DATA REPROCESSING IN SIGNAL UNDERSTANDING SYSTEMS

A Dissertation Presented

by

FRANK I. KLASSNER, III

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial
fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 1996

Department of Computer Science

DATA REPROCESSING IN SIGNAL UNDERSTANDING SYSTEMS

A Dissertation Presented

by

FRANK I. KLASSNER, III

Approved as to style and content by:

Victor R. Lesser, Chair

Roderic A. Grupen, Member

Allen R. Hanson, Member

S. Hamid Nawab, Member

Lewis E. Franks, Member

David W. Stemple, Department Chair
Computer Science

Cursum perficio.

ACKNOWLEDGMENTS

Since ancient times, patience and wisdom have always been ranked among the more desirable virtues to find in another person. I'm very grateful to have had a thesis committee of virtuous men. Lewis Franks, Roderic Grupen, and Allen Hanson each helped me shape this multi-disciplinary thesis with advice about how to improve my ideas with regard to the signal processing and machine perception communities. Hamid Nawab not only trained the scientist within me in the field of signal processing but also challenged the philosopher within me during our off-work conversations. I can't even begin to describe the debt of thanks I owe to my advisor Victor Lesser for his patience with a perfectionist who didn't always know when to let go of a problem.

Victor runs an interesting research group collectively called the Distributed AI group. Although I've often joked about how in my own research I didn't "do DIS," I'm grateful for the intellectual and personal support I received from the group's students over the years: Alan Garvey, David Hildum, Susan Lander, Dorothy Mammen, Maram V. Nagendrapassad (Nagi), Dan Neiman, Tom Wagner, and Bob Whitehair. I'm particularly indebted to Norm Carver for his work on RESUN, and for his patience with me as I learned how to really use the framework. I also want to thank Malini Bhandaru, my coworker in the IPUS project, for her curiosity (it debugged a lot of IPUS) and mirthful friendship.

Having Hamid as a co-advisor gave me the benefit of working with students in his Knowledge-Based Signal Processing lab: Erkan Dorken, Ramamurthy Mani, and Joe Winograd. They did their best to train me in the arts and sciences of signal processing, and I'm deeply grateful for their help and friendship.

The computer science department has a very dependable and friendly staff of secretaries, and I want to say thanks to Darlene Fahey, Laura LaClaire, Sharon Mallory, Michele Roberts,

and Barbara Sutherland for their help in navigating me through the paperwork of being a graduate student and getting a degree.

After hanging around a large university for eight years, even a cloistered computer science student will meet many kinds of people. I've been fortunate to have usually met the right people for the right times in my doctoral journey. When I first arrived at UMass I met the "Bad Boys:" Raj Das and Concha Grande, Glenn Fala and Janine Purcell, David and Lisa Oskardmay, and Oddmar Sandvik, and Deirdre Smith. I'm thankful for their keeping me sane and sociable during those first three years of heavy courseloads.

Later on in my research trek I met Bruce Draper and Scott Anderson, who helped me to develop the rigor and confidence I needed to finish this thesis. In between academic and political debates they and their wives (Kristen Draper and Holly Anderson) also managed to get me hooked on swing dancing and fine dark beers. Thanks to them all! I also want to thank Betty Adkins, Ian Beatty, Francis Bogusz (Shadow), Jeff Clouse, Joshua Grass, Eric Hansen, Ken Magnani, Ellen Riloff, and Adolfo Socorro for their friendship and support during my graduate years.

From my high school and college days I've been blessed with a group of good friends who've done more than just "keep in touch" over the years: John and Liz Dunstone, Jeff Gabello, Michael Kane and Dorothea Mast, Carol Latzanich, Kathy Ozark, Barbara Svachak, Cindee Zawacki, and Bill Zuerblis. If they weren't calling or writing they were visiting me from exotic places such as Poland or Washington, DC. I'm particularly grateful to Michael and Dorothea for housing and entertaining me during my many road trips to Boston, as well as for their unlimited supply of optimism.

I close these acknowledgments with the people dearest of all to me, my family. My brother Steven and his wife Tonya treated me to healthy rounds of hospitality and encouragement every time I visited home for a break (contrary to popular opinion, questions about when I'd finally finish were great motivators for me!). Now I'm looking forward to having time to spend with my nephew, Steven Jr., born the day before I handed in this thesis! Finally, I thank my

parents, Frank and Lorraine Klassner, for their loving support, as the years drew on, even when they didn't fully understand what it was I was working on, or why I felt compelled to do it. I dedicate this work to them in gratitude for the life, upbringing, and faith that they gave me.

ABSTRACT

DATA REPROCESSING IN SIGNAL UNDERSTANDING SYSTEMS

SEPTEMBER 1996

FRANK I. KLASSNER, III

B.S., UNIVERSITY OF SCRANTON

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Victor R. Lesser

Signal understanding systems have the difficult task of interpreting environmental signals: decomposing them and explaining their components in terms of an arbitrary number of instances of perceptual object categories whose properties can interact with one another. This dissertation addresses the problem of designing blackboard-based perceptual systems for interpreting signals from complex environments. A "complex environment" is one that can (1) produce signal-to-noise ratios that vary unpredictably over time, and (2) can contain perceptual objects that mutually interfere with each others' signal signature, or have arbitrary time-dependent behaviors. The traditional design paradigm for perceptual systems assumes that some particular set of fixed front-end signal processing algorithms (SPAs) can provide adequate evidence for reliable interpretations regardless of the range of possible scenarios in the environment. In complex environments, with their dynamic character, however, a commitment to parameter values inappropriate to the current scenario can render a perceptual system unable to interpret entire classes of environmental events correctly.

To address these problems, this research advocates a new view of signal interpretation as the product of two interacting search processes. The first search process involves the dynamic, context-dependent selection of signal features and interpretation hypotheses, and the second

involves the dynamic, context-dependent selection of appropriate SPAs for extracting evidence to support the features. For structuring bidirectional interaction between the search processes, this dissertation presents the Integrated Processing and Understanding of Signals (IPUS) architecture as a formal and domain-independent blackboard-based approach. The architecture is instantiated by a domain's formal signal processing theory, and has four components for organizing and applying signal processing theory: discrepancy detection, discrepancy diagnosis, differential diagnosis, and signal reprocessing. IPUS uses an iterative process of "discrepancy detection, diagnosis, reprocessing" for converging to the appropriate SPAs and interpretations. Convergence is driven by the goal of eliminating or reducing various categories of interpretation uncertainty.

This dissertation discusses the IPUS architecture's features, the basic problem of auditory scene analysis (the application domain used in testing IPUS), and evaluates performance results in experiment suites that test the utility of the reprocessing loop and the ability of the architecture to apply special-purpose SPAs effectively. Although the specific research reported herein focuses on acoustic signal understanding, the general IPUS framework appears applicable to the design of perceptual systems for a wide variety of sensory modalities.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This thesis addresses the problem of designing systems for interpreting signals from complex environments. In this work, a "complex environment" is one that can produce signal-to-noise ratios that vary unpredictably over time, can contain perceptual objects that mutually interfere with each others' signal signature, and can have perceptual objects that have arbitrary time-dependent behaviors. Although the specific research reported herein focuses on acoustic perceptual systems, the general design framework discussed in this thesis is applicable to the design of perceptual systems for a wide variety of sensory modalities. The initial sections of this chapter introduce the central ideas of the thesis by presenting 1) the traditional design paradigm for signal interpretation systems, 2) the difficulties it encounters in complex environments, and 3) the alternative design approach explored in the thesis. The concluding sections of this chapter establish evaluation criteria for the thesis by summarizing 4) the research and validation issues for the new approach, 5) the thesis contributions, and 6) the organization of the thesis.

1.1 Traditional Interpretation System Design

The problem of signal interpretation, the generation of a set of symbolic hypotheses that best explains which perceptual objects and their attributes could have produced a particular numeric signal, has a long and venerable history in the field of machine perception. The need for interpretation in perceptual systems became apparent in the early 1970's with the recognition in both the machine vision and speech recognition communities that numeric signal representations alone do not provide a suitable basis for specifying systems for recognizing perceptual objects [Brady and Wielinga, 1978, Erman *et al.*, 1980]. This idea led researchers to consider augmenting perceptual systems with the ability to use *symbolic signal representations*.

Symbolic representations are formal entities with which a perceptual system could make inferences about the abstract structures within a signal, such as surfaces in images or harmonic track sets in spectrograms [Milios and Nawab, 1989]. Consider in Figure 1.1 the example of the spectrogram, which is a discrete representation of the time-dependent frequency content of a signal. Specifically, it is a matrix of values indicating, for particular time regions of the signal, the coefficients (energy) for discrete sinusoid functions into which some frequency-analysis algorithm has decomposed the signal. Examination of a three-dimensional view of the spectrogram shows that, over time, certain patterns in a signal's frequency content can be discerned. The local maxima within the coefficients from a single time point represent "peaks," or frequencies that are prominent in the signal at the given time. The regions within the spectrogram where a peak remains prominent over time form structures which are called *tracks*. The tracks can be modelled symbolically within a system as abstract structures with particular frequency variations, duration time durations, and energy variations, permitting the system to represent still "higher-level" structure such as observed relationships among the average frequency values of several tracks. A *harmonic set*, for example, is a set of tracks whose frequencies are integer multiples of some fundamental frequency f_0.

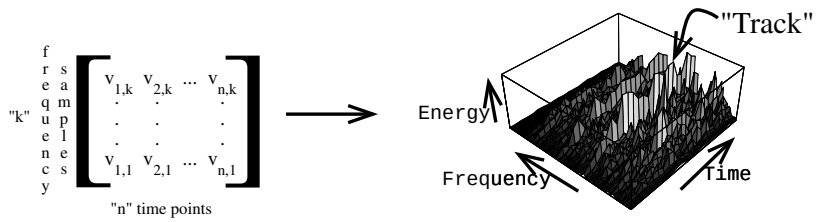


Figure 1.1. Abstract "Tracks" in a Spectrogram.

Symbolic representations can serve as a basis for perceptual object models that impose top-down constraints on how a system processes a signal. For a basic example, consider

when a track is detected in a spectrogram (generated "bottom-up" from the spectrogram values). Symbolic sound-source models that list the frequency ranges in which a sound will generate tracks could limit (provide "top-down" constraints for) the time-frequency areas where additional tracking is done in the spectrogram to those within the track-regions of sounds containing the observed track.[1] The presumed existence of one symbolic structure can also impose constraints on the nature of other structures to be searched for. An example of this occurs in the case of co-articulation phenomena in connected speech processing, where a phoneme at the end of one word can impose constraints on the acoustic characteristics to be expected at the beginning of the next word [Lee *et al.*, 1990, Lowerre and Reddy, 1980].

Of course the decision to add interpretation processes to perceptual systems involved far more than the symbolic representations themselves. The inferencing processes that manipulate the representations, the control processes that schedule the inferencing and data-collection processes, and the data structures that organize data and the symbolic representations had to be built into the new perceptual systems as well. The body of knowledge to be incorporated in the software for perceptual systems would be quite complex. It was natural, therefore, for researchers to design software architectures that would provide "scaffolding" around which system code could be organized. Several architectures were developed from the mid-1970's to the mid-1980's, of which the most widely-used today is the *blackboard architecture* [Carver and Lesser, 1993a, Nii, 1986]. This architecture provides support for 1) a blackboard, or shared central storage data structure, that contains and organizes hypotheses about various signal features, 2) knowledge sources (KSs), or pieces of code each implementing expert knowledge about the relationships among signal hypotheses and the data that support them, and 3) a control component, or code that schedules the access of KSs to data and hypotheses stored on the blackboard, as well as the order of their execution.

Since the early 1980's, most perceptual architectures have incorporated the basic design shown in Figure 1.2. This design scheme produces systems with a numeric-oriented front end

[1] As discussed in Dorken's thesis [Dorken 1994], the bottom-up generation of spectrogram tracks is a highly context-dependent problem.

that is logically separated from a symbolic-oriented interpretation component. The front end is permitted only one pass over the incoming signal, and the interpretation component is designed with the assumption that the front end's output is always an "adequate" decomposition of the signal. Interpretation processes do not usually provide structured feedback to the front end about either the adequacy of the signal processing outputs to be interpreted or any anticipated signal behavior. The development of this scheme can be attributed to several factors, foremost among them being the influence of Marr's reconstructionist school of thought in computer vision [Marr, 1982] and early psychophysical research on human perception which ignored the role of expectations in human interpretation of visual and auditory signals. Both influences led to the view that symbolic interpretation follows and depends upon signal decomposition by the front end through inversion of the geometry and physical processes that led to the original signal [Draper, 1993].

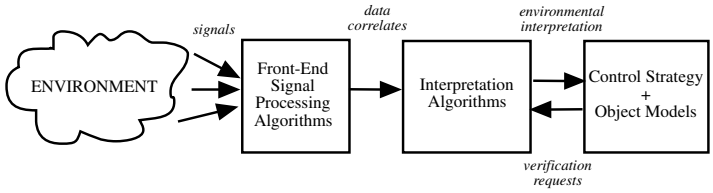


Figure 1.2. Classic Signal Interpretation Architecture.

Within the blackboard paradigm, this design scheme resulted in a split at the control level between KSs that implement numeric signal processing algorithms (SPAs) and KSs that implement symbolic interpretation algorithms (SIAs).[2] Control components generally pursue a strategy that first applies to the incoming signal a predetermined set of SPAs (the *front end*) with fixed control parameter values to obtain data correlates, or SPA outputs.[3] The control

[2]This view, to be sure, is *not* the only one within the machine perception community. There is an alternative school of thought [Draper, 1993, Strat 1991, Kohl *et al.*, 1987, Nagao and Matsuyama, 1980] that advocates feedback (in various degrees) between the front end and interpretation components of a signal interpretation system. As will be discussed in Chapter 3 the research in this thesis is complementary with this school.

[3]Note that the term "correlate" is taken from the voice-recognition literature. Within the signal processing

parameters of the SPAs are fixed to some setting that would provide correlates of adequate quality for generating hypotheses about symbolic structures. These correlates are then interpreted as reasonably certain support for symbolic signal structures at various abstraction levels. The control component uses these "islands of certainty" [Lesser *et al.*, 1977] within the signal to index into an object-model database. The retrieved models then inform the control component in the application of additional interpretation algorithms to verify other signal structures that are required by the models.

1.2 Complex Environments and Traditional Design

The traditional design paradigm assumes that a fixed front end can provide adequate (not necessarily optimal) evidence for reliable interpretations regardless of the range of possible scenarios in the environment. This assumption is plausible for systems that monitor stable environments, but not for those that monitor complex environments. In complex environments with interacting objects and variable signal-to-noise ratios, the choice of front-end SPAs and their parameter settings greatly impacts the generation of adequate correlates for interpretation processes. Indeed, parameter values inappropriate to the current scenario can render a perceptual system unable to interpret entire classes of environmental events correctly.

An SPA's parameter values induce capabilities or limitations with respect to the scenario being monitored. Consider the use of the generic Short-Time Fourier Transform (STFT) algorithm [Nawab and Quatieri, 1988] to produce spectrograms for acoustic signals. Conceptually, the algorithm implementing the STFT computes a series of Discrete Fourier Transforms (DFTs) [Oppenheim and Schafer, 1989] on successive blocks of data points in a discrete time signal. Referring to Figure 1.1, consecutive columns in the spectrogram matrix from an STFT are the DFTs of consecutive blocks of signal data. An STFT instance has particular values for its parameters: analysis window length (number of signal data points analyzed at a time), frequency-sampling rate (number of points computed per column in Figure 1.1's matrix), and decimation interval (number of signal data points between consecutive analysis

research community SPA outputs are referred to variously as "measurements," "functionals," or "statistics."

window positions).[4] Depending on assumptions about a scenario's spectral features and their time-variant nature, these parameter values increase or decrease the usefulness of the spectrogram produced by the instance.

It can be shown through analysis of Fourier theory (see [Oppenheim and Schafer, 1989], Section 11.3, for example) that a fixed STFT with a long analysis window length will provide fine frequency resolution for scenarios containing sounds with time-invariant components, but at the cost of poor time resolution for sounds with time-varying components. Tracks of sounds that change in frequency over time, or "chirp," too quickly for the STFT will produce correlate peaks that look too widely separated to be linked as a track with a steep slope, while the onsets and decays of impulsive sounds will be "smeared" over the spectrogram, making it difficult to detect the presence of such sounds. Conversely, a fixed STFT instance with short window lengths will provide fine time resolution for scenarios containing sounds with time-varying components such as chirps or reverberatory decays, but at the cost of poor frequency resolution for sounds with close frequency components. The tracks of sounds which are too close to each other in frequency to be resolved (separated) by the STFT will produce correlate peaks that indicate only a single merged track in the output spectrogram. It should be noted at this point that analysis of the Uncertainty Principle [Gabor 1946] implies that one cannot obtain an STFT SPA instance (or, for that matter, design a new SPA) that simultaneously provides infinite frequency resolution and infinite time resolution.

Figure 1.3 illustrates the difficulty in using fixed front ends to interpret a complex acoustic environment. Figure 1.3a shows the stylized frequency tracks of four sounds as they would appear in an ideal spectrogram if they were processed with STFT SPAs appropriate for each portion of the scenario. Darker shading indicates higher energy. Figure 1.3b shows how the tracks appear when the entire scenario is processed by one STFT SPA appropriate only

[4] Note that this is a nonstandard usage of the term "decimation." The term "decimation" is most often used in the signal processing literature to refer to the process of first *downsampling* (reducing the number of sample values considered) a discrete signal and then lowpass filtering the result. (see [Oppenheim and Schafer, 1989], Section 3.6.1) There is no standard terminology for the STFT parameter being described here; "decimation interval" was selected because the parameter effectively controls how often windows of signal data are analyzed (i.e. how often the signal is "window-sampled") to produce DFTs.

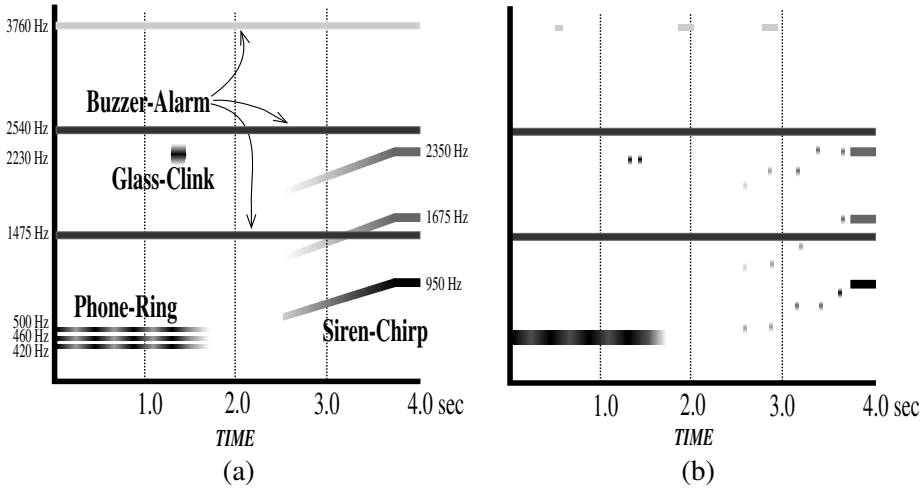


Figure 1.3. Example Problems From Fixed-Front-End Processing.

for the steady-state portion of the last sound (`Siren-Chirp`'s last 0.5 seconds) in the scenario. The STFT parameter settings used throughout Figure 1.3b were FFT-SIZE: 512, WINDOW: 256, and DECIMATION: 256, while the peak-picker's parameter setting was PEAK-THRESHOLD: 0.09. The signal was sampled at 8KHz. DECIMATION is the separation between consecutive analysis window positions; the value was set to 512 to permit the fastest possible processing of the data. PEAK-THRESHOLD is the energy required for a Discrete Fourier Transform point to be considered as a peak, and its value here was selected to keep low-energy noise in the spectrogram from generating false-alarm tracks.

