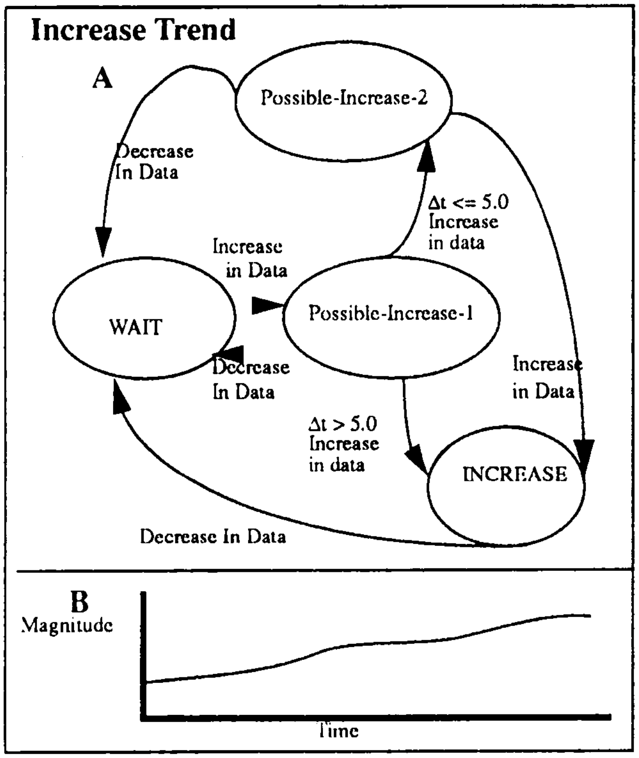

Figure 2 Simple Increasing Trend

Trend recognition machines are implemented hierarchically. Any number of layers may be built up where the output of lower-level machines serves as input to the next level up. This layering enables complex trends to be broken down into components consisting of simpler trends. The following examples will illustrate trend machines at the first and second level of this hierarchy.

Figure 2A shows a trend machine used to recognize a simple increase trend, like the one illustrated in Figure 2B. In this case, an increase is defined as being two jumps in the data occurring greater than 5 time units apart, or three jumps in the data if the first two jumps happen in less than or equal to 5 time units. The reason for the time limit is to differentiate between an increase and a spike. Any time a decrease in the data is seen, the machine will transition back to the initial state. When the machine enters the Increase state an increasing trend has been identified.

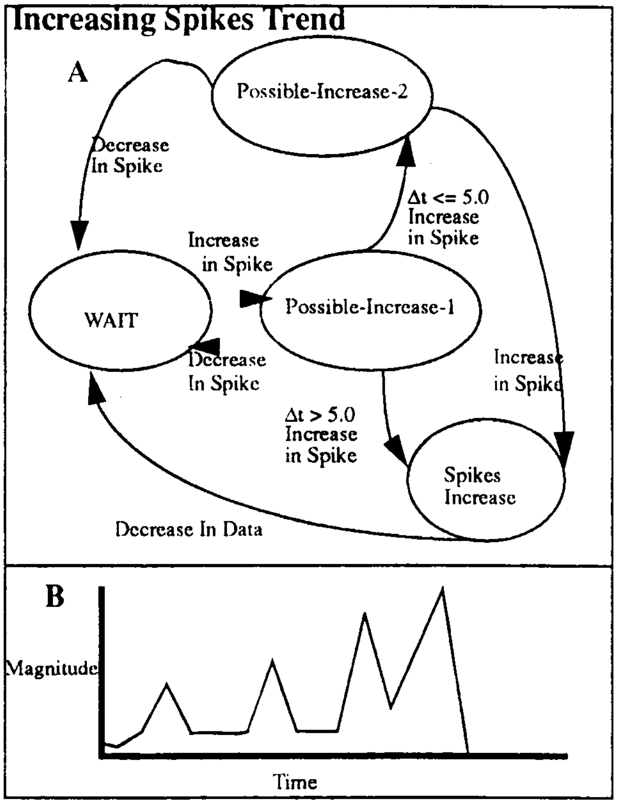

Figure 3 Increasing Spike Trend

Figure 3 shows a trend machine defined at the second level of the hierarchy. This machine is used to recognize an increasing-spikes trend, i.e. a trend of spikes with increasing magnitude, like the one in Figure 3B. This trend is at the second level of the hierarchy because it takes simple trends (spikes in this case) as input and outputs trends made up of those trends. When a spike is noticed, its magnitude is used as input into the increasing spikes machine. Note that the only difference between the machines of Figure 2 and Figure 3 is that the conditions are different, the states are otherwise identical. This illustrates how one basic trend machine can be applied in different circumstances by simply changing the transition conditions.