Due to inappropriate processing, the analysis of the first two seconds of signal data introduces some distortions that would lead to ambiguous interpretations and completely undetected sources. A distortion is a process caused by poor SPA parameter settings that produces correlates that inaccurately represent the state of the environment to interpretation processes. In the first block of data (time 0.0 to 1.0 seconds) in Figure 1.3b, `Phone-Ring`'s tracks are merged because the frequency resolution afforded by the STFT is not adequate for features so close in frequency. `Glass-Clink`'s frequency track is not even detected in 1.3b's next data block (time 1.0 to 2.0 seconds) because the STFT's analysis window does not provide adequate time resolution to isolate the sound's spectral features. The high peak-energy

threshold causes the peak-picker to miss low-energy peaks in the STFT spectrogram that could have served as evidence for `Buzzer-Alarm`'s high-frequency, low-energy track.

Within the traditional design paradigm the approach to handling these types of problems is to add to the system's front end more SPAs with settings appropriate to the problematic environmental scenarios. This requires an exhaustive analysis of the environment and intricate tailoring of front ends to each possible combination of perceptual objects in the environment. This approach is feasible only for significantly constrained environments. To avoid ambiguous signal-symbol mappings in complex environments, interpretation systems often require combinatorially explosive SPA sets with multiple parameter settings [Dorken *et al.*, 1992], with consequent processing time costs.

At this point one might question these criticisms of interpretation systems with fixed, one-pass front ends by claiming that the human auditory system, with its cochlear signal processing, is an example of a fixed, one-pass system that handles a wide variety of complex acoustic scenarios quite well. However, the claim misses (1) the fact that the human auditory system's front end is only well-adapted for interpreting sounds from a moderately restricted environment that was evolutionarily important to the species, and (2) the possibility that two-pass revision, or reprocessing, occurs in the intermediate stages of the auditory system's interpretation process.

The cochlea's time-frequency processing has been shown to be similar to that performed by wavelet analysis [Rioul and Vetterli, 1991] parameterized to produce spectrograms having fine frequency resolution with poor time resolution in the lower frequency regions, and poor frequency resolution with fine time resolution in the higher frequency regions. Thus, the human ear is able to discriminate low-frequency sounds (which, from an evolutionary standpoint, is useful for differentiating the growls and calls of predators or mates) and determine precise times for high-frequency events (which, from an evolutionary standpoint, is useful, for example, for localizing the position of stalking predators which may have snapped twigs or scrapped rocks). However, the fixed front end of this system creates difficulties for humans when they must handle scenarios from a more unrestricted environment. For example, the human ear's poor

frequency resolution in the high-frequency spectral regions makes it ill-suited for discriminating among various aircraft engine whines that could indicate metal fatigue or poor balancing. Such unrestricted environments ultimately require additional "SPAs" (electronic hardware tuned to the particular engines) beyond those of the human ear for adequate interpretation, leading to the same type of combinatorial SPA explosion previously described.

In regard to the second objection, Ellis [Ellis, 1996] summarizes work [McAdams, 1984, Warren, 1984] that indicates there is a possibility that the human auditory system reprocesses its initial interpretations. The McAdams work reported that during the initial presentation of an oboe note to human observers, a single note is perceived. However, as the note progresses, the even harmonics of the note undergo progressively deeper frequency modulation. This led observers to change their initial interpretation of the spectral energy as a single note to one in which there were two distinct sounds, and to apply that changed interpretation to the entire note. The Warren work examined how changes in the order of presentation of alternating wider (0-2KHz) and narrower (0-1KHz) noise-bands influenced the grouping humans performed on the signal components. When the narrower band came first, observers interpreted the signal as containing a continuous 0-1KHz sound with a periodic 1KHz-2KHz sound. That is, the lower 0-1KHz energy of the wider bands was merged over time with that of the narrower bands, as one might expect of a one-pass interpretation system that is primed to group new data with data that has just been observered. However, when the wider band started the alternation, observers reported that the first 0-2KHz band was interpreted as a single sound. After the initial wider band was completed, and as the alternation continued, however, the interpretation of the first band was *revised* to that obtained for the first alternation. That is, the observer reported that the first 0-2KHz band was perceptually "broken" into a 1KHz-2KHz burst simultaneous with a continuous narrower 0-1KHz sound, *in light of the rest of the signal*. Though by no means conclusive, both experiments lend support to the possibility that the human auditory system can revise its earlier interpretations.

1.3 Thesis Paradigm

1.3.1 Motivation

To circumvent the combinatorial explosion of fixed SPAs, a small SPA set would be sufficient if comparisons could be made between the SPAs' computed correlates and dynamically-generated signal structure expectations. Failure in the verification of expectations about SPA outputs can indicate that either 1) the signal structure expectations are based on incorrect interpretations or that 2) the SPA's computed correlates have been distorted because the SPA's parameter values are inappropriate to the current scenario. In the first case a perceptual system could follow the behavior of traditional interpretation systems and reinterpret the current scenario based on the SPA's given correlates. In the second case a system could reconfigure the SPA's parameters or selectively replace it with a more appropriate SPA, and reprocess the signal in a focused manner to obtain expected correlates.

Adopting a search-oriented model of these two possible system responses to unmet expectations, one can see that the first behavior corresponds to a decision to evaluate how much better an alternative state in the interpretation state space might explain the front end's output. The second behavior then translates into a decision to evaluate how much better an alternative state in the front end state space might be at generating evidence for unambiguously supporting an interpretation. Figure 1.4 shows how an interleaving of these behaviors results in progress in each search space. Figure 1.4A shows an interpretation system's progress within its interpretation space. The label outside each state indicates the front end(s) being used to provide support evidence for the state's interpretation (set of instances of perceptual objects Hn). Figure 1.4B shows the front ends explored by the system. The label outside each state indicates the interpretation expected for the front end, and whether the front end's correlates supported the interpretation.

The system behavior in Figure 1.4 can be summarized as follows. Initially, the interpretation system uses front end $\mathcal{A}$ to collect evidence, and hypothesizes that one perceptual object of type H1 is present. Attempting to account for more signal energy, the system then explores the

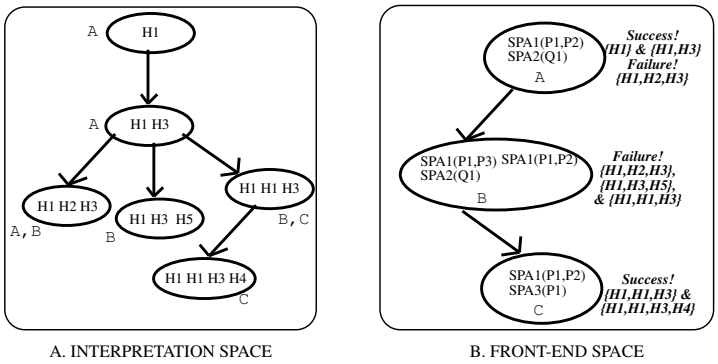


Figure 1.4. Example of Dual Interpretation and Front-End Search.

interpretation state {**H1, H3**} and finds that $\mathcal{A}$'s SPAs have also produced evidence to support the interpretation. When attempting to explain the remaining signal energy, the system finds that an additional single object of either type H1, H2, or H5 could be hypothesized. Choosing H2 first (i.e. the interpretation state {**H1,H2,H3**}), the system finds that the SPAs in front end $\mathcal{A}$ do not provide sufficient or unambiguous evidence for the H2 instance. Deciding that $\mathcal{A}$ is not suited for the interpretation, the system applies front end $\mathcal{B}$'s SPAs to the signal data, and this time finds negative evidence for H2's instance, causing it to abandon interpretation state {**H1,H2,H3**} and proceed to explore {**H1,H3,H5**}. The SPAs in front end $\mathcal{B}$ provided indisputable negative evidence for any instance of H5, leading the system to explore interpretation state {**H1,H1,H3**}. This time although $\mathcal{B}$ did not provide positive evidence for the second instance of type H1, the system finds that $\mathcal{B}$ was inappropriate to the interpretation. The system then uses signal processing constraints to determine that front end $\mathcal{C}$ should be appropriate for supporting or disproving the existence of the extra H1. According to Figure 1.4B the correlates produced by the new front end do in fact support the second H1 hypothesis, and ultimately support the creation of a final H4 hypothesis. The final interpretation state is {**H1,H1,H3,H4**}, and because it accounts for enough signal energy, interpretation search stops.

A domain's explicitly-represented signal processing theory can play three generic roles in controlling the application of SPAs within constraints that dynamically arise from the emerging

list of observed symbolic structures:

discrepancy detection: provide methods to determine discrepancies between an SPA's expected correlate set and its computed correlate set.

diagnosis: define distortion processes that explain how discrepancies between expectations and an SPA's computed correlates result when the SPA has inappropriate values for specific parameters.

reprocessing: specify new strategies to reprocess signals so that distortions are removed or ambiguous data is disambiguated.

These observations about the power of formal signal processing theory in analyzing complex environments are the reasons behind the claim in this thesis that the explicit representation of the knowledge in signal processing theory is crucial to systems that monitor complex environments. The processing of signals from complex environments will benefit from a new view of signal interpretation as the product of two interacting search processes. The first search process involves the dynamic, context-dependent selection of signal features and interpretation hypotheses, and the second involves the dynamic, context-dependent selection of appropriate SPAs for extracting correlates to support the features. Signal interpretation architectures should support the use of theoretical relationships between SPA parameters and SPA outputs to structure these dual searches for SPAs appropriate to a scenario and for interpretations appropriate to the SPA correlates.

1.3.2 Architecture Overview

This thesis proposes the *Integrated Processing and Understanding of Signals* (IPUS) architecture as a formal and domain-independent blackboard-based framework for structuring bidirectional SIA/SPA interaction in complex environments. This interaction combines the search for front end SPA configurations appropriate to the environment with the search for plausible interpretations of front end processing results. The architecture is instantiated by a

domain's formal signal processing theory. It has four primary components as conceptual "hooks" for organizing and applying signal processing theory: discrepancy detection, discrepancy diagnosis, differential diagnosis, and signal reprocessing. These components have the following functionality:

- detect discrepancies between data expectations and actual data observations,
- diagnose these discrepancies and ascribe reasons for observational uncertainty,
- determine reprocessing strategies for uncertain data and expected scenario changes, based on the results of the diagnosis, and
- determine differential diagnosis strategies to disambiguate data with several alternative interpretations.

To exploit the constraints that signal processing theory can impose on the dual searches within the signal interpretation problem, IPUS is designed with a "discrepancy detection, diagnosis, reprocessing loop." The architecture uses an iterative process for converging to the appropriate SPAs and interpretations. For each block of data, the loop starts by processing the signal with an initial configuration of SPAs (KSs). These SPAs are selected not only to identify and track the signals most likely to occur in the environment, but also to provide indications of when less likely or unknown signals have occurred. In the next part of the loop, a *discrepancy detection* process tests for discrepancies between the correlates of each SPA in the current configuration and (1) the correlates of other SPAs in the configuration, (2) application-domain constraints, and (3) the correlates' anticipated form based on high-level expectations. Opportunism in the architectural control mechanism permits this process to execute both after SPA output is generated and after interpretation problem solving hypotheses are generated. If discrepancies are detected, a *diagnosis* process attempts to explain them by mapping them to a sequence of qualitative distortion hypotheses. The loop ends with a *signal reprocessing* stage that proposes and executes a search plan to find a new front end (i.e., a set

of instantiated SPAs) to eliminate or reduce the hypothesized distortions. After the loop's completion, if there are any similarly-rated competing top-level interpretations, a *differential diagnosis* process selects and executes a reprocessing plan to find correlates for features that will discriminate among the alternatives.

Although the architecture requires the initial processing of data one block at a time, the loop's diagnosis, reprocessing, and differential diagnosis components are not restricted to examining only the current block's processing results. If the reprocessing results from the current block imply the possibility that earlier blocks were misinterpreted or inappropriately reprocessed, those components can be applied to the earlier blocks as well as the current blocks. Additionally, reprocessing strategies and discrepancy detection application-constraints tests can include the postponement of reprocessing or discrepancy declarations until specified conditions are met in the next data block(s).

The dual searches discussed earlier become apparent in IPUS with the following two observations. First, each time signal data is reprocessed, whether for disambiguation or distortion elimination, a new state in the SPA instance search space is examined and tested for how well it eliminates or reduces distortions. Second, failure to remove a hypothesized distortion after a bounded search in the SPA instance space leads to a new search in the interpretation space. This happens because the diagnosis and reprocessing results represent an attempt to justify the assumption that the current interpretation is correct. When either diagnosis or reprocessing fails, there is a stronger likelihood that the current interpretation is not correct and a new search is required in the interpretation space.

1.4 Analysis of IPUS

The IPUS architecture implements perception as the integration of search in a front-end-SPA space with search in an interpretation space. This integration raises several issues, and encourages use of an alternative methodology for the design of perceptual systems' front ends. This section briefly describes the major issues and introduces the real-world problem domain in which the thesis validates IPUS.

1.4.1 Architectural Implications

The ability to interleave searches within the interpretation and front-end spaces raises the question of how one search process is determined to be the guide for the other, and how their roles can switch. In general, the search process whose current state produces the lower uncertainty serves as the standard against which progress toward a complete interpretation or adequate front end is measured in the other. Within the interpretation search process "uncertainty" refers to the portion of the signal[5] explained by the current interpretation state and the strength of the negative (i.e. missing or incomplete) evidence against each hypothesis in the interpretation. Within the front-end search process "uncertainty" refers to the degree of inconsistency found among the data correlates from SPAs whose outputs are supposed to be related according to their domain signal processing theory. This reliance on uncertainty for driving and halting the dual searches in IPUS requires some mechanism for representing uncertainty; Chapter 2 details how the RESUN [Carver and Lesser, 1993b, Carver and Lesser, 1991] planning framework provides this mechanism.

Another important question to consider about the interleaved searches is whether the interleaving process will converge. Although it is beyond the scope of this thesis to provide a formal convergence proof, the following line of reasoning serves as an informal indication that for a given finite subset of an input signal, convergence to a final interpretation hypothesis-set and a final front-end SPA-set will occur. With a given interpretation as a standard, IPUS might iterate on several diagnoses and reprocessings of a portion of the signal in attempts to verify particular missing data correlates required by the interpretation. With each iteration, correlates from various SPA-sequences with different control parameter settings will be generated. In the IPUS paradigm these correlates are said to have originated from different *processing contexts*. Assuming that not just each processing context but also their data correlates are recorded during reprocessing, the successive diagnose-then-reprocess iterations will generate tighter and

[5] In the IPUS framework evaluated in the thesis this will be the percent of the total input signal energy accounted for by the current interpretation.

tighter constraints on the types of signal features that could remain unobserved given all previous reprocessings. Eventually a point will be reached when the domain's formal signal processing theory and the narrowing constraints on the possible values of an interpretation's feature (e.g. track energy) preclude the existence of the desired data correlates. If search is restarted in the interpretation space, the results from previous reprocessings will constrain the new interpretation search by eliminating from consideration objects with features requiring correlates that should have been found during the reprocessing.

As an IPUS-based system performs reprocessing, it will generate correlates from various processing contexts. Given the previous discussion's conclusion that these correlates should be saved rather than discarded, the question arises as to how this data should be managed and exploited to offset its storage costs with time savings. As will be seen in Chapter 2, the explicit representation of processing contexts and the domain's signal processing theory permits IPUS-based systems to examine their reprocessing history for processing contexts that would provide data correlates that were at least as detailed as those required by a current reprocessing request. If data correlates from such previous processing contexts exist, IPUS-based systems can save reprocessing effort by reusing them as evidence for missing evidence. This process is referred to as *context mapping*, and will be seen to be a useful mechanism for a type of sensor fusion where results from various reprocessings can be combined as evidence for time-dependent signal features.

The final issue considered in this section involves the implications of the IPUS architecture for front-end design in interpretation systems. Since the traditional design paradigm has emphasized one pass over input signal data, there has been a tendency to build systems with fixed front ends that are expensive because they must provide detail for the most ambiguous cases even when the detail is unnecessary. Because IPUS has the ability to selectively reprocess uncertain portions of a signal with specialized SPAs, the framework provides many opportunities for using *approximate processing* techniques to reduce the complexity of front ends while sacrificing precision in SPA output where permissible. Approximate processing [Decker *et al.*, 1990]

refers to the deliberate limitation of search processes in order to trade off certainty for reduced execution time. Approximate SPAs are algorithms whose processing time can be limited in order to trade off precision in their output correlates for reduced execution time. The availability of approximate SPAs permits the formulation of IPUS control strategies that first use approximate SPAs to generate a rough picture of the environment that is refined only where the front-end correlates' interpretations are too uncertain. Refinement is achieved by reprocessing these limited signal portions with SPAs that produce correlates having greater precision. These non-approximate SPAs would ordinarily be quite expensive if applied to the entire signal, but when they are applied only in restricted signal regions their costs become manageable.

1.4.2 Architecture Validation Domain

This thesis validates the IPUS architecture on the auditory scene analysis problem [Bregman 1990], which involves the segregation and identification of simultaneous and sequential sounds in an acoustic signal. Auditory scene analysis is an interesting problem that arises in applications such as assistive devices for the hearing impaired and robotic audition. The field is replete with issues concerning the relationship between the determination of an SPA's appropriateness and multi-sound interactions in complex environments. In particular, the thesis work focuses on the problem of adaptively generating spectrograms that provide time- and frequency-resolution (i.e. detail in time or frequency) adequate to the task of separating signal signatures of simultaneous sounds with a variety of time-dependent behaviors. A secondary reason for performing validation in the acoustic domain is that signals from individual real-world sounds are relatively simpler to collect and easier to combine for experimental work than signals from other perceptual modalities such as vision or taction.

All thesis evaluation experiments are performed on an IPUS-based Sound Understanding Testbed (SUT) implemented within the blackboard framework [Lesser *et al.*, 1995, Lesser *et al.*, 1993]. The system has 10 KSs implementing SPAs that can be used in front-end processing and 7 KSs implementing SIAs (signal interpretation algorithms) that generate high-level interpretations, as well as other KSs implementing the key components within the

IPUS architecture specification: discrepancy detection, discrepancy diagnosis, and differential diagnosis.

Table 1.1. IPUS Sound Library Categories

CATEGORY	PROPERTIES	EXAMPLES
chirp	time-dependent frequency shifts	owl hoot, door creak
harmonic	sound has frequencies $f_1, \ldots, f_n$ that are integer multiples of some fundamental f_0.	fire alarm, car horn
impulsive	short acoustic bursts cause wideband energy over entire frequency spectrum.	door knock, pistol shot
repetitive	need not have a precise period.	footsteps, phone ring
transient	signal onset or signal turn-off behaviors differ from those in steady-state.	bell toll, hairdryer start

The testbed has a library of 40 real-world sound models from which to generate signal interpretations. The library sounds were selected to provide a reasonably complex subset of the acoustic behaviors and sound interactions that can arise in random real-world auditory scenarios. Table 1.1 summarizes the acoustic behaviors that can be found in the library. As an indication of the potential for interactions among sounds randomly selected from the library and placed in scenarios with random start times, it should be noted that the expected frequency range of each narrowband track (e.g. $\leq$ 100 Hz wide) of each library sound overlaps a track of at least one other sound. Note that the greater the number of overlapping tracks there are in a spectral region, the greater the amount of interpretation search that must be done to disambiguate competing sound hypotheses that may arise.

The IPUS performance experiments in this thesis are designed first to demonstrate in general the framework's dexterity at effectively applying both special-purpose (e.g. approximate) SPAs and general-purpose (e.g. non-approximate) SPAs and adapting interpretations in complex scenarios and second to provide indications of the importance of the framework's major "reprocessing loop" components to signal interpretation.

1.5 Contribution Summary

To summarize, this thesis makes the following research contributions:

- a generic architecture for designing perceptual systems for complex environments that represents a significant departure from conventional systems,
- a framework for fusing correlates obtained from disparate front ends' analysis,
- through validation of the IPUS architecture in the real-world problem of auditory scene analysis:
  - a demonstration of the role of reprocessing and SPA theory in improving the quality of interpretations
  - a demonstration of the applicability and potential advantages of approximate processing within the IPUS architecture,
- a platform for future exploration of how to computationally approximate theories of auditory perception.

1.6 Thesis Organization

The subsequent chapters in this thesis are organized as follows. Chapter 2 presents a detailed description of the IPUS architectural paradigm. Chapter 3 discusses work in signal interpretation architectures that is related to IPUS. Chapter 4 presents a description of how IPUS was instantiated in a sound understanding testbed (SUT). Chapter 5 reports on the SUT's interpretation performance on acoustic scenarios composed of real-world sounds. Chapter 6 concludes with an evaluation of the SUT's performance results and the IPUS framework's organization, with respect to the research contributions outlined earlier. Appendix A supplies the interested reader with details on the sounds in the SUT library that was used in the experiment suites, while Appendix B contains a trace of the SUT's behavior for one of the acoustic scenarios in Chapter 5's experiments.

CHAPTER 2

INTEGRATED PROCESSING AND UNDERSTANDING OF SIGNALS

This chapter presents the abstract, domain-independent specification for the IPUS architecture. It has five sections. The first section recounts an extended example of the basic behavior that this thesis aims for in a blackboard interpretation system for complex environments. With this description as a backdrop, the second section justifies and describes the RESUN planning framework that was selected to implement the architecture and provide appropriate control over blackboard KSs' execution. The third section describes the RESUN implementation of the basic IPUS control strategy as well as the data structures and code that provide IPUS with the concepts of processing-context and adaptive front ends. The fourth section discusses the generic specifications of each component of the architecture's reprocessing loop: discrepancy detection, discrepancy diagnosis, reprocessing, and differential diagnosis, including their abstract realization in the planning framework. The fifth section closes the chapter with a summary of what the architecture requires for instantiation.

2.1 IPUS-Based System Behavior: An Example

In Section 1.2 the acoustic scenario in Figure 1.3 was used to illustrate the pitfalls of fixed front-end processing in complex environments. This section returns to that scenario for a concrete example of how an IPUS-based interpretation system with an initially inappropriate front end would ideally behave in processing a complex environment's signal.

Figure 2.1a shows the time-domain waveform for the signal, while Figures 2.1b and 2.1c duplicate the information from Figure 1.3 for the reader's convenience. Figure 2.1b shows how the correlates for the sounds in the scenario would appear in the time-frequency domain using context-appropriate processing. `Phone-Ring` and `Siren-Chirp` are 1.2 times as energetic

as `Buzzer-Alarm`, and `Glass-Clink` is an impulsive source 3.0 times as energetic as `Buzzer-Alarm`. Darker shading indicates higher frequency-domain energy. Figure 2.1c shows how the sounds' correlates are distorted when a front end reasonably designed for the end of the fourth data block is applied indiscriminately throughout the scenario.

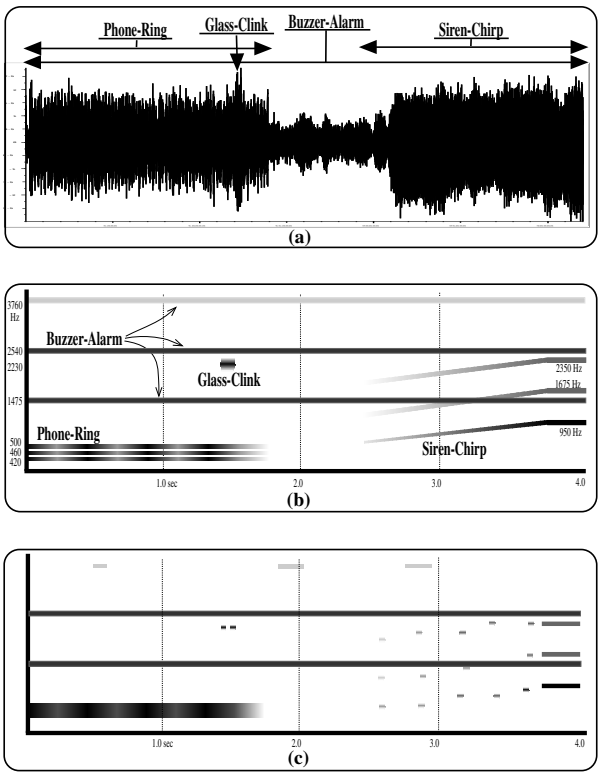


Figure 2.1. IPUS Processing Example.

Assume that an IPUS interpretation system has an STFT SPA, a local-peak-picker SPA, a time-domain energy-tracker SPA, and is configured to interpret the 8KHz-sampled waveform data in 1.0-second blocks. Assume also that the system's sound-model database was loaded with models for the five narrowband sources shown in Figure 2.2. The vertical axis represents frequency and the horizontal axis represents time. The range below the time axis indicates the minimum and maximum expected durations for the sound's spectral tracks. The energy changes for each track are represented qualitatively by shading gradations, with darker shades

indicating higher energy. In the figure the sounds' frequency components are labelled by single-frequency values only for clarity; the formal sound definitions would have frequency *ranges* specified for each component.

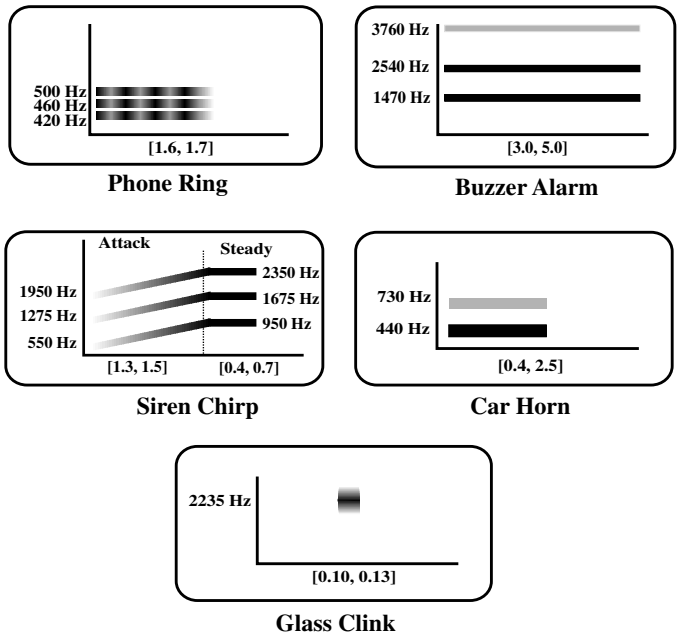


Figure 2.2. IPUS Processing Example's Sound Database.

Next, assume that the front end SPAs were initialized with the following parameter values under the assumption that `Siren-Chirp`'s steady behavior (seen at the end of block 4) would predominate in the scenario:

FFT-SIZE: 512

The number of uniformly-spaced frequency samples computed for each Short-Time Fourier Transform (STFT) analysis window position.

WINDOW: 256

The number of data points to which each FFT in the STFT algorithm is applied (≤ FFT-SIZE).

DECIMATION: 256

The number of points between consecutive STFT analysis window positions.

PEAK-THRESHOLD: 0.09

Spectrum points with energy below this value are rejected by the peak-picking algorithm.

That is, assuming that `Siren-Chirp`'s steady frequency tracks are all separated by approximately 700 Hz, and that this kind of signal would predominate in the environment, the 31.25 Hz frequency resolution permitted by the length of the 256-point analysis window would enable the STFT to resolve the siren's tracks with little difficulty. The decimation value of 256 is selected to ensure no overlap between consecutive analysis windows, which in turn provides the quickest complete processing of the signal data points, at a cost of interacting with the analysis window length to produce a spectrogram that can resolve frequency events no closer than 0.032 seconds in time. Lastly, the peak-threshold was selected to prevent the system from becoming distracted with noise-generated peaks whose energies are lower than that generally observed for `Siren-Chirp`'s steady tracks.

Finally, assume that the system pursues an "island-driving" control strategy for each block where it executes the front end and produces peak hypotheses, then retrieves the models of sounds whose tracks roughly overlap the frequency regions covered by the peaks, then confirms the models' tracks by verifying that the peak bounds unambiguously lie within the tracks' expected regions.

There are several critical behaviors that the IPUS system should perform if it is to reasonably analyze Figure 2.1a's signal. In block 1, after initial front-end processing and model-retrieval, the system finds three alternative interpretations of the data in the $[420, 500]$ frequency region. That is, there is the possibility that it could be caused by `Phone-Ring` or `Car-Horn`, or perhaps both occurring simultaneously. One reason for this confusion stems from the fact that the energy threshold setting for the peak-picking algorithm is high and would prevent `Car-Horn`'s low-energy microstream from being detected if in fact it were present. The second reason is that the frequency-sampling provided by the STFT algorithm's fft-size parameter does not provide

enough frequency sample points to resolve the $[420, 500]$ region into `Phone-Ring`'s three microstreams.[1] The uncertainty in this situation should be resolved through reprocessing under the direction of differential diagnostic reasoning, which increases resolution and decreases the energy threshold.

While verifying that the data in block 1's supports all of `Buzzer-Alarm`'s tracks, the system also should detect that the sound's track at 3760 Hz is missing. After engaging in signal processing diagnosis of the discrepancy, the system should find that one reason for this is that the track's energy might be too low for the peak-picker's peak-threshold parameter setting. The discrepancy would be resolved when the system reprocessed the existing spectrogram with a peak-picker having a lower peak-threshold value.

In block 2, as it executes its front end, the system should detect a discrepancy between the outputs of its time-domain energy estimator SPA and its STFT SPA. Due to *Glass-Clink*'s presence, the energy tracking SPA's correlates will indicate a substantial energy increase followed about 0.1 seconds later by a precipitous decrease. The STFT SPA, however, will produce a spectrogram with only one or two high-energy peaks, which, in a noisy environment, is not significant enough for believing that new short frequency track accounting for the time-domain energy flux is present at 2235 Hz. The system should determine through diagnosis that this could be because the STFT's decimation value is too long. The testbed also detects a discrepancy between expectations established from block 1 for the $[420, 500]$ frequency region and the STFT SPA's output. The STFT SPA produces short contours that cannot support the expected microstreams for `Phone-Ring` because of inadequate frequency sampling in the region. Both discrepancies are resolved by reprocessing. The first discrepancy is resolved through reprocessing with a smaller decimation value and smaller STFT intervals, while the second is resolved through reprocessing with the finer frequency sampling provided by a 1024 fft-size.

[1] Although the center frequencies of each track are further apart than the $8000/512 = 16.25$ Hz frequency sampling afforded by the STFT, the tracks' expected frequency *ranges* are too close.

In block 3, `Siren-Chirp`'s attack interacts with the poor time-resolution of the STFT SPA to produce a set of widely-separated peaks that the testbed cannot immediately interpret as the sound's attack region. In block 4, however, the testbed should use the discovery of `Siren-Chirp`'s steady region as the basis for looking back to block 3's region for the attack. After finding no conclusive peak support for the sound's attack region, the system should diagnose the discrepancy as being attributable to poor time resolution in the spectrogram. It should then reprocess the waveform in the time-region of the expected chirp with an STFT suitable for detecting the behavior (e.g. one with a short WINDOW around 64 points long).

2.2 Control in IPUS

From the preceding description of desired IPUS behavior, one should be struck by the important role the concept of "discrepancy" plays in how the architecture controls the application of SPAs and other KSs. It would appear useful therefore to design IPUS with the ability to represent discrepancies as symbolic, explicit factors that can influence the confidence levels a system maintains for interpretation hypotheses or numeric SPA outputs. Such *sources of uncertainty* (SOUs) could provide some of the cues for a control mechanism to use in making context-sensitive decisions to engage or interrupt the architecture's reprocessing loop or dynamically modify a system's default front end.

Indeed, the overall context-sensitive nature of the hypothetical system's behavior is a key feature. Though the hypothesized system required that the signal be processed one block at a time, the diagnosis, reprocessing, and differential diagnosis components were not restricted to examining only the current block's processing results. When the current block's processing results implied the possibility that earlier blocks were misinterpreted or inappropriately reprocessed, those components were applied to the earlier blocks as well as the current blocks. Although it did not happen in the example scenario, the postponement of reprocessing until specified conditions are met would also be a useful context-sensitive behavior for cases when uncertain signal data at the very end of a block requires the next block's data for meaningful reprocessing.

For these reasons, IPUS uses Carver's RESUN [Carver and Lesser, 1993b] planner framework to control blackboard KS execution. This framework views interpretation as a process of gathering evidence to resolve hypotheses' SOUs. It incorporates a rich language for representing SOUs as structures which trigger the selection of appropriate interpretation strategies. For some idea of the this representation's scope, consider the following partial list of SOUs in the language. There is an SOU called `Partial-Support`, which, when found on a blackboard hypothesis, represents the situation that the hypothesis has uncertainty because support evidence has not yet been sought for it (e.g. a spectrogram track's termination has not been searched for yet). Another SOU called `Possible-Alternative-Explanation` represents the situation that a hypothesis is uncertain because there exist other explanations for it that have not yet been considered. A third SOU called `Support-Exclusion` represents the uncertainty that a hypothesis has because some subset of the support evidence desired for it has not been found because it is highly likely that the evidence in fact does not exist.

In addition to its SOU language, the RESUN planning framework provides an elaborate language for specifying and executing the plans available to a system for satisfying the goals it generates as it solves an interpretation problem. The following brief description of the control-plan framework concentrates on the RESUN features that are relevant to IPUS; interested readers can find more detailed treatments in the planning community's literature [Carver and Lesser, 1993b, Carver and Lesser, 1991]. Problem-solving under RESUN is driven by information in a *problem solving model* (PSM), which is a data structure that maintains a list of the current highlevel blackboard interpretation hypotheses and the SOUs associated with each one's supporting hypotheses. SOUs from the PSM are selected by the planner to be resolved. That is, with selection of an SOU, the RESUN planner establishes a goal that the SOU be eliminated or reduced, if possible. Goals are expressed as predicate statements such as (**Have-SOU-Solved** ***SOU***) where *SOU* would be an SOU to be solved. The planner controls the selection and execution of blackboard KSs for achieving this goal with its library of *control plans* and *focusing heuristics*. Both Figures 2.3 and 2.4 should be consulted closely

in the following discussion of control-plan definitions, while Figure 2.3 should aid in the later discussion of focusing heuristics.

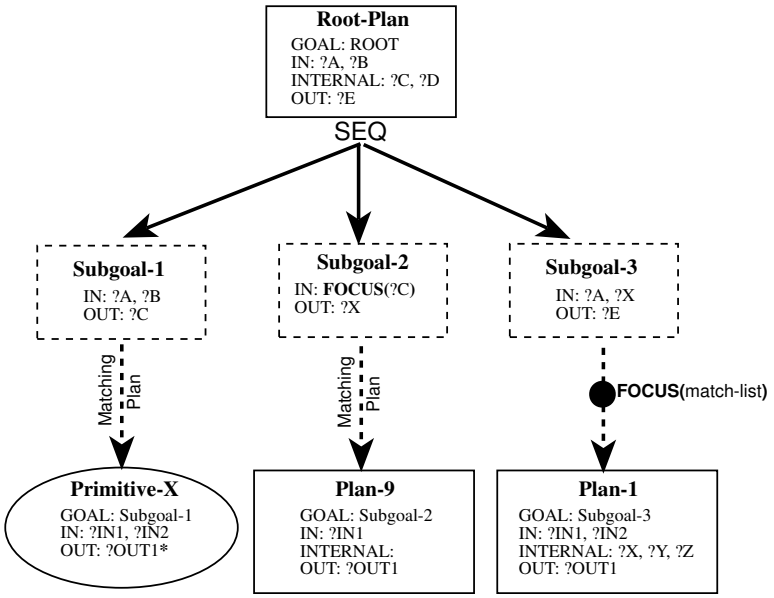


Figure 2.3. Control Plan and Focusing Heuristics.

A control plan is a plan schema that specifies either a set of subgoals that must be met in order achieve the plan's goal, or a primitive action (e.g. blackboard KS or other code for manipulating objects on the blackboard or planner's data structures) that can be taken immediately to achieve the plan's goal. As shown in Figure 2.3, a control plan's definition includes a GOAL form indicating the predicate form of the goal that the plan is intended to satisfy. The RESUN planner uses this information to retrieve any relevant plan from its plan library, then logically unify the goal form with the plan's goal form and variables. IN and OUT plan variables are used to pass information to higher-level plans, while INTERNAL variables are used to pass information among a plan's subgoals. RESUN's subgoal grammar supports the specification of rather complex scheduling relationships and dependencies among the subgoals of a plan. To aid the reader in understanding the structure of IPUS control plans, Figure 2.4 illustrates the five common subgoal relationships used in plans for IPUS:

(A) Subgoals must be achieved in sequential order left-to-right.

(B) Subgoal-1 must first be achieved, then depending on the value returned by g(), either subgoal-2 ($g(?C) = 0$) or subgoal-3 ($g(?C) = 1$) must be achieved.

(C) Subgoal-1 and subgoal-2 both must be achieved, but in any order.

(D) Subgoal-2 needs to be achieved only if the condition is true.

(E) Subgoal-2 must be repeatedly achieved until the condition is false.

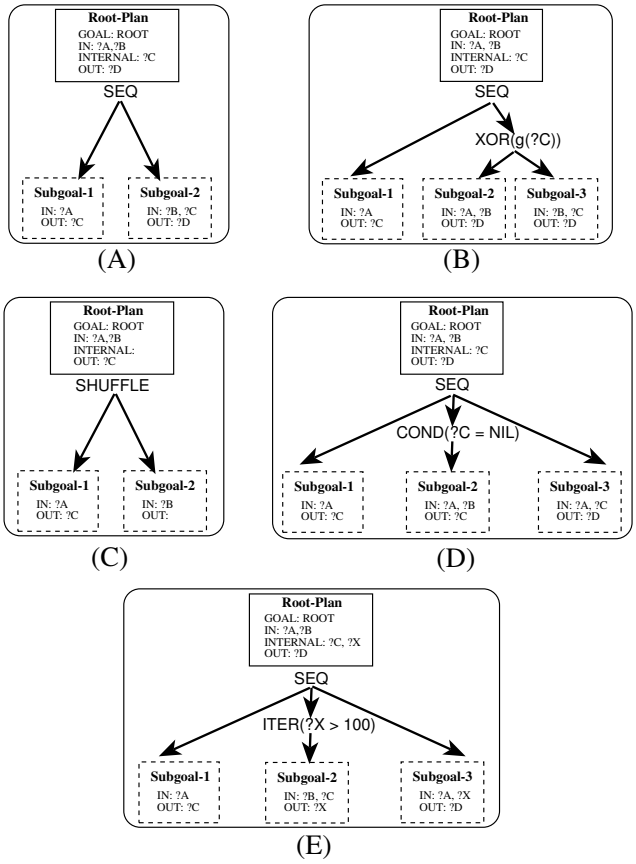


Figure 2.4. RESUN Subgoal Relationships

Focusing heuristics are context-sensitive tests for making a choice when there is more than one way to refine a plan. Figure 2.3 shows the two types of focusing heuristics used in IPUS:

variable-focusing heuristics and match-focusing heuristics. To see the role of variable-focusing, note that Primitive-X returns an *uncertain value* (?OUT1) which means that Subgoal-1's ?C variable could have two or more possible values. A variable-focusing heuristic is applied to uncertain-valued variables when an attempt is made to use them in another subgoal (Subgoal-2 in Figure 2.3's case). To see the role of match heuristics, note that Plan-1 was not the only plan in the control-plan database that could possibly satisfy Subgoal-3. A match-focusing heuristic, however, selected Plan-1. In both types of focusing, the decision is not final; if a subgoal could not be achieved with the first result of a focusing heuristic, the planner can (if the user so specifies) re-apply the heuristic to obtain the next best choice.