4.0 State Machine Development Environment

Typical software development goes through numerous revisions and test cycles before it can be considered robust. Programming SBFR state machines is no different. The State Machine Development Environment (SMDE) was developed as a tool to make the definition, development, and testing of the state machines tractable.

The two key components of the SMDE are the State Machine Development Language (SMDL) and the Debug and Test Facility (DTF). SMDL provides an English-like syntax for defining state machines, aliasing parameters, and defining other aspects of a load module. SMDL allows a user to associate symbolic names with state machines, local variables, states, parameters, and time components. The SMDL compiler takes load modules written in SMDL and compiles them into an executable load module, which can run with the state machine interpreter.

The DTF portion of the SMDE contains a version of the interpreter that can be used for testing the machines. Like most debugging environments, SMDE can work with a precompiled load module, but is most useful with a load module that has been compiled and loaded from the source SMDL. Debug commands can be given to report on the status of the machines as they are executing. Commands are currently implemented to print a machine's state, current transition, status register value, and local variable values. The current parameter value can also be displayed.

5.0 Related Work

Considerable work has been done by other researchers in the field of trend recognition and data monitoring, as well as all varieties of diagnostics.

One of the better developed applications is the Intelligent Cardiovascular Monitor (ICM) that implements a process trellis architecture [Sittig and Factor, 1990]. A process trellis is based on a hierarchical graph of concurrently running decision processes that move from narrowly-defined domain problems at the lower levels to abstract, broadly-defined ones at the upper levels [Factor and Gelernter, 1990].

SBFR could be viewed as an architecture like the process trellis with each machine being an individual decision process and the state machines organized as a web rather than a trellis. In its current representation, however, SBFR is better discussed as one of the techniques that can be incorporated into the trellis.With the advantage that state machines can provide data filtering, trend detection, diagnostic, and prognostic information for any level of a trellis.

Another technique for detecting and representing trends in medical data is the trend template [Haimowitz and Kohane, 1993]. A trend template is a partially ordered set of temporal intervals with uncertain endpoints. They allow a user to describe the trends as constraints that restrict certain variables over time allowing uncertainty in both the time interval of the feature and in the data values of the feature.

Trend templates and SBFR are quite similar in that they each incorporate multiple parameters and encode patterns that represent the trends to be recognized. Trend templates, however, encode a separate template for each trend, SBFR, on the other hand, builds upon a hierarchy of trend machines. Trend templates have the ability to provide an explanation of the reasoning steps that led to a particular diagnosis, SBFR does not explicitly store this information since it was developed with a computationally limited environment in mind.

6.0 SBFR Applications

The bulk of the work done in SBFR development has been on space and aviation vehicle health management (VHM), a domain characterized by limited computational resources, and hard real-time constraints. One of the most mature applications, the Sensor Interface Module System (SIMS), is a smart sensor-like device that has embedded SBFR. Plans exist to develop the SIMS into a space-qualified system perhaps hosted on a single card or even as a hybrid chip.

While VHM applications are concerned with computational efficiency, it is important to remember that SBFR provides this efficiency while retaining the expressiveness necessary to detect complex, interrelated features including those associated with diagnostics and prognostics.

Thus, SBFR is also suited for applications where the computational environment does not impose overriding design constraints, such as medicine. In general, medical applications have the luxury of fielding systems that satisfy the real-time constraints of medical monitoring and diagnosis by using powerful computers with lots of memory and processing power. For these systems, the power of SBFR lies in the expressiveness of the representation. Other medical applications, however, may be limited by computational resources, e.g. portable medical devices, and medical implants. In these cases, the efficiency of SBFR comes into play as well.

The following will discuss, some areas where SBFR can be applied to medical applications. Some are straight forward modifications of existing application areas, while others are more ambitious stretches of the technology.