Most significantly for IPUS, RESUN is an incremental planner, which means that plans are only expanded into constituent subgoals until the next primitive action that can be executed is identified. The action is then executed, and its effects on the blackboard or its output plan-variable results are used to update planner structures before more planning is performed. This has the important consequence of interleaving planning and KS execution, which in turn makes possible the kind of context-dependent behavior the thesis seeks for IPUS.

The RESUN framework was developed to address existing interpretation systems' limited ability to express and react to the reasons for interpretation hypotheses' uncertainty. It emphasizes the separation of hypothesis belief evaluation from control decision evaluation by making control responsive not only to the levels of numeric belief in hypotheses but also to the presence of specific SOUs in the problem-solving model or on hypotheses. The refocusing formalism supports opportunistic control by enabling the planner to switch among several plan elaboration points (current leaf nodes in the plan tree) in a context-dependent manner. RESUN facilitates two basic problem-solving modes: *evidence aggregation* and *differential diagnosis*. Problem-solving for evidence aggregation seeks data for increasing or decreasing the certainty of one particular interpretation, whereas problem-solving for differential diagnosis seeks data for resolving ambiguities (uncertainties) that produced competing interpretations. IPUS-based systems need to use both problem-solving approaches in deciding when to reprocess

data previously examined under one SPA with another SPA to obtain evidence for resolving uncertainties.

2.3 Generic Architectural Strategy

The generic IPUS architecture, with its primary data and control flow, appears in Figure 2.5. Solid arrow lines indicate dataflow relations. Dotted arrow lines indicate classes of plans that the planner can pursue when trying to reduce or eliminate particular uncertainties (discrepancies) in the problem solving model that were selected by the focusing heuristics. Note that the figure shows reprocessing being done only on the lowest-level SPAs for clarity's sake. Within IPUS, reprocessing can cause SPA execution at any SPA output level, not just the lowest. Two types of signal interpretation hypotheses are stored on the hierarchical blackboard: interpretations of correlates from current and past signal analyses, and expectations about the interpretations of data correlates from future analyses.

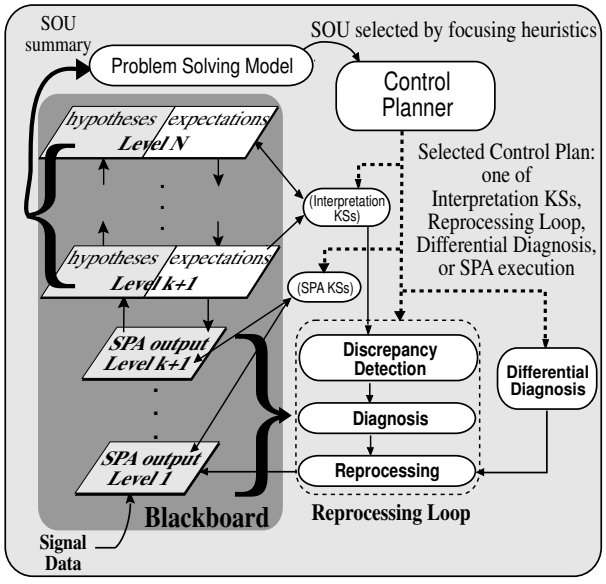


Figure 2.5. The Abstract IPUS Architecture

The IPUS architecture is designed to support variations on the following four-phase general control strategy. Within the first phase, each block of signal data is processed with an

initial configuration of SPAs (initial front end), producing data correlates at various blackboard abstraction levels up to some cutoff level. As the initial front end is executed, discrepancy detection tests are performed to check for discrepancies between each SPA's data correlates and (1) the correlates of other SPAs in the configuration, and (2) application-domain constraints. Depending upon the system designer's intentions, IPUS provides the ability either to proceed to the second phase of execution, to act immediately upon detected discrepancies at this point (*third* phase), or to interleave these behaviors. In the second control-strategy phase, IPUS applies signal interpretation algorithms (SIAs) to the front end results both to confirm support for expectations from previous data blocks' interpretations and to generate new hypotheses to explain unexpected correlates' occurences.[2] In the course of confirming expected correlates, IPUS performs a third set of discrepancy-detection tests to compare the initial front end's output with the correlates' anticipated form based on high-level expectations. During the third general control phase IPUS selects various discrepancies, generates diagnostic explanations for them, and, if explanations exist, searches for and applies new front ends to eliminate or reduce the discrepancies. As before, IPUS permits a system designer to interleave actions from this phase with actions from the second phase. In the fourth and final control-strategy phase, if there are any similarly-rated competing top-level interpretations, IPUS can perform differential diagnosis to select and execute reprocessing plans to find correlates for features that will discriminate among the alternatives. At the end of this phase, IPUS also allows the system designer to specify methods for updating the initial front end for the next data block so as to prevent reprocessings for predictable discrepancies.

2.4 Basic IPUS Machinery

This section describes both the RESUN elements and the IPUS-specific mechanisms that provide the framework for the general IPUS architecture. The first subsection describes the role of RESUN's SOU concept in IPUS, while the second subsection presents the basic control

[2] for example, new tracks from a newly-started sound or a new shadow from an object that has just moved into a camera's field of view.

plans that implement IPUS. The third subsection concludes with a discussion of the IPUS mechanisms for supporting processing contexts and adaptable front ends.

2.4.1 SOUs and IPUS

The key to using RESUN to implement the general IPUS control strategy yet provide generality for system designers using IPUS in various domains lies in RESUN's SOU concept. It is the generality of the RESUN SOU language that gives IPUS its own generality. IPUS requires a system designer to specify control decision tests using both application-specific focusing heuristics and application-specific control plans. These tests must be expressed in terms of what kinds of SOUs are present in the PSM or on a particular hypothesis.

There are three general categories of SOU that IPUS borrows from RESUN, which are named according to where SOUs in the category can be found within IPUS:

1. **Problem-Solving-Model SOUs** These SOUs are used to summarize the general state of the current interpretation. IPUS makes use of four RESUN SOUs in this category:

   (a) *No-Evidence*: Summarizes where the current interpretation is uncertain because there is unprocessed signal data beyond the current data block.

   (b) *Uncertain-Hypothesis*: Summarizes which hypotheses in the current interpretation are uncertain and why. This SOU is used with symbolic interpretation hypotheses that are not at the "answer" blackboard abstraction level (the level of hypotheses at which a system is supposed to return results).

   (c) *Uncertain-Answer*: This SOU is used with symbolic interpretation hypotheses that are at the "answer" blackboard abstraction level, and that have sufficient (though possibly incomplete) support to be believed. Summarizes which believed answer-level hypotheses are uncertain and why.

   (d) *Uncertain-Nonanswer*: This SOU is used with symbolic interpretation hypotheses that are at the "answer" blackboard abstraction level, and that have sufficient

(though possibly inaccurate) negative evidence to be **dis**believed. Summarizes which disbelieved answer-level hypotheses are uncertain and why.

2. **Hypothesis SOUs** These SOUs are used to express general reasons for why a particular hypothesis is uncertain. IPUS adopts eight from the RESUN SOU language, and uses a ninth SOU specific to the architecture:

   (a) *No-Support*: This SOU represents the uncertainty in a hypothesis that arises because it was generated through top-down inferencing and not any support hypotheses have yet been sought for it.

   (b) *No-Explanation*: This SOU represents the uncertainty in a hypothesis that arises because no bottom-up inferencing has yet been done to explain it as a feature of (i.e. evidence for) a higher-level hypothesis.

   (c) *Partial-Support*: This SOU represents the uncertainty in a hypothesis that arises because not all of the potentially available support evidence for it has been searched for.

   (d) *Support-Exclusion*: This SOU represents the uncertainty in a hypothesis that arises because some subset of its required support evidence has been sought for but not found, which raises the possibility that the evidence may not in fact exist

   (e) *Possible-Alternative-Explanation*: This SOU represents the uncertainty in a hypothesis that arises because there exist explanations for it that have not been considered due to pruning decisions in the interpretation search space.

   (f) *Possible-Alternative-Support*: This SOU represents the uncertainty in a hypothesis that arises because there exist other classes of support that have not been considered due to pruning decisions in the interpretation search space or the front-end search space.

   (g) *Alternative-Extension*: Within RESUN (and IPUS), to represent the fact that a hypothesis at one abstraction level is supported by one or more other hypotheses at

lower abstraction levels, inference structures that link the hypotheses' data structures are created. When the inference structure is created, copies of the involved hypotheses' data structures are also created with updated attribute values according to the constraints of the inference. These copies are called *hypothesis extensions*. The alternative extension SOU represents the uncertainty in a hypothesis that arises because it has two or more extensions that support higher-level competing hypotheses.

(h) *Uncertain-Support*: This SOU summarizes the uncertainty that a hypothesis has due to the uncertainty in the support hypotheses upon which it depends. Within IPUS this SOU serves as a "placeholder" in which focusing heuristics can find summary information about the uncertainty in the support for a hypothesis.

(i) *Constraint-Inconsistency*: This SOU defined for IPUS represents the uncertainty in a hypothesis that arises because one or more domain-dependent relationships between it and other hypotheses have not been observed. For example, assume that the local maxima selected by a peak-picking SPA from a spectrum hypothesis must account for 50% of the spectral energy. If this constraint is not met, the spectrum hypothesis will be annotated with a constraint-inconsistency SOU.

3. **Inference SOUs** IPUS includes two SOUs in this category for representing uncertainty about the validity of inferences between support hypotheses and conclusion hypotheses.

(a) *Constraint*: Occasionally some constraint in the application-rule that permits a support inference to be made is "moderately" violated. For example, assume a domain-specific rule that states that a set of frequency tracks in a spectrogram can be considered to represent (i.e. provide support for) a harmonic set if they are all integer multiples of a fundamental frequency. Depending on the application domain's processing requirements, this rule might be considered to have been "moderately" violated when a set of tracks is used as support for a harmonic

set even though their frequencies each are 5 to 10 percent off from true integer multiples. *Constraint* SOUs represent the uncertainty in the inference caused by such situations.

(b) *Possible-Inappropriate-Parameter*: This SOU represents the uncertainty in a negative inference that support hypotheses might be missing or distorted due to signal processing with an inappropriate front end, rather than because the support is not present. For example, an inference recording that no spectral peaks were found for an expected track hypothesis would be annotated with this SOU if the track's possible range of expected energy included values below an energy threshold parameter used by the peak-picking SPA for identifying spectrogram values as peaks.

2.4.2 Basic IPUS Control Plans

Figure 2.6 depicts the control-plans used to implement the most general control in IPUS: PSM-based control. After an IPUS system is initialized, it will iterate within the `Resolve-PSM-Uncertainty` plan until some iteration's achievement of the `Have-PSM-SOU-Resolved` subgoal results in the absence of a *No-Evidence* SOU in the PSM. In instantiating IPUS, the user must specify a variable focusing heuristic function for selecting the value of ?psm-sou from all sous in the PSM. Once a value is established, predefined match focusing heuristics based on the type of PSM SOU selected select a plan from those in Figures 2.7 through 2.11.

The "No Evidence" control plan (Figure 2.7) effectively implements the first control-strategy phase as well as the end of the fourth control-strategy phase defined earlier in this section. The "Uncertain Hypothesis" control plan and the "Uncertain Answer" control plan (Figures 2.8 and 2.9) will be seen to provide the hooks for users to add control plans implementing the remaining control-strategy phases. The "Uncertain Nonanswer" control plan (Figure 2.10) is provided as a hook for adding control plans that can re-evaluate certain disbelieved answer hypotheses when (for example) new signal data becomes available, while the "Solve PSM

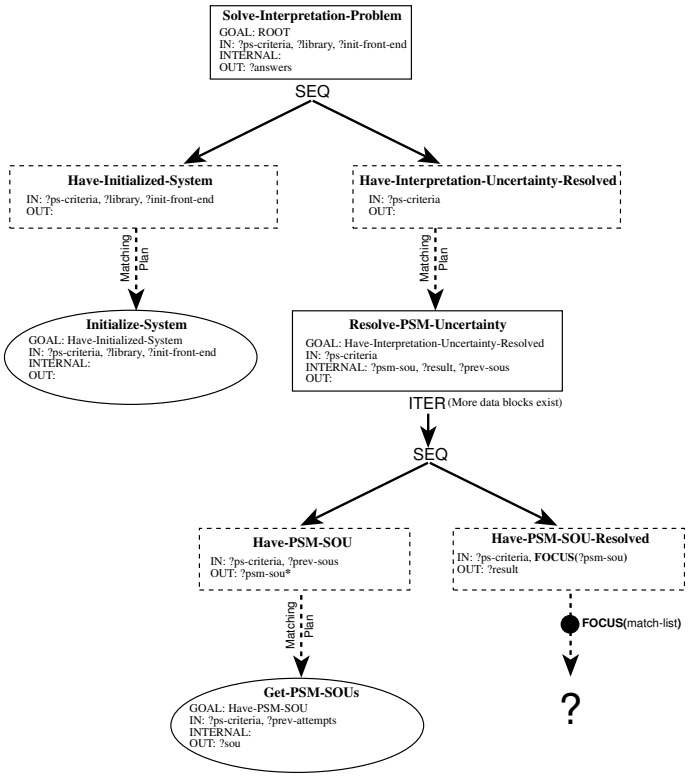


Figure 2.6. Highest-Level IPUS Control Plans.

SOU List" control plan (Figure 2.11) is provided so that the system designer is free to have the PSM-SOU focusing heuristic return more than one SOU to be handled simultaneously. The ?ps-criteria input variable for these four plans is a means for making available a data-structure capturing high-level problem-solving constraints that govern the long-term behavior of a system, such as minimum required rating values for determining if a hypothesis is believed, or names of critical answer-level hypotheses that should be considered before all others. Such problem-solving criteria could have been incorporated in focusing heuristics, but they would then be static and unalterable for the duration of the system's execution. Maintaining them in a modifiable data structure allows system designers to specify control-plans that could modify these constraints if the system's environment changed from what it was when then system was originally deployed.

At this point it is instructive to examine how the PSM-level control plans produce the general processing-strategy stages identified in Section 2.3. The discussion will first focus

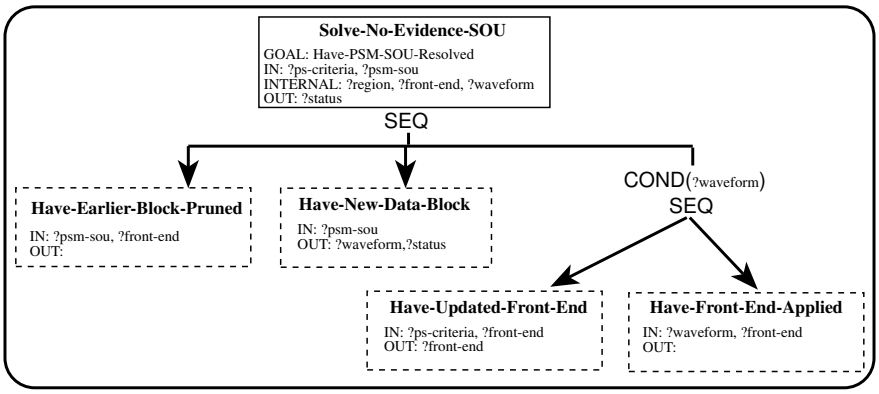


Figure 2.7. "No Evidence" Control Plan.

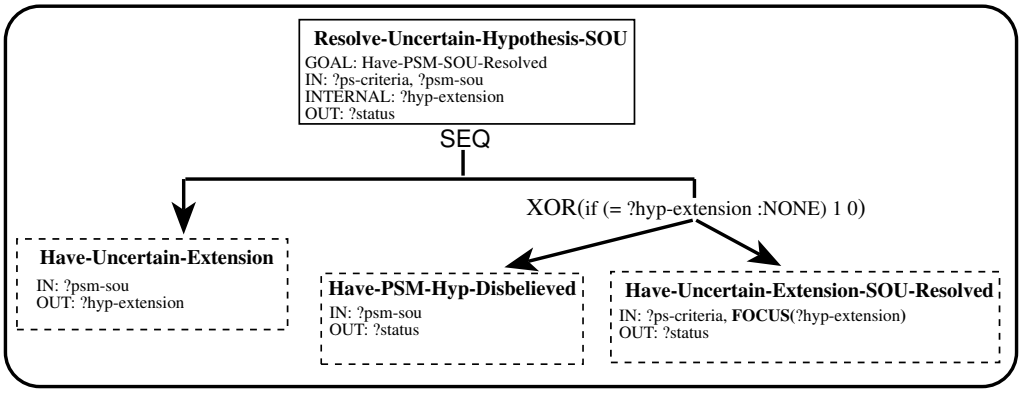


Figure 2.8. "Uncertain Hypothesis" Control Plan.

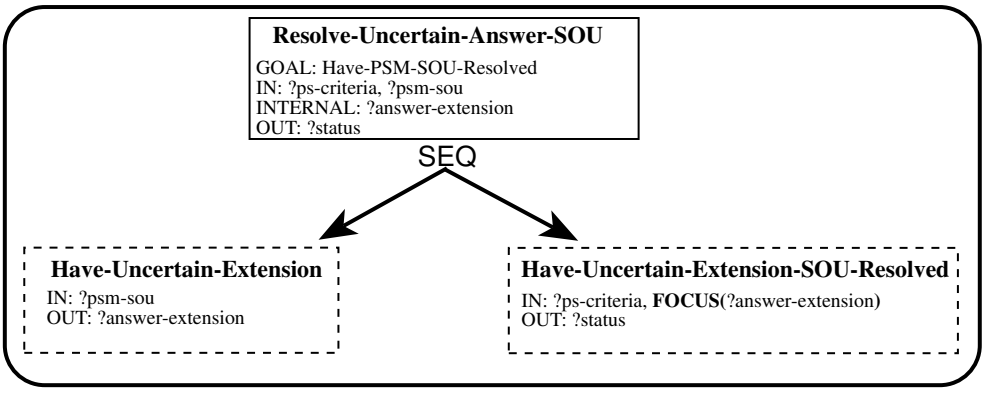


Figure 2.9. "Uncertain Answer" Control Plan.

on the "No Evidence" control plan. The *Have-Earlier-Block-Pruned* subgoal of the "No Evidence" control plan provides system designers with a hook for incorporating criteria and strategies for simplifying the current interpretation before the next data block is examined. The "Simplify Interpretation" control plan in Figure 2.12 provides a general strategy for achieving the subgoal. This plan can be customized through focusing heuristics and domain-dependent plans and primitives for its subgoals.

The overall strategy shown in Figure 2.12 has five stages. The first stage is represented by the *Have-Answer-Summary* subgoal, and involves the summarization of the answer-level hypotheses for the most recent data block into three categories: viable, dropped, and suspended. Viable hypotheses are those answer-level hypotheses that are believed, while dropped hypotheses are those answer-level hypotheses that have been disbelieved, or "pruned" from the interpretation,

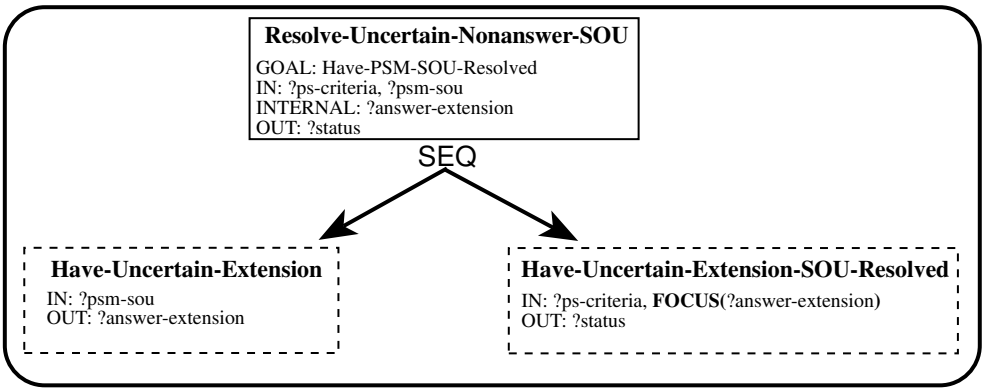


Figure 2.10. "Uncertain Nonanswer" Control Plan.

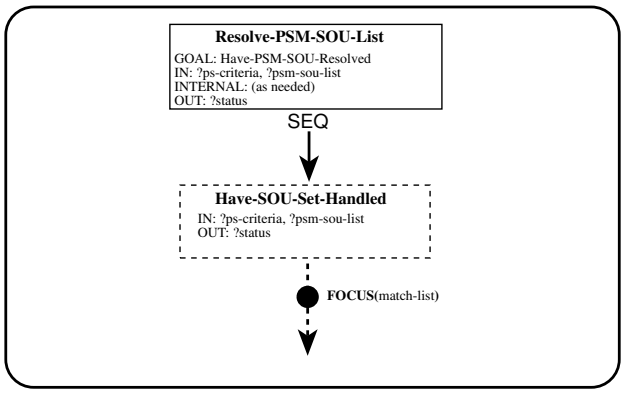


Figure 2.11. "Solve PSM SOU List" Control Plan.

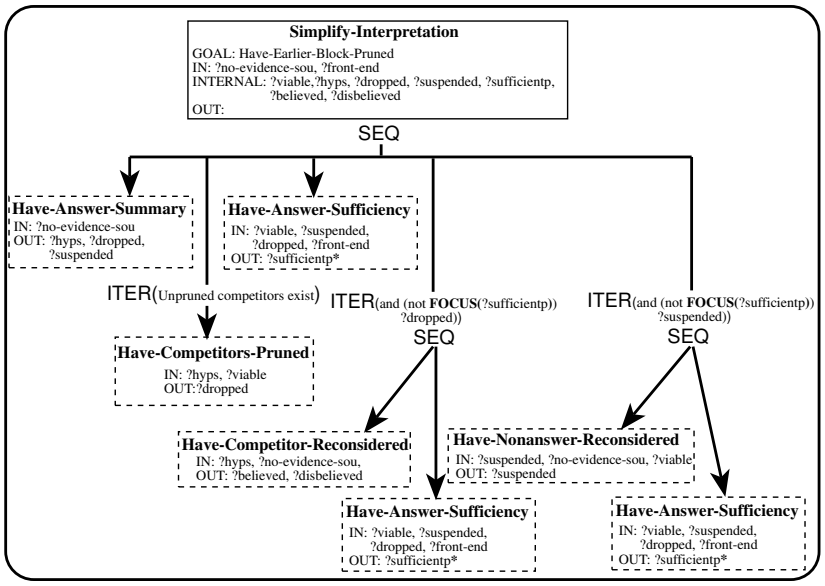


Figure 2.12. "Simplify Interpretation" Control Plan.

in favor of a higher-rated competing hypothesis. Suspended hypotheses are those answer-level hypotheses that were disbelieved or eliminated from further consideration before all possible evidence for them had been sought because of a decision in some search-pruning focusing heuristic. The second stage is represented by the *Have-Competitors-Pruned* subgoal, and entails iteratively disbelieving hypotheses whose competitors have higher belief,[3] until all lower-ranked competitors have been removed from consideration. The third stage is represented by the first *Have-Answer-Sufficiency* subgoal, which is achieved through a domain-specific focusing heuristic that determines the adequacy of the current data block's interpretation (i.e. selects a true or false value for the ?sufficientp variable). "Adequacy" in this context refers to some set of domain-specific features that define the stopping criteria for problem-solving on a data block. When these criteria are met, the interpretation is assumed to be complete for the current data block, and the system can proceed to work on the next data block. The fourth and fifth "Simplify Interpretation" stages involve work on repairing interpretation inadequacies; their order can be switched according to application-domain requirements. In the iteration

[3]The belief metric can include tests for SOUs as well as numeric ratings.

that reconsiders dropped hypotheses, the first subgoal involves the determination of whether in fact a dropped hypothesis **and** its competitor are both present in the signal, in which case a pruning decision based on arbitrarily different ratings would be superseded. This subgoal can be achieved by the IPUS-supplied differential diagnosis control plan to be discussed in Subsection 2.5.4 or by domain-specific plans supplied by the system designer. In the iteration that reconsiders suspended hypotheses, the first subgoal involves the determination of whether in fact a suspended hypothesis had been discarded from consideration too soon, in which case a pruning decision based on an arbitrary rating at an arbitrary time would be superseded. This subgoal can be achieved by control plans whose subgoals require the IPUS-supplied "Uncertain Nonanswer" control plan in Figure 2.10 or by domain-specific plans supplied by the system designer.

Continuing the elaboration of the subgoals of the PSM-level control plans, this subsection turns to the "Uncertain Hypothesis" and "Uncertain Answer" control plans, postponing discussion of the last two subgoals of the "No Evidence" plan to the next subsection. This is done since the material in Subsection 2.4.3 strongly interacts with the design of the plans for those subgoals. In general, the plan for uncertain-hypothesis PSM SOUs (Figure 2.8) leads to "bottom-up" system interpretation behavior or to diagnosis and reprocessing. That is, achievement of the plan's subgoals involves either the explanation of hypotheses at lower abstraction levels as evidence for higher-level hypotheses, or the application of the reprocessing loop to resolve discrepancies noted among hypotheses produced by the front end. The first subgoal requires the selection of one extension when a hypothesis has more than one (Subsection 2.4.1, *Alternative-Extension*), and may involve focusing heuristics. The branch at the end of the plan permits the system designer to have the hypothesis disbelieved if no extension is found whose uncertainty can be resolved. This action might be justified, if, for example, the selection of an extension involved preliminary tests for whether the candidate extension could be used to hypothesize any higher-level or answer-level hypotheses. Failure to pass such a test could be an indication that the hypothesis represented a bottom-up derived "hallucination" that should be

disbelieved, or removed from further consideration by the interpretation processes.[4] The second subgoal of the branch is handled by the short plan in Figure 2.13, whose subgoals require first the selection of some hypothesis SOU through a focusing heuristic, and second the resolution of the selected SOU. When a *No-Explanation* or *Possible-Alternative-Explanation* SOU is selected, domain-dependent control-plans are considered that will cause the problematic hypothesis to be explained as evidence for a higher-level or answer-level hypothesis. These "No Explanation" plans implement the part of the second general control-strategy phase that generates explanation hypotheses for unexpected hypotheses from SPA outputs, or generates preliminary answer-level explanations. When, however, a *Constraint-Inconsistency* SOU is selected, domain-dependent control-plans are considered that will perform discrepancy diagnosis and reprocessing for discrepancies detected in front-end results. These plans implement part of the third general IPUS control-stategy phase.

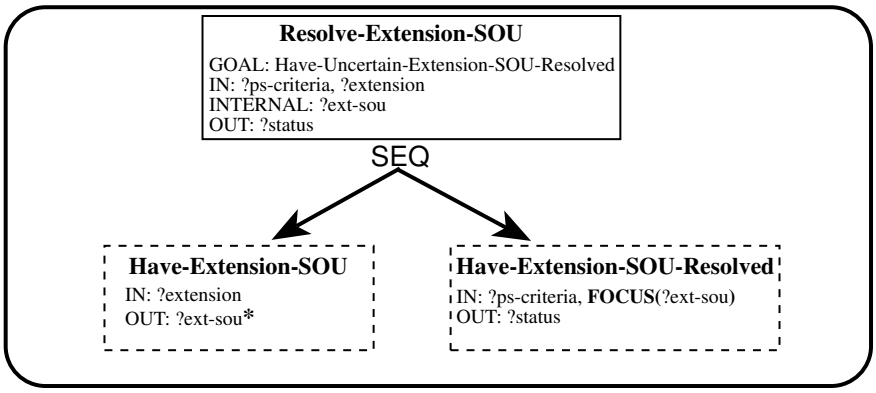


Figure 2.13. "Resolve Extension SOU" Control Plan.

The "Uncertain Answer" PSM-level control-plan involves the achievement of subgoals analogous to those associated with the "Uncertain Hypothesis" plan, but the SOUs that it leads

[4]Of course, this conclusion is completely reasonable only if the system designer can assume that all possible types of hypotheses at the answer-level are known to the system (i.e. all possible perceptual objects are modelled in the system's model library). When this assumption is not plausible, that is, there may be unknown objects whose structure (hypotheses below the answer-level) might be discerned but whose identity may be unknown, the system designer must exercise care in permitting consideration of the first XOR subgoal.

the planner to consider resolving on answer-level hypotheses or their supporting hypotheses are different. The planning associated with this plan's subgoals will cause an IPUS system to resolve *No-Support*, *Partial-Support*, *Possible-Alternative-Support*, or *Uncertain-Support* SOUs indicating unsought support evidence at lower support-levels if focusing heuristics decide that the system should be engaged in the second part of the second phase of the general IPUS control stategy. That is, the system attempts to verify expectations from high-level interpretation hypotheses. Should focusing heuristics select *Support-Exclusion*, *Alternative-Extension*, or *Uncertain-Support* SOUs indicating low-quality support, domain-dependent plans implementing the remainder of the third general control strategy phase are considered. That is, planning is performed to apply the reprocessing loop to resolve discrepancies detected between interpretation expectations and front-end outputs.

2.4.3 IPUS and Front Ends

This subsection discusses how SPAs are defined and embedded within an IPUS-based system and how the processing-context mechanism supports their manipulation.

Before discussing the implementation of SPAs within IPUS, it is important to review the context-dependent nature of certain SPA executions. This review will serve as background for the reasons why various information categories about SPAs are included in IPUS. For the purposes of IPUS, the features that perceptual systems monitor in complex environments can be divided into two classes. The first class contains features which can be used to indicate the existence of one or more perceptual objects, though not necessarily the identities of those objects. These features often have supporting correlates that can be computed independent of the context being analyzed. In the auditory domain, for example, any collection of one or more "sound objects" may be conceptualized as an acoustic intensity distribution with minimum and maximum limits on gross features such as temporal spread, frequency spread, duration of silence intervals, and degree of randomness in intensity fluctuations. The SPA correlates for supporting such gross features can generally be computed in a context-independent manner; hence they are termed *context-independent features*.

The second feature class contains those features which can be used to identify an object or monitor the behavioral changes of an object. The computation of correlates to support these features is often very sensitive to the context being analyzed; hence they are termed *context-dependent features* in the IPUS design framework. In the auditory domain, for example, a frequency track would be a context-dependent feature of a sound ("acoustic object"). As Figure 2.14 illustrates, if the current interpretation for a scenario contains only one sound (sound A) having spectral energy at frequency f1, then even an STFT SPA with an analysis window length providing only very coarse frequency resolution (STFT-1, in addition to the other STFTs shown) would still produce correlates (a sequence of high-energy spectrogram values) that could support the claim that track $T_{A_{f1}}$ exists. However, as in Figure 2.15, an interpretation that assumes another sound (sound B) in the environment with spectral energy at a frequency f2 near f1 creates a new context with different processing needs. In this new scenario only STFTs providing frequency resolution of at least the minimum difference between f1 and f2 (STFT-3 only) will produce correlates that could unambiguously support the existence of $T_{A_{f1}}$ (or, for that matter, the existence of $T_{B_{f2}}$). Figure 2.16 uses an instance of an environment suddenly changing from having one sound-source to having two sound-sources to provide a more concrete example of the context-dependent nature of track detection in a spectrogram in a real-world system.

It is important to note that the distinction between context-independent and context-dependent features lies in the usage of the features. If a feature is used only to indicate the *presence* of some object(s), the feature is considered context-independent. However, if the same feature were to be used as support for the *identity* of some object(s), it would in general require context-dependent correlate computation, and would therefore be considered a context-dependent feature.

Each time an SPA is executed within IPUS, the blackboard hypotheses representing the output correlates from the execution are annotated with the name of the SPA and the control parameter values used in the execution. This annotation is a data structure referred to as

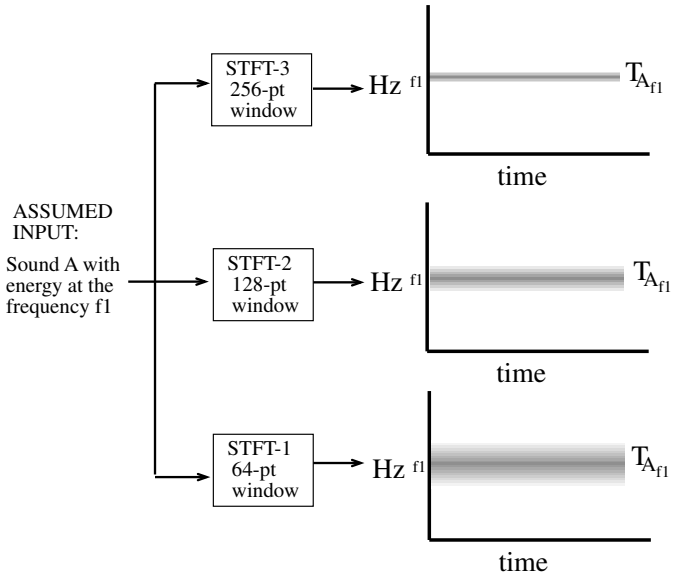


Figure 2.14. Context-Dependent Feature Example Part 1.

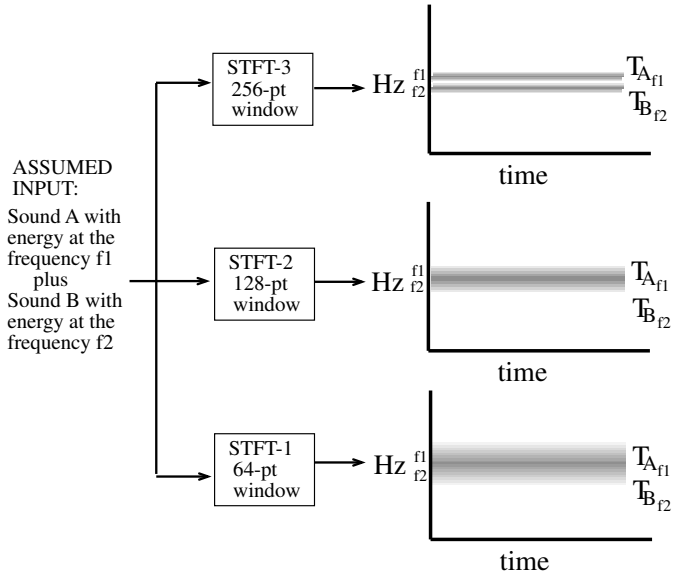


Figure 2.15. Context-Dependent Feature Example Part 2.

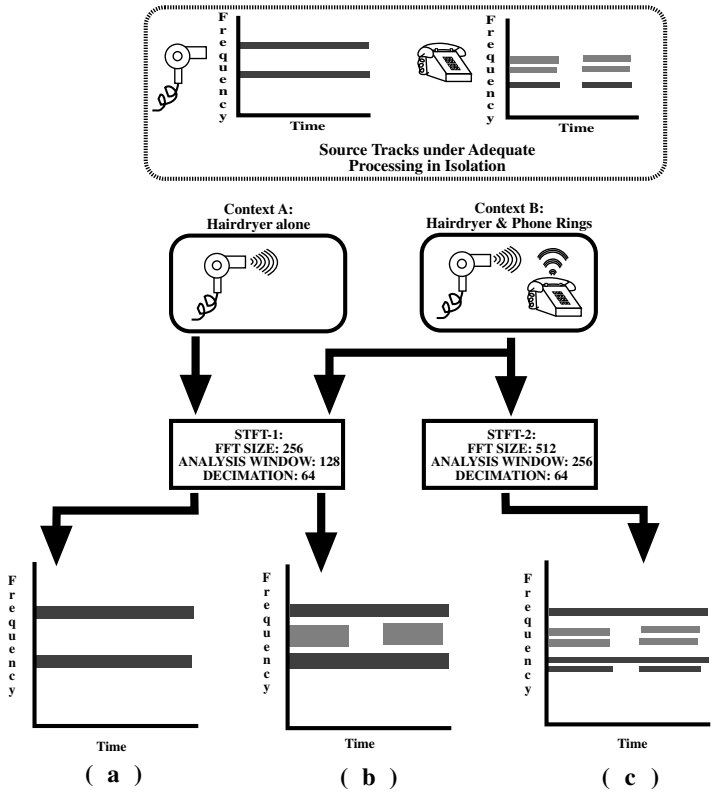


Figure 2.16. Context-Dependent Example Part 3.

the correlates' *parameter context*. In addition to the parameter context, each SPA correlate is annotated with a *processing context*, or a data structure listing the sequence of SPAs that produced the correlate from the input signal.

In IPUS SPAs are defined with six elements. The first element is the actual code for the algorithm. The second element is the list of specific parameters that control how the algorithm produces output correlates, and bounds on the range of values each can take on. The third element is a set of rules defining how individual control parameters should be modified to eliminate or reduce various classes of distortions that could be manifested in the algorithm's correlates. The preconditions of these rules are expressed in terms of one or more distortions, while the conclusions or actions of these rules specify methods for computing parameter values to ameliorate instances of the distortions. These rules actually are stored with reprocessing strategies. The concept of "distortion" is discussed further in Subsection 2.5.2. The fourth

element of an SPA definition is a set of discrepancy tests. These are application-dependent tests that compare the output of one SPA with the output of another SPA for consistency according to theoretical signal processing constraints. The fifth definition element is a list of "supercontext methods" that take as input a parameter-context and an optional "information category" label. These methods return patterns indicating the range of values for each control-parameter in an SPA parameter-context that would permit the SPA to produce correlates having the same or greater detail in the specified "information category" as found in the specified parameter context. The returned patterns[5] must designate all SPAs which produce the same type of output as that produced by the SPA specified in the supplied parameter context, regardless of whether they share the same control-parameter sets. As an example, assume an SPA that selects local maxima from a spectrogram on the basis of whether their energy values are greater than a threshold control-parameter. The supercontext method for this SPA would, when supplied with a particular parameter context and the information category "peaks," return a pattern indicating that any execution of the SPA with a parameter context having threshold values below that of the given parameter context would provide at least the same number of peaks as were produced by the given parameter context. The sixth and final SPA definition element is a mapping function that takes two parameter contexts and a list of correlates produced from the first context, and returns a list of the correlate hypotheses modified to reflect how they would appear had they been produced by the second context.