6.1 Equipment Diagnosis

A natural extension of the current VHM work is to apply similar techniques to medical equipment health management. Medical equipment is becoming more complex and expensive as advancements are made. Adequately maintaining this equipment is crucial both for cost effectiveness and for patient safety. SBFR has been used in VHM in several important ways that can be analogously applied to medical applications:

- Diagnose failures in a component before they can propagate to other components, effectively limiting the damage.
- Differentiate between critical and less critical failure modes. For instance, differentiating between a sensor failure and a component failure can be very important. This knowledge could allow the equipment to continue working until it was convenient to fix or replace it.
- Monitor components for features that indicate preventative maintenance is required. For example, some components of a device may wear out over time and require regular servicing. Optimizing the preventive maintenance schedule adds safety and minimizes the system downtime and repair costs.

6.2 Real-time patient data monitoring

Another logical extension to the SBFR VHM applications is to detect clinically significant features from the patient monitor data. This is a common application for other medical health monitoring systems as well. The goal, of course, is to relieve the human observer from some monitoring duties and to provide an extra level of reliability, speed, and fidelity. The intensive care unit (ICU), in particular, is an area that benefits from patient monitoring and diagnosis where patients are being monitored continuously and timely detection is critical.

6.3 Medical Implants

A longer range application involves embedding SBFR technology in medical implants. The migration of SBFR to smart sensors has already begun and, as embedded medical technology becomes increasingly sophisticated, SBFR technology could be installed on these devices as well. This embedded capability could provide a monitoring capability for both the patient and the device itself.

6.4 Tracking Drug Effects

One of the more ambitious applications of SBFR is tracking the effects of drugs on a patient. Potential applications include:

- Monitor a patient's tolerance to a drug and track changes to that tolerance.
- Optimize dosages based on the drug's effects for a particular patient. For example, the effects of ventolin (a bronchodilator) typically last 1.5 to 2.0 hours. If the effects lasted 2.5 hours in one patient, the dosage could be adjusted accordingly. In the case of ventolin, which is delivered during hand bag ventilation, this would reduce the amount of time spent off of the mechanical ventilator [AIM-94 prep].
- Verify that a drug has the desired effect on the patient. For example, dopamine is given in different dosages for different effects, monitoring this will ensure the dosage is correct for the effect desired [AIM-94 prep].

Of course, there are many factors that influence patient's data including interactions between drugs, physiological changes, and environmental changes. Detecting the effects of the drugs from these other influences will almost certainly involve integrating several technologies. In particular, an effective system should incorporate real-time trend and feature detection with global knowledge about drug effects, dosage levels, etc.

7.0 Conclusions

In conclusion, SBFR is a good candidate for application to real time medical monitoring systems. The technology has matured to the point where application to other domains is straight forward. SBFR is a compact, expressive method for detecting and tracking trends and features in real-time data. Making it ideal as an embedded real-time monitoring system or as a data reduction technique in a larger, wide scope medical diagnosis system.

8.0 Bibliography

AIM-94 Prep Notes
Preparation notes distributed with the real time data for the 1994 AI in Medicine Spring Symposium.
March 21-23, 1994.

Factor, M and Gelemter, D
The Trellis Architecture for Intelligent Monitors
5th IEEE International Symposium on Intelligent Control, 1990
pp. 300-303

Hadden, G, Nelson, K., and Edwards, T.
State-Based Feature Recognition
17th Annual American Astronautical Society Guidance and Navigation Conference
February 2-6, 1994 Keystone, Colorado

Haimowitz, I and Kohane, I
Automated Trend Detection with Alternate Temporal Hypotheses
IJCAI-93, Chambery, France. August 1993.

Hopcroft, J. and Ullman, J.
Introduction to Automata Theory and Computation
Addison-Wesley, 1979

Nelson, K and Hadden, G.
Trend Recognition and Failure Prediction of the Attitude Determination and Control System for the Space Station Freedom
15th Annual American Astronautical Society Guidance and Navigation Conference
February 8-12, 1992 Keystone, Colorado

Sittig, D and Factor, M
Parallel Multi-Channel Biologic Signal Processing
Proceedings of the 12th Annual International Conference of the IEEE Engineering in Medicine and Biology Society
Vol. 12, 1990, pp. 1365-1366.

Washington, R. and Hayes-Roth, B.
Input Data Management in Real-Time AI Systems
International Joint Conference on Aritificial Intelligence,
IJCAI-11 1989 pp. 250-255