Figure 2.17 shows the basic control plan used by IPUS for executing an SPA. The first subgoal, *Have-Supercontext-Output*, encourages the reuse of correlates from earlier SPA executions that could provide the same "information" as would be provided by the execution of the specified SPA with the parameter values specified in the ?context variable. This subgoal involves the application of supercontext tests. The ?context variable can hold either a parameter context, a processing context, or a front end, which is a list of parameter-contexts specifying the default control-parameter values to be used by the front end. In general, reuse occurs

[5] In the IPUS Sound Understanding Testbed these are simply parameter contexts with range-values for the control-parameters.

during reprocessing, and not during the first execution of the SPAs in the default front end. The second subgoal updates the region of application for the SPA based on what data was found that could be reused. The *Have-SPA-Executed* subgoal applies the SPA with the control-parameter values specified for it in the ?context variable. The final subgoal involves the application of discrepancy-detection tests for comparing outputs from the SPA for consistency with the outputs from other front-end SPAs or with environmental constraints.

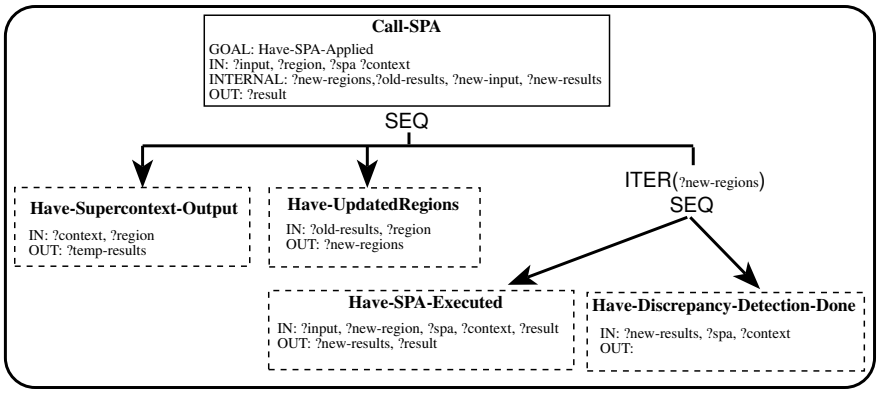


Figure 2.17. "Call SPA" Control Plan.

The plan for executing the front end in IPUS is domain-dependent and must be specified by the system designer as a plan with COND clauses with calls to the *Have-SPA-Applied* subgoal. The COND test will check for whether the front-end SPA list contains a parameter-context with the name of the SPA the clause's subgoal would cause to execute. The *Have-Updated-Front-End* subgoal of the **Resolve-No-Evidence-SOU** general IPUS control plan (Figure 2.7) involves the execution of rules for checking how often the current front end has caused diagnosable discrepancies and deciding whether a change in front-end SPAs or a change in SPA control-parameters is necessary to reduce the time spent in reprocessing. If changes are necessary, this subgoal will effect changes in the front-end SPA list.

2.5 IPUS Reprocessing Loop Components

This section presents the generic specifications of each component of the IPUS reprocessing loop as depicted in Figure 2.5, and, where appropriate, describes the generic control-plan implementation for the component.

2.5.1 Discrepancy Detection

The discrepancy detection process is crucial to the IPUS architecture's iterative approach. The process is required to recognize three groups of discrepancies, based on the source of the expectation (i.e. ground truth) used in the comparisons.

fault A discrepancy between an SPA's computed correlates and correlates from other SPAs applied to the same signal data. This class is included based on two propositions. The first is that correlates for context-dependent features, if computed by SPAs appropriate to the context, do not contradict the correlates for context-independent features. The second is that correlates for context-dependent features, if computed by SPAs appropriate to the context, do not contradict other context-dependent correlates computed by other SPAs from the same data. Figure 2.18 illustrates this type of comparison with an acoustic example comparing the context-independent output of a time-domain energy-tracking SPA with the context-dependent selection of peak values from the output of an STFT SPA. The energy tracking SPA indicates a short burst of energy while the first STFT's correlates do not support new frequency tracks during the burst's time period. A fault should be declared in this case since Fourier theory requires indications of the burst's presence in both analyses,[6] given the assumption that the peak-picking analysis done on the STFT's output was appropriate to the environment.

violation A discrepancy detected between an SPA's data correlates and domain constraints. This class is included based on the proposition that correlates, if computed by SPAs appropriate

[6] Specifically, Parseval's Theorem $\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$ requires that time-domain energy (lefthand side) be conserved in the frequency-domain (righthand side).

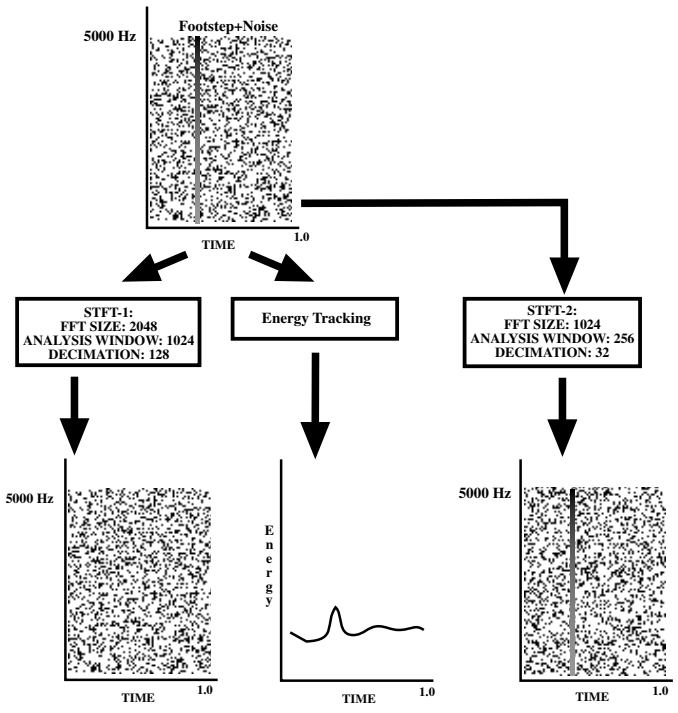


Figure 2.18. Fault Discrepancy Example.

to the context, do not support features that violate the environment's physical constraints. As an example, if the application domain is considered subject only to wideband gaussian noise (5000 Hz wide), STFT output correlates showing only a narrowband noise signal (say 500 Hz wide) would give rise to a violation. Note that violations can indicate either that an SPA was inappropriately applied or that the environment's characteristics have changed from those in the assumed in the current interpretation. In the first case reprocessing based on the interpretation should succeed in eliminating the discrepancy. In the second case reprocessing based on the invalid interpretation should fail.

conflict A discrepancy between an SPA's computed correlates and model-based expectations. Model-based expectations arise from two sources. The first source is the set of models for objects already assumed to be present. The second source is the set of models for objects under consideration for interpreting newly-detected correlates in the current block of data. Conflict discrepancies may involve either a total or a partial mismatch

between correlates and the hypotheses they were supposed to support. This class is included based on the proposition that features supported by correlates computed from appropriate SPAs ought to be completely consistent with the object features specified by the context expected to be observed. The concept of "object features" includes not only features that are not expected to be distorted but also features that *are* expected to be distorted because of the existence of other objects in the environment. Conflicts can indicate that an SPA is not appropriate to the context or that the context actually contained objects different from those expected. As a simple example, a conflict would occur when the interpretations of past correlates predict a sound with two tracks at 230 Hz and at 250 Hz with no decline in their amplitudes, but peaks picked from the current spectrogram support only one or none of the tracks. This discrepancy could indicate that possibly the peak-picker's energy threshold is inappropriate because the sound's volume decreased, or that a new sound's tracks are overwhelming those of the expected sound.

Fault and violation discrepancies are tested for during the achievement of the *Have-Discrepancy-Detection-Done* subgoal in the "Call SPA" control-plan of Figure 2.17. The tests for these discrepancies are application-dependent and are specified in the "discrepancy-test" element of each SPA's definition. When discrepancies are found, they are represented as *Constraint-Inconsistency* SOUs on the problematic hypotheses. Conflict discrepancies are tested for within SIAs while they search for evidence confirming expectations. If no evidence is found, the conflict is represented as a *Support-Exclusion* SOU on the expectation hypothesis. When there is reason to believe that the lack of evidence could be attributable to use of an inappropriate front end,[7] a *Possible-Inappropriate-Parameter* SOU is attached to the negative inference structure linking the unverified expectation hypothesis with an empty evidence set. On the other hand, if partially supportive evidence is found, the support inference structure between the expectation hypothesis and the support evidence is annotated with a *Constraint*

[7] This determination should be limited to simple heuristic checks of SPA parameter values rather than complex tests involving deep SPA knowledge that would be performed only by the discrepancy diagnosis KS.

SOU. In general, it is the presence of inference SOUs that identifies the presence of conflicts that may be rectifiable by the reprocessing loop. In all cases the SOUs are annotated with the processing contexts used in the failed or partially successful attempts at finding desired evidence.

Examination of a wide range of domains reveals two generic classes of correlates: *point correlates* and *region correlates*. A point correlate is a value associated with one point in the SPA output coordinate space. A region correlate is a value associated with a subset of the SPA output space. Consider the following examples. A spectral peak energy value in the "time, frequency, energy" space of acoustic signal processing and an image pixel intensity value in the "x, y, intensity" space of image processing are examples of point correlates. A noise-distribution tag for a region in a radar sweep and a mean-intensity value for a region in the output of an image filtering SPA are examples of region correlates. A track of spectral peaks over several time slices from a spectrogram is an example of a region correlate comprised of non-contiguous subsets of the SPAs' output space.

For both point and region correlates, the IPUS discrepancy detection component must support tests for the following generic discrepancies between an SPA's anticipated correlate set and its computed correlate set.

1. **missing:** An anticipated correlate is not in the computed correlate set. An example of this discrepancy in the acoustic domain occurs when a spectral peak is expected in the output of an FFT SPA, but is not found.

2. **unassociated:** An unanticipated correlate occurs in the computed correlate set. An example of this discrepancy in the radar domain occurs when an unanticipated clutter region is produced during a radar sweep.

3. **value-shift:** A correlate is found in the computed correlate set at its anticipated coordinates, but with an unanticipated value. In the visual domain this discrepancy would be encountered when an image region's hue label produced by a color-analysis SPA is different from the one expected.

4. **coordinate-shift:** A correlate with an anticipated value is found in the computed correlate set but at unanticipated coordinates. This includes the situation where a region's boundaries shift from their expected locations. An example of this discrepancy in the acoustic domain occurs when a track of spectral peaks produced by a curve-fitting algorithm has the correct energy value but is 30 Hz from its expected position.

5. **merge:** Two or more anticipated correlates are deemed to have appeared as one unanticipated correlate in the computed correlate set. The criteria for this merging are domain-specific and often depend on relationships between the missing correlates' values or coordinates and the unanticipated correlate's value or coordinates. An example of this discrepancy in the visual domain occurs when two adjacent regions with different expected textures are replaced by one region with an unanticipated texture.

6. **fragmentation:** An anticipated correlate is deemed to have been replaced by several unanticipated correlates in the computed correlate set. The criteria for this splitting are domain-specific and often depend on relationships between the missing correlate's values or coordinates and the unanticipated correlates' values or coordinates. An example of this discrepancy in the radar domain occurs when a noise-analysis SPA computes two or more small regions with a particular noise-distribution label instead of an expected single region with that label.

2.5.2 Discrepancy Diagnosis

SPA processing models serve as the basis for defining how the parameter settings of an SPA can introduce distortions into the SPA's computed correlates. These distortions cause correlate discrepancies. Consider an SPA processing model rule relating the STFT's WINDOW-LENGTH parameter to the characteristics of SPA output and how this model can be used to define distortions. Formally, Fourier theory defines the following relationship. Assume that an STFT with a rectangular analysis window of W sample points is applied to a discrete signal that has been sampled at R samples per second. If the signal came from

a scenario containing frequency tracks closer than R/W Hz, the correlates for those tracks in the STFT's spectrogram (values in the spectrogram matrix indicating prominent energy at the track frequencies over time) will be merged and will support at most one distinct track. Referring to Figure 2.16, as the WINDOW-LENGTH parameter's value increases, merged and missing correlate discrepancies between (b) and the tracks expected from the updated interpretation disappear. Conversely, as the parameter's value decreases, merged and missing correlate discrepancies occur more frequently.

The primary formal task of an IPUS discrepancy diagnosis KS is to generate a sequence of distortions that can explain the discrepancies between an *initial state* that represents the assumed ground-truth (e.g. application-specific signal constraints, high-level expectations, or outputs from alternative SPAs whose outputs are less precise but more reliable) and a *goal state* that represents the observed SPA output. In effect the distortion list provides an "inverse mapping" explaining how the ground-truth could have been misrepresented by the front end so as to appear like the observed correlates. Note that there is a difference between discrepancies and signal distortion processes; there is not usually a one-to-one mapping between discrepancies and distortions. It is possible for several distortion processes to explain the same kinds of discrepancies. A "low frequency resolution" distortion explains the "missing" discrepancy in Figure 2.19's example, but a "low time resolution" distortion would explain the "missing" discrepancy between the outputs from the first two SPAs back in Figure 2.18's example.

Within IPUS, the actual discrepancy-detection KS is considered domain-dependent, as is the implementation of the distortion knowledge. It will be seen in Chapter 4 that the Sound Understanding Testbed (SUT) used a discrepancy-detection KS that relied on means-ends analysis to generate a list of distortion operators that reduced the differences between an initial state representing the expected form of front-end data or the expected form of high-level interpretation hypotheses, and a final state that represented the observed data or hypotheses. However, a system designer using IPUS would be free to specify some other KS design, perhaps based on neural nets or cased-based reasoning.

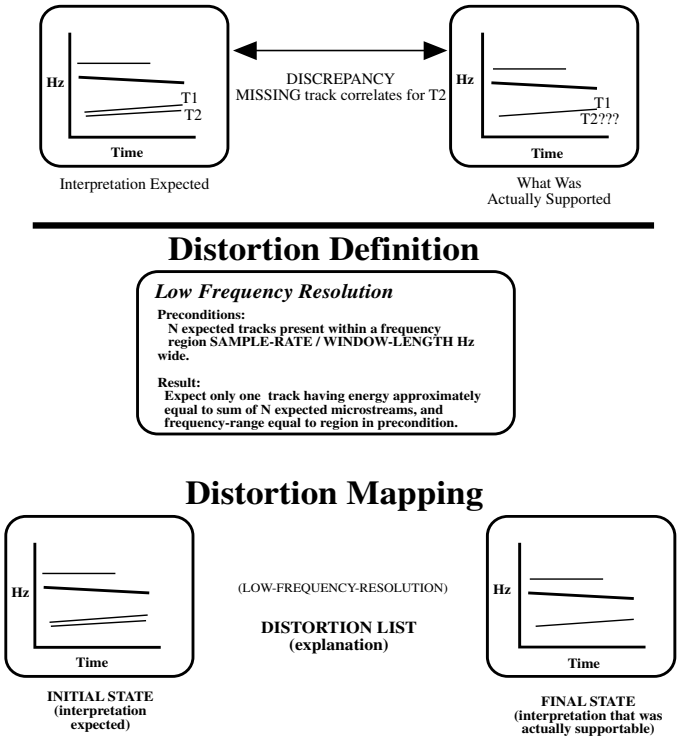


Figure 2.19. Sample Distortion Explanation.

The domain-independent input requirement for the KS is that it accept as input an SOU structure that will point to the hypotheses on which it is stored and that can contain a record of failed diagnosis explanations that did not enable the reprocessing component to resolve the SOU. The domain-independent output requirement for the KS is that it return an explanation structure containing a list of distortions, the processing goals for eliminating them, and the region in which diagnosis was performed. In principle the processing goals depend on the domain. However, in the development of the Sound Understanding Testbed reported in Chapter 4, only two general goals were found necessary. The first is of the form *(Have-Hypothesis-Supported ?hyp)*, which is specified when the SOU under consideration is related to some hypothesis (?hyp) for which no support evidence was found, and the explanation requires that the hypothesis be supported. The second general processing goal is of the form (Have-Hypothesis-Characterized ?hyp ?feature). This goal is specified when the elimination of the distortions merely requires a more refined value for ?hyp's feature named in ?feature, or

the calculation of the feature if it hadn't already been calculated. The SOU being diagnosed is annotated with the explanation structure, permitting the IPUS planner the flexibility of postponing reprocessing for the SOU in case a PSM SOU unrelated to the diagnosed SOU demands attention.

2.5.3 Reprocessing

The *signal reprocessing* component of IPUS uses explanations from the diagnosis components to propose and execute search plans for finding new SPA control parameter values that eliminate or reduce the hypothesized distortions. In the course of a reprocessing plan's execution, the signal data may be reprocessed several times under different SPAs with different parameter values. The incremental search is necessary because the diagnosis explanation is at least partially qualitative, and therefore it is generally impossible to predict *a priori* exact parameter values to be used in the reprocessing. The reprocessing component relies on user-encoded knowledge from SPA processing models to select new SPAs and/or parameter values when instantiating the proposed reprocessing plan.

The input to the general reprocessing control-plan is the diagnostic explanation for resolving an SOU. Strategies for eliminating various distortions are specified as domain-dependent control plans to be selected by the IPUS planner as indexed by the diagnostic explanation. From the retrieved set of applicable plans, one is selected by domain-dependent focusing heuristics. Selections can be governed by criteria such as estimated plan execution time, or the availability of data from earlier executions of SPAs that were supercontexts of SPAs to be executed. The execution of a reprocessing plan consists of incrementally adjusting the control parameters of SPAs within the plan, applying the SPA sequence to the portion of the signal data that is hypothesized to contain distortions, and testing for discrepancy removal. The incremental process is necessary because the situation description is at least partially qualitative, and therefore it is not generally possible to predict exact values for the control parameters to be used in the reprocessing.

Reprocessing continues until the goal of distortion removal is achieved or it is concluded that the reprocessing plan has failed. There are two general criteria for determining plan failure in IPUS. The first criterion simply considers the number of plan iterations. If the number surpasses a fixed threshold supplied with the control-plan definition, failure is indicated automatically. The second criterion relies on fixed lower and upper bounds for SPA processing parameters. If a reprocessing reiteration requires a parameter value outside of its prespecified range, the plan is considered to have failed.

When plan failure is indicated, the diagnosis process can be re-invoked to produce an alternative explanation for the SOU being resolved, if appropriate focusing heuristics for hypothesis-sou-level control plans have been defined. If no alternative explanation is available (i.e., the diagnostic knowledge source fails to find another distortion explanation), the IPUS system annotates the hypotheses involved in the discrepancy with SOUs indicating low confidence due to unresolvable discrepancies. These SOUs' effects on the entities' confidence levels are then propagated to interpretations that were generated from those hypotheses. If the SOUs caused serious enough rating decreases in the affected hypotheses, the associated answer-level hypotheses could be disbelieved.

2.5.4 Differential Diagnosis

In the course of processing signal data, IPUS-based systems will encounter signals that could support several alternative interpretations. In addition to overlapping features from multiple object models, ambiguous sets of alternative interpretations can also arise from interactions among co-occurring objects and from application of SPAs inappropriate to a set of objects. The differential diagnosis component in IPUS must support in these situations the context-dependent selection of features to disambiguate perceptual objects and processing strategies to find those features. The basic control-plan for performing differential diagnosis in IPUS appears in Figure 2.20.

A system designer instantiates the component by supplying a KS as a primitive for satisfying the first subgoal. The specified input for the differential diagnosis KS is the ambiguous data's

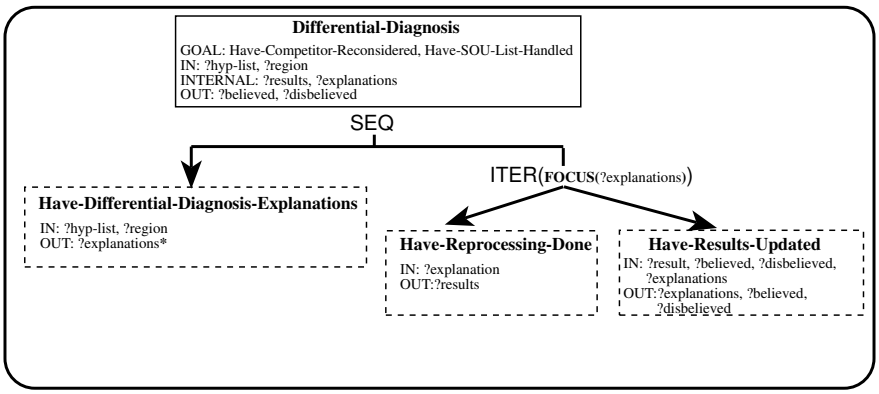


Figure 2.20. "Differential Diagnosis" Basic Control Plan.

set of alternative interpretations and the regions (e.g. time) in the signal where they compete. The KS's output is a list of explanations for each alternative, each specifying 1) the region in the signal data to be reprocessed 2) the verification goals that must be met for confirming each alternative, and 3) a set of distortions hypothesized as the reason for why the verification goals hadn't been achieved earlier. These verification goals are of the same variety as those supplied by the discrepancy diagnosis KS.

As with the discrepancy diagnosis KS, the differential diagnosis KS applies knowledge from SPA processing models to predict how the front-end SPAs' parameter values could have made correlates for different features of alternative objects appear similar. Based on these predictions, the reprocessing component can then propose a reprocessing strategy to disambiguate the features' correlates.

2.6 Summary

This chapter has presented the domain-independent specification for IPUS in terms of the basic RESUN control plans used to achieve the general control strategy for the architecture. It also described the specifications for the reprocessing-loop components and their interactions with the data structures and code for capturing the formal signal processing knowledge of a perceptual domain. This section concludes the chapter with a summary of the information that must be supplied to instantiate the architecture.

1. Blackboard Signal Representations

2. SPA Definitions

    (a) SPA code

    (b) control parameters

    (c) distortion-reduction rules

    (d) violation and fault discrepancy tests

    (e) supercontext tests

    (f) mapping functions

3. SIA Definitions

    (a) conflict discrepancy tests

4. domain-dependent control plans and focusing heuristics

    (a) focusing heuristic for selecting PSM SOUs for resolution

    (b) plans and heuristics for resolving hypothesis SOUs

    (c) heuristics for determining the highest abstraction level produced by the front end before top-down processing commences

    (d) plan for executing front end

    (e) plan for determining adjustments to front-end SPAs

    (f) interpretation sufficiency test for Simplify-Interpretation plan

5. Discrepancy Diagnosis and Differential Diagnosis Components

    (a) discrepancy diagnosis KS

    (b) differential diagnosis KS

    (c) distortion processes

6. Reprocessing Component

   (a) iterative parameter adaptation rules

   (b) control-plans encoding the SPA sequences of reprocessing strategies

CHAPTER 3

RELATED RESEARCH

This chapter discusses related work in three areas connected with IPUS. The first section discusses general approaches to the problem of integrating signal processing and environmental interpretation in perceptual systems that do not rely on IPUS' dual search approach. The second section clarifies the relationship between the adaptive behavior provided by control-theoretic approachs and that provided by IPUS. The third section describes related work in the area of auditory scene analysis, the application domain that is used to validate the IPUS framework in this thesis.

3.1 Architectural Work

The IPUS architecture represents the formalization and extension of concepts explored in earlier work on a diagnosis system that exploited formal signal processing theory to debug signal processing systems [Nawab *et al.*, 1987] and in work on meta-level control [Hudlická and Lesser, 1987, Hudlická and Lesser, 1984] that used a process of fault-detection, diagnosis, and replanning to decide the most appropriate parameters for controlling a problem-solving system.

Several recent systems have been developed that try to integrate interpretation activity and numeric-level signal processing without recourse to IPUS' planner-based dual search approach. These systems will be examined as representatives of general architectural approaches to the problem of controlling the interaction of signal processing and environmental interpretation in perceptual systems. For clarity the approaches are described in terms of the IPUS components they functionally include.

The perceptual framework of Hayes-Roth's GUARDIAN system [Hayes-Roth *et al.*, 1989] is typical of systems whose input data points already represent useful information and require

little formal front-end processing other than to control the rate of information flow. The system was designed to monitor the basic vital signs of patients in an intensive care unit. The system incorporates an input-data management component that controls the sampling rate of signals in response to workload constraints. Information flow is controlled through variable sample-value thresholds and variable sampling rates. This control framework is somewhat limited since it is based only on the system's time requirements for reasoning about classes of signals, and provides good performance primarily because the signals monitored are relatively simple and noise-free in nature: heart-rate, temperature fluctuations, etc. The framework's lack of centralized components for any of the four IPUS reprocessing-loop tasks leads to inadequate generality for the wide range of signal/environment interactions which can include signals containing complex structures that must be modeled over time in the presence of variable noise levels. No implication should be inferred from this discussion that frameworks in this class do not perform any diagnostic reasoning. Rather, the observation should be made that this reasoning capability is not applied to the identification of potentially adverse interactions between the environmental signal and the front-end processing.

The framework used by Dawant to build a system for interpreting EKG traces [Dawant 1991] is closer in spirit to IPUS. It is typical of systems designed with the intent of providing alternative evidence sources as "backup" evidence when moderate deviations are observed between signal behavior and partially-matched signal event models. The framework does not support the selective reprocessing or selective application of specialized SPAs since data is always gathered from every front-end SPA whether required for interpretation improvement or not. This reliance on a fixed set of SPAs that are all always executed leads to systems where more and more SPAs are added to front ends[1] as the environmental complexity increases, ending in a combinatorial explosion in the number of SPAs necessary to unambiguously identify all signals in an environment. Unlike IPUS, most architectures in this category operate on the implicit assumption that the signal-generating environment will not interact adversely with

[1] as opposed to a library of SPAs to be used only when needed.

the signal processing algorithms' limitations to produce output distortions that might not have occurred if more appropriate processing algorithms had been used. Any deviations between observed signal behavior and available signal event models are attributed to chance variations in the *source* being monitored, never to the signal's *interaction* with inappropriate SPAs or with other sources in the environment.

De Mori et al. [De Mori *et al.*, 1987] developed a formal interaction framework in a system to recognize spoken letters of the English alphabet. This framework is representative of architectures with strong reliance on differential diagnosis techniques. These architectures are often employed in domains where there is little or no dependence between consecutive signal events. Interpretations in the system were generated by learned rules expressing letter identifications in terms of a signal-event grammar. Often more than one letter could be indicated by a single rule (in their terminology the rule has a *confusion set*). When such rules are activated, the system pursues a differential diagnosis strategy relying on rules describing SPAs that are suited to disambiguating confusion sets with given members. Thus, the system makes use of selective SPA application and differential diagnosis strategies. However, given the framework's relatively restricted application domain, there is a serious question of whether the approach can be scaled up without including the ability to model the environment's signal processing theory. Since the environment of the system considers its objects (letters) as isolated, unrelated entities, the framework does not incorporate any use of diagnosis in conjunction with environmental constraints (e.g., A 'C' has been identified at time t_{-1} and a 'B' is expected at time t_0 since there is an environmental constraint that 'B's follow 'C's. When no evidence supporting the expectation is observed, diagnostic reasoning should be attempted to explain why).

GOLDIE [Kohl *et al.*, 1987] is an image segmentation system that uses high-level interpretation goals to guide the choice of numeric-level segmentation algorithms, their sensitivity settings, and region of application within an image. The system's architecture represents the set of architectures that place strong emphasis on selective SPA application without explicit

guidance from formal signal-processing theory. The system uses a "hypothesize-and-test" strategy to search for algorithms that will satisfy high-level goals, given the current image data. While it incorporates an explicit representation of algorithm capabilities to aid in this search, and an explicit representation of reasons for why it assumes an algorithm is appropriate or inappropriate to a particular region, the system notably does not incorporate any diagnosis component for analyzing unexpected "low quality" segmentations. If an algorithm were applied to a region and the resulting segmentation were of unexpectedly low quality, the framework would not parallel IPUS and attempt to diagnose the discrepancy and exploit this information to reformulate the algorithm search but would select the next highest rated algorithm from the original search.

In the same category as GOLDIE is TraX [Bobick and Bolles, 1992], a system for interpreting image frame sequences. Although its design was driven by the goal of supporting multiple, concurrent object descriptions, the system incorporates some concepts similar to those in formulation of the IPUS architecture. The system supports detection of deviations from expected measurements and determination of the possibility that these deviations might have resulted from processing techniques inappropriate to the current context. In a manner similar to conflict discrepancy detection in IPUS, TraX compares higher-level expectations from previous frames against its segmentation SPAs' outputs for the current frame. In contrast to the IPUS architecture specification, however, TraX does not use models derived from an underlying theory for its SPAs to inform the discrepancy detection and diagnosis processes. It relies instead on empirically derived statistical performance models for the segmentation algorithms. While TraX allows for the use of different SPAs for different contexts, it does not support the adaptation of SPAs' control parameters for different contexts.

Bell and Pau [Bell and Pau, 1990a, Bell and Pau, 1990b] formalize the search for processing parameter values in numeric-level image understanding algorithms in terms of the Prolog language's unification and backtracking mechanisms. They express SPAs as predicates defined on tuples of the form $(M, p_1, \ldots, p_n)$, where M represents an image pattern and the p's

represent SPA control parameters. These predicates are true for all tuples where M can be found in the SPA output when its control values are set to the tuple's p values. Prolog's unification mechanism enables these predicates to be used in both goal-directed and data-driven modes. In a goal-driven mode, M is specified and some of the parameters are left unbound. The unification mechanism verifies the predicate by iteratively binding the unspecified parameters to values from a permissible value set, applying the SPA, then checking if the pattern is found. In a data-driven mode, M is not bound and the parameter values are set to those of the front-end processing. M is then bound to the SPA results. The method relies on Prolog's backtracking *cuts* [Giannesini *et al.*, 1986] to limit parameter-value search. A cut is a point in the verification search space beyond which Prolog cannot backtrack. This reliance on a language primitive makes it difficult to explicitly represent (and therefore to reason about) heuristic expert knowledge for constraining parameter-value search as can be done in IPUS's reprocessing component. The cut mechanism also does not permit the use of formal diagnostic reasoning to further constrain parameter-value search based on the cause of an SPA predicate failure.

3.2 IPUS and Control Theory

Research in active vision and robotics has recognized the importance of tracking-oriented front-end SPA reconfiguration [Swain and Stricker, 1991], and tends to use a control-theoretic approach for making reconfiguration decisions. It is indeed sometimes possible to reduce the reconfiguration of small sets of front-end SPAs to problems in linear control theory. In general, however, the problem of deciding when an SPA (e.g., a specialized shape-from-X algorithm or an acoustic filter) with particular parameter settings is appropriate to a given environment may involve nonlinear control or be unsolvable with current control theory techniques.

It is important to clarify the relationship between the IPUS approach and the classic control theoretic approach [Seborg 1986]. Control theory uses stochastic-process concepts to characterize signals, and these characterizations are limited to probabilistic moments, usually no higher than second-order. Discrepancies between these stochastic characterizations and an

SPA's output data are used to adapt future signal processing. In contrast, the IPUS architecture uses high-level symbolic descriptions (i.e., interpretation models of individual sources) as well as numeric relationships between the outputs of several different SPAs to characterize signal data. Discrepancies between these characterizations and SPAs' output data are used to adjust future signal processing. Classic adaptive control should therefore be viewed as a special case of an IPUS architecture, where the interpretation models are described solely in terms of probabilistic measures and low-level descriptions of signal parameters.

3.3 Auditory Scene Analysis Work

The segregation and identification of simultaneous sounds in an acoustic stream has been an important problem for psychoacousticians and psychologists for over a century [Helmholtz 1885]. Bregman's comprehensive book, *Auditory Scene Analysis* [Bregman 1990], provides a thorough treatment of the problem's history and many signal cues psychoacousticians have identified as useful to humans in perceptually separating sounds. In the field of machine perception, however, investigation of the problem of computational auditory scene analysis (CASA) has only recently started in earnest, with most work focused on feature extraction. In addition to the nascent CASA field, the specialized fields of speech recognition and music transcription and interpretation have bearing on the problem of acoustic stream segregation.

Except for simultaneous multiple-speaker speech segregation [Weintraub 1986, Parsons 1976], most speech recognition work abstracted the problem of auditory scene analysis away by assuming that the acoustic signal either contained only speech or had a high signal-to-noise ratio of speech to non-speech sounds. Recently, work in speech recognition in noisy environments [Brown and Cooke, 1992, Cooke and Brown, 1994] has begun to eliminate this assumption with some success. However, most efforts in this research direction tend to focus more on resynthesis of a "clean" speech signal from a noisy signal than on automatic recognition of all sounds in the signal. Thus, most of the segregation approaches adopted in this research divide acoustic signals into "speech" and "non-speech" categories, with little or no further specialization on "non-speech." A notable exception to this trend can be found in

Nawab et al.'s work [Nawab *et al.*, 1995], which developed a detailed taxonomy decomposing non-speech sounds into acoustic primitives on the same scale as phonemes. The research found that knowledge of the particular kinds of sounds that co-ocurred with speech aided in the selection of SPAs for separating the speech and non-speech signals.

The field of automatic music transcription and interpretation has explored approaches that come closer to being general methods for the CASA problem [Brown and Cooke, 1994, Kashino and Tanaka, 1993, Mellinger, 1991], including interleaved bottom-up signal processing and top-down interpretation. Most work in this area, however, concentrates on designing and applying fixed front ends that only incorporate processing strategies (e.g. SPA sequences) discerned from the human auditory system. In effect, the "architecture" studied in this research is the human auditory system. Such study has been and should continue to be useful in finding inspiration for the design of new individual SPAs though, as noted at the end of Section 1.2, it places limits on the range of capabilities considered for perceptual architectures.

Within the CASA field itself, most research is devoted to evaluating various acoustic signal features' utility in identifying certain signals and designing extraction techniques for the useful features. However, some research on general systems has begun. Much of the work has been done within the blackboard paradigm [Nii, 1986], under the assumption that auditory stream segregation can benefit from some kind of interleaved symbolic interpretation and adaptive numeric signal-processing [Cooke *et al.*, 1993, Ellis, 1996, Lesser *et al.*, 1995, Lesser *et al.*, 1993]. The work reported by Ellis is of particular interest in that it builds upon earlier reported IPUS work on SOU-usage, conflict-detection and reprocessing and incorporates these mechanisms in a framework for modelling the human auditory system's sound-segregation behaviors. That is, the framework uses predictions from models of sound-producing objects to constrain application of front-end SPAs in directed search for secondary evidence when direct evidence for an expected sound is not found.

The residue-driven architecture of Nakatani *et al.* [Nakatani *et al.*, 1995] is a non blackboard-based CASA framework that relies on the agent metaphor. In this research specialized agents are created to actively track spectral energy features such as frequency tracks

in a harmonic set, or wide-band background energy. The approach is *residue-driven* because new agents are created only when the current block of data has residual unaccounted-for energy (e.g. the possibility exists that a new sound has appeared in the environment). The approach has produced good interpretation results in tests on two- and three-source scenarios, but may be prone to a combinatorial explosion of increasingly-specialized agents as more complicated scenarios with a variety of feature-interactions are considered.

3.4 Research Summary

Taken individually, each reprocessing-loop component and control behavior of the IPUS architecture has been partially explored within at least one of the fields cited as relevant to this thesis. What qualifies the work in this thesis as a contribution to all the fields is that it represents the first time that an interpretation architecture using all the components and organizing control around the signal processing theory of a system's SPAs has been proposed and evaluated in a complex environment.

CHAPTER 4

THE IPUS SOUND UNDERSTANDING TESTBED

This chapter provides a description of the domain knowledge used to instantiate an IPUS Sound Understanding Testbed (SUT) that was developed to evaluate the IPUS architecture in a real-world application. Beyond providing a description of one system for interpreting acoustic signals from a complex environment having the potential for several interpretation problems, it serves as the logical conclusion to the abstract IPUS specification in Chapter 2. In studying the descriptions of the particular SPAs included in this instantiated IPUS system and in evaluating the testbed performance reported in Chapter 5 the reader should note that the SPAs themselves are not being directly evaluated in this work. While they are certainly reasonable SPAs to include in an acoustic interpretation system, this thesis does not mean to imply that collectively they are exhaustive or optimal. The descriptions of the SPAs are provided to facilitate understanding of how the testbed handles interactions between front-end signal processing and symbolic interpretation components.

The chapter has two sections. The first section summarizes the acoustic domain knowledge in the SUT. This includes the blackboard abstraction levels on which hypotheses about acoustic signal structures can be posted, the SPAs available to the testbed's front end, and the sound library used for generating the scenarios used to evaluate the system. The second section describes the information used in instantiating the SUT's control framework (control plans, SOUs, and focusing heuristics) and reprocessing loop (discrepancy-detection tests, context-

Table 4.1. SUT Sound Library

alarm clock1	chicken	foghorn	razor (electric)
alarm clock2	chime	footsteps	smoke alarm1
bell1 toll	clap	glass clink	smoke alarm2
bell2 toll	clock chime	gong	telephone dial
bicycle bell	clock ticks	hairdryer	telephone ring
bugle note1	cuckoo clock	knock	telephone tone
bugle note2	door chime	ovenbuzzer	truck motor
burglar alarm	door creak	owl	vending machine hum
car engine	firengine bell	pistolshot	viola
car horn	firehouse alarm	policecar siren	wind

manipulation functions, discrepancy and differential diagnosis components, and reprocessing strategies).

4.1 SUT Acoustic Knowledge

4.1.1 Sound Library

For the evaluation experiments presented in Chapter 5 the SUT was supplied with an acoustic library containing 40 sound-source models. As was briefly discussed in Chapter 1 the particular sources chosen for the library were selected to simulate a moderately complex acoustic environment containing a variety of real-world acoustic behaviors that would lead to interesting processing interactions among sounds in a random scenario. The range of acoustic behavioral categories in the library is defined in Table 4.2, while Table 4.1 lists all sounds in the library and their behavioral categories. Note that the categories are not mutually exclusive, since sounds can exhibit more than one behavior over time. A comprehensive catalog of the sound-source models in the library can be found in Appendix A. To give an indication of the potential for interactions among sounds randomly selected from the library and placed in scenarios with random start times, Figure 4.1 shows a histogram of the number of overlapping narrowband (e.g. $\leq$ 100-Hz wide) sound tracks in the library. Note that the higher the number of overlapping tracks there are in a spectral region, the greater the processing work that must be done to decide (1) whether in fact overlapping tracks are present in a scenario, and (2) which subset of the tracks that could be in the region of overlap are actually present.

Table 4.2. Sound Library Category Definitions.

CATEGORY	chirp
DEFINITION	the sound has relatively smooth time-dependent frequency shifts
SOUNDS	door creak, hairdryer, owl, wind

CATEGORY	harmonic
DEFINITION	the sound has a set of frequencies $f_1, \ldots, f_n$ that are integer multiples of some fundamental frequency f_0. Some multiples can have zero energy.
SOUNDS	alarm clock1, alarm clock2, bell1 toll, bell2 toll, bicycle bell, bugle note1, bugle note2, burglar alarm, car engine, car horn, chicken, chime, clock chime, cuckoo clock, doorbell chime, firengine bell, firehouse alarm, foghorn, gong, gong, hairdryer, owl, ovenbuzzer, policecar siren, razor, smoke alarm1, smoke alarm2, telephone dial, telephone ring, telephone tone, triangle, vending machine hum, viola

CATEGORY	impulsive
DEFINITION	the sound's acoustic energy is concentrated in a short time period. Such sounds tend to have significant energy throughout the spectrum.
SOUNDS	clap, clock ticks, footsteps, glass clink, knock, telephone dial, pistolshot

CATEGORY	repetitive
DEFINITION	the sound is repeated, though not necessarily with a regular period.
SOUNDS	clock chimes, clock ticks, cuckoo clock, footsteps, owl, policecar siren, telephone ring, telephone tone, telephone dial

CATEGORY	transients
DEFINITION	the sound's attack (signal onset) or decay (signal turn-off) behaviors differ from those found during the steady-state.
SOUNDS	bell1 toll, bell2 toll, chime, clap, clock chimes, door creak, gong, hairdryer, knock, razor, triangle

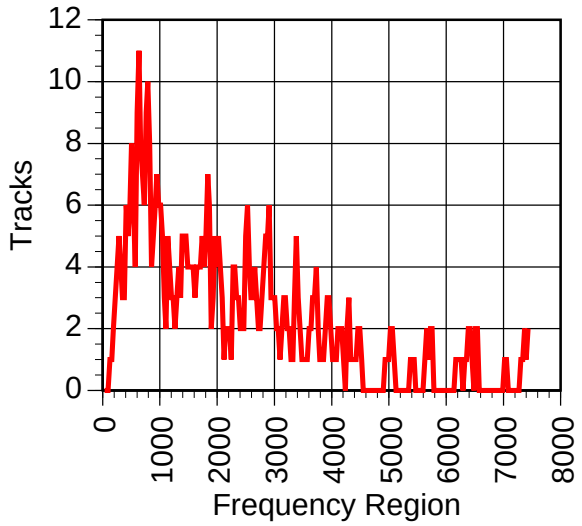


Figure 4.1. SUT Library Spectral Histogram

4.1.2 Acoustic Structure Knowledge

The testbed uses thirteen partially-ordered evidence representations to construct an interpretation of incoming signals. They are implemented through thirteen levels on the testbed's hypothesis blackboard. Figure 4.2 illustrates the support relationships among the representations, while the following discussion highlights the representations' content:

WAVEFORM: the raw waveform data. This representation is maintained since the testbed architecture will sometimes need to reprocess data. To conserve storage, only the last 3 seconds of waveform data are kept on the testbed's blackboard.

ENVELOPE: the envelope, or shape, of the time-domain signal. This level also maintain statistics such as zero-crossing density and average energy for each block of signal data.

SPECTROGRAM: spectral hypotheses derived for each waveform segment with algorithms such as the Short-Time Fourier Transform (STFT) and the Quantized Short-Time Fourier Transform [Nawab and Dorken, 1995] (QSTFT), an approximate SPA.

PEAK: peak spectral energy regions in each time-slice in a spectrogram. These indicate narrow-band features in a signal's spectral representation.

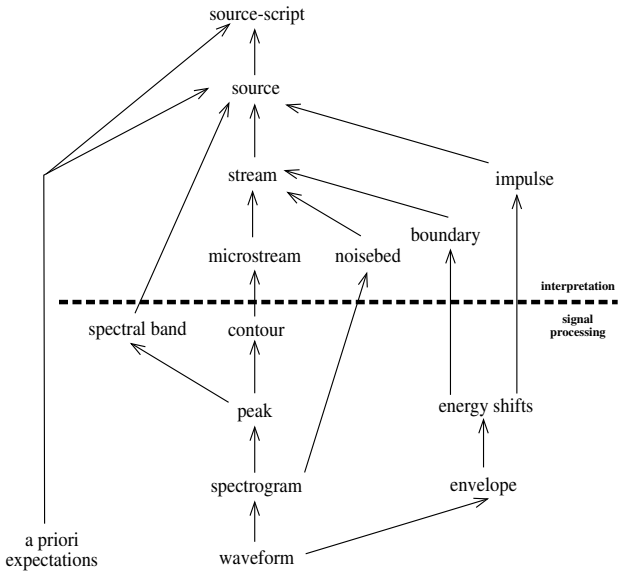


Figure 4.2. SUT Acoustic Abstraction Hierarchy.

SHIFT: sudden energy changes in the time-domain envelope.

EVENT: time-domain events, which group shifts into boundaries (i.e. a step-up or step-down in time domain energy indicating the possible start or end of some sound) and impulses (i.e. sudden spikes in the signal).

CONTOUR: groups of peaks whose time indices, frequencies, and amplitudes represent a contour in the time-frequency-energy space with uniform frequency and energy behavior.

SPECTRAL BAND: regions of spectral activity (i.e. clusters of peaks) in a spectrogram. this abstraction level implements a knowledge approximation technique that avoids over-reliance on strict narrowband descriptions of sounds by mapping rough clusters of spectral activity in a spectrogram to those sounds in the sound library that overlap those frequency regions.

MICROSTREAM: a track. A microstream represents a sequence of contours that has an energy pattern consisting of an attack region (signal onset), a steady region, and a decay (signal fadeout) region. The attack and decay regions can have frequency chirps in

addition to their energy change. The steady region of "long-term" tracks have stochastic entropy-like measures of the frequency and energy variability among the peaks in the track.

NOISEBED: the wideband component of impulsive areas within a sound source's acoustic signature. Microstreams (tracks) often form "ridges" on top of noisebed "plateaux," but not every noisebed has well-defined microstreams associated with it.

STREAM: a set of microstreams and noisebeds grouped as a perceptual unit on the basis of streaming criteria developed in the psychoacoustic community [Bregman 1990]. Specifically, the SUT's KSs group microstreams together when they have similar fates (e.g. synchronized onset- and end-times, synchronized chirp behavior), or when they share a harmonic relationship. Noisebeds are predicted and searched for only after a stream has been identified as a particular source's signature. Expectation-derived stream hypotheses also record the expected energy ratios among the steady-state regions of component microstreams.

SOURCE: stream hypotheses, with their durations supported by boundaries, are interpreted as sound-source hypotheses according to how closely they match source-models available in the testbed library. Partial matches (e.g. a stream missing a microstream, or a stream with duration shorter than expected for a particular source) are accepted, but hypotheses resulting from these matches will later cause the testbed to attempt to account for the missing or ill-formed evidence (e.g. microstreams or noisebeds) as artifacts of inappropriate front-end processing.

SCRIPT: temporal streaming of a sequence of sources into a single unit (e.g. a periodic source such as footsteps being composed of a sequence of footfalls, or the combination of cuckoo-chirps and bell-tones in a cuckoo-clock chime).

The following discussion elaborates the information about microstream entropies[1] and noisebed features that is maintained in the acoustic database's sound-source models.

4.1.2.1 Microstream Entropy Values

Microstream entropies are intended to express the variability in frequency and amplitude exhibited by the steady-state peaks in a sound's spectrogram. They are calculated only for sounds whose microstreams' steady-state behavior lasts longer than 0.75 seconds, and only for peaks generated from a 2048-point short-time Fourier Transform with a 256-point decimation and a 1024-point rectangular analysis window. Whenever the SUT needs to verify entropy values for a track, it must use peaks from that STFT parameter context. Since both frequency and amplitude entropies are calculated in the same manner, only the amplitude entropy calculation will be described in detail.

The first step in calculating a microstream's amplitude entropy is to calculate the *segment amplitude entropy*,

$$\nu_a^k = \frac{\sigma_k^a}{\mu_k^a}$$

for each segment in the steady region of the microstream, where μ_a^k and σ_a^k are the mean and standard deviation, respectively, of the amplitudes of the M peaks in the k'th segment in the track. In the library a segment has $M = 30$ peaks, which covers 0.5 seconds in a 16KHz-sampled signal's spectrogram. Note that segments overlap; the k'th segment includes the k'th through $k + M - 1$'th consecutive peaks. By considering tracks of length at least 0.75-seconds, this gives us at least 15 segments per track. A microstream's *amplitude entropy* is defined by the values μ and σ, which are the mean and standard deviation of the segment entropies, respectively.

[1] Although the term "entropy" is used most often in information theory to measure the logarithmic information content of a signal, in this thesis it is borrowed to refer to the stability, or organization, within a track. High entropy values will indicate unstable, highly-variable tracks, while entropy values near zero will indicate stable tracks. Statistics textbooks often refer to this measure as the "coefficient of relative variation."

4.1.2.2 Noisebed Models

The noisebed model is generated from an impulsive sound's spectrogram as follows. The spectrogram is produced from a 128-point short-time Fourier Transform with 32-point decimation and a 64-point rectangular analysis window. A 0.2-second region in the spectrogram is divided into an 8×10 (frequency $\times$ time) grid, starting at a point just before the sound reaches maximum energy. The grid's energy values are normalized with the maximum grid tile set to 1.0. From this grid four noisebed features are calculated: (1) the spectral center of gravity at the time the impulse is at its maximum energy, (2) the difference between the energy of the maximum-energy grid tile and that of the tile one frequency-slice *above* it at the same time, (3) the difference between the energy of the maximum-energy grid tile and that of the tile one frequency-slice *below* it at the same time, and (4) the least-squares exponential decay rate fitted to the energy values in the tiles in the same frequency-slice as the maximum-energy tile, but at time-slices after and including the maximum tile.

4.2 Architecture Instantiation

This section summarizes the information used to meet the instantiation checklist provided at the end of Chapter 2. The first part presents highlevel accounts of the SPAs and SIAs used in the testbed. The second part completes the representation information from the previous section with a description of how the SUT summarizes belief in types of signal hypotheses. The third part discusses the domain-dependent control plans and focusing heuristics used in the SUT, and the fourth, fifth and sixth parts discuss the instantiation of the reprocessing loop's diagnostic components and the reprocessing component, respectively.

4.2.1 SPAs and SIAs

4.2.1.1 SPAs and Support Information

The SUT has 9 KSs implementing SPAs that can be used in front-end processing, and 7 KSs implementing SIAs that can be used to generate high-level interpretations, in addition to 3 KSs for discrepancy-detection, discrepancy diagnosis, and differential diagnosis. SPA KSs can be

distinguished from SIA KSs by their task. SPA KSs implement mathematical transformations involving no search on the input data, whereas SIA KSs implement search processes for evidence to support some hypothesis. In lieu of code, Tables 4.3 and 4.4 summarize all the testbed SPAs and their their control parameters.

While examining the entry for the STFT SPA, readers familiar with signal processing applications will note that in many applications the window of signal data for each Discrete Fourier Transform is first "tapered" toward the ends by multiplication with another window of scale values. The shape of the scale-value window can be shown (see [Oppenheim and Schafer, 1989], Chapter 7) to have an effect on both the frequency resolution of the resulting spectrogram and the introduction of spurious peak-values of low-to-moderate energy appearing on either side of true spectrogram peaks.[2] In the SUT a uniform, rectangular, scaling window is used, which provides the sharpest resolution at the expense of increased spurious peak values. The two peak-picking SPAs (THRESH-PEAK and MAX-PEAK) in the SUT were designed to ignore these spurious peaks.

In the SUT only the SHIFT and THRESH-PEAK SPAs perform fault and violation discrepancy tests on their results. Recall from Section 2.5.1 that fault discrepancy tests check to make sure that output from two front-end SPAs are consistent, while violation discrepancy tests check to make sure that output from an SPA does not exceed certain gross environment constraints and that the output is not subject to boundary effects of applying an SPA. "Boundary effects" refers to the results of applying an SPA to data that do not meet some assumption. In the SUT the SHIFT SPA performs a violation test to make sure that all shift hypotheses occur away from the trailing edge of the current envelope, since the envelope edge does not provide a sufficient neighborhood for determining if a shift represents a boundary (i.e. long-term sound's onset) with a sustained signal energy, or an impulse with a momentary burst of signal energy. When a violation is detected, the SHIFT SPA attaches a *Constraint-Inconsistency* to the

[2] These are referred to as peaks caused by the sidelobes of the scale window's Fourier Transform. See Chapter 7 of [Oppenheim and Schafer, 1989] for more details.

Table 4.3. Summary of SUT Front-End SPAs, Part 1.

SPA	TRANSFORM	PARAMETERS	DESCRIPTION
Short-Time Fourier Transform (STFT) *see [Nawab and Quatieri, 1988]*	Spectrogram ⇑ Waveform	Window, Decimation, FFT-Size	Produces a spectrogram from `FFT-Size`-point FFTs on `Window`-point data blocks whose start points are separated by `Decimation` data points.
Limited Short-Time Fourier Transform (LSTFT)	Spectrogram ⇑ Waveform	Freq-Point, Window, Decimation, DFT-Size	Produces a spectrogram evaluated only at the `Freq-Point`'th frequency in `DFT-Size`-point Fourier Transforms on `Window`-point signal blocks with starts `Decimation` data points apart.
Quantized Short-Time Fourier Transform (QSTFT) *see [Nawab and Dorken, 1995]*	Spectrogram ⇑ Waveform	QWindow, QFFT-Size, QDecimation, Freq-Radius	Quantizes the signal to (-1,0,1) and computes only frequencies within `Freq-Radius` of estimated max-energy frequency. SPA is an order of magnitude faster than the STFT.
Adaptive Time-Frequency (ATF) *see [Jones and Parks, 1990]*	Spectrogram ⇑ Waveform	Window-List, FFT-Size, Decimation	Similar to STFT, except spectrogram maximizes the energy concentration of locally dominant tracks over those produced by STFT analysis with windows in `Window-List`.
Max Peak-Picker (MAX-PEAK)	Peak ⇑ Spectrogram	Cutoff, Freq-Region	Classifies the max energy value within `Freq-Region` in each spectrum of a spectrogram as a peak, so long as it is $\geq$ `Cutoff`% of max energy value over all spectra.

Table 4.4. Summary of SUT Front-End SPAs, Part 2.

SPA	TRANSFORM	PARAMETERS	DESCRIPTION
Thresholded Peak-Picker (THRESH-PEAK)	Peak ⇑ Spectrogram	Threshold, Limit, Neighborhood	Classifies as peaks the (at most) `Limit` energy values in a spectrum that are greater than `Threshold` and are the local maximum over the adjacent `Neighborhood` values on either side.
Banding (BAND)	Spectral Band ⇑ Peak	Threshold, Max-Dead-Time	Clusters regions of at least `Threshold` peaks into bands. Regions closer than `Max-Dead-Time` are merged.
Envelope-Tracker (ENV)	Envelope ⇑ Waveform	Order	Envelope is generated from signal points that are local maxima within a window equal to the avg. zero-crossing separation. This process is done recursively an `Order` number of times.
Shift-Detection (SHIFT)	Shift ⇑ Envelope	Rel-Diff	Identifies places in the envelope where the amplitude changes by more than `Rel-Diff`% from one envelope point to the next.
Contouring (CONTOUR)	Contour ⇑ Peak	Freq-Radius, Energy-Radius, Shift-Threshold	Groups peaks into time sequences. Each peak is <= `Freq-Radius` and `Energy-Radius` from its immediate neighbors. Contours' energy shifts cannot have internal angles less than `Shift-Threshold` degrees.

potentially-unreliable shift hypothesis, and stores an *Uncertain-Hypothesis* SOU in the PSM corresponding to the shift.

The THRESH-PEAK SPA performs a fault discrepancy test on the set of peak hypotheses produced to check that the total spectral energy contained within the peaks accounts for at least 30% of the time-domain energy in the waveform from where the peaks' spectra were generated. When a fault is detected, the SPA places a *Constraint-Inconsistency* SOU on each peak hypothesis, and stores an *Uncertain-Hypothesis* SOU in the PSM corresponding to the peaks.

Supercontext tests check to see whether there are already results on the signal-representation blackboard which contain the information to be produced by a soon-to-be-invoked SPA. If such

results already exist, and the information can be more cheaply extracted from existing results than computed from scratch, context-mapping functions are called to extract the information and return the (possibly estimated) SPA results. Supercontext tests and mapping functions are defined for the STFT SPA, the LSTFT SPA, and both peak-picking SPAs. The LSTFT supercontext test checks if any LSTFT or STFT spectrogram has already been produced which covers the time-frequency region to be analyzed and which has a `Decimation` factor equal to or higher than the desired spectrogram, or a frequency-sampling rate (`FFT-Size`) greater than or equal to that of the parameter-context to be executed, coupled with an analysis window `Window` equal to that of the parameter-context to be executed. The STFT's SPA uses the same test. For both SPAs, their mapping functions take existing results from supercontexts and subsample the existing spectrograms in time or to approximate the desired spectrograms.

The THRESH-PEAK SPA's supercontext test checks for results from earlier THRESH-PEAK parameter contexts that used `Threshold`s equal to or lower than that of the desired parameter context. The MAX-PEAK SPA's supercontext test checks for peaks in the new `Freq-Range` from earlier contexts that had `Freq-Range`s wider than that of the desired context. The context-mapping function for both SPAs is the identity function: peaks meeting the criteria are simply returned as if they'd been produced by the desired parameter context.

Distortion-reduction rules for the SPAs define how individual control parameters should be modified to eliminate or reduce various classes of distortions that could be manifested in an SPA's correlates. For the SUT's SPAs these are incorporated into the reprocessing plans discussed in Section 4.2.6.

4.2.1.2 SIAs and Support Information

Each SIA KS is used to make inferences about hypotheses on adjoining levels above the interpretation line in Figure 4.2. They are referred to by the two levels for which they create inferences: BAND-AND-SOURCE, CONTOUR-AND-MICROSTREAM, etc. The 7 SIA KSs can be used in two modes: (1) explain a set of lower-level hypotheses as a higher-level hypothesis, or (2) generate expectation hypotheses about how lower-level interpretations that

could support a higher-level hypothesis should appear. In IPUS it is the role of SIAs to execute conflict discrepancy tests to check for unsuccessful matches between SPA correlates and model-based expectations being confirmed. Since the control strategy used by the SUT versions in this thesis depends only on the conflict discrepancy tests of the CONTOUR-AND-MICROSTREAM SIA, this section describes only those tests.

When the CONTOUR-AND-MICROSTREAM SIA (or, for that matter, any SIA in an IPUS system) detects no supporting contour evidence at all for a microstream expectation, it places a *Support-Exclusion* SOU on the expectation hypothesis, annotated with the processing context under which support had been sought. The SIA uses a heuristic test of the amount of energy in the spectrogram and time-domain signal region being examined for contours to estimate whether no tracks exist in the region or whether there are peaks in the region that for some front-end-related reason were not combined into contours. If the test shows that (1) more than half of the expected microstream region has unexplained peak energy, or that (2) envelope energy has suddenly changed, or that (3) higher-energy microstreams related to the expectation microstream have missing evidence in the same region, the SIA annotates the empty inference link on the expectation microstream with a *Possible-Inappropriate-Parameter* SOU. This SOU will reduce the negative evidence weight caused by the missing evidence (preventing the sound hypothesis of the microstream from being disbelieved) until the discrepancy diagnosis KS can be applied to the *Support-Exclusion* SOU.

If support contours meeting all expectations (i.e. contours are all within the time, energy, frequency and entropy bounds of the microstream) the SIA links the support and expectation with an inference structure and places an *Uncertain-Support* SOU on the expectation hypothesis. If the contours overlap but do not precisely match the expectation microstream's constraints, a *Constraint* SOU is placed on the inference link from the contours to the microstream. As will be seen in Section 4.2.2 these SOUs will reduce the positive evidence weight provided by the contours and increase the negative evidence weight against the microstream.

4.2.2 Hypothesis Beliefs and Summarization

In RESUN (and the SUT) the decisions to consider a some hypothesis as an answer or to consider some hypothesis such as a microstream as credible ultimately depend on the levels of uncertainty in the hypothesis. This section presents the process of generating beliefs for blackboard interpretation-level hypotheses. The RESUN framework annotates each hypothesis with a *summary unit* that numerically summarizes several categories of uncertainty in the hypothesis. For the purposes of the SUT only three entries in the summary unit are considered here: support rating, negative evidence rating, and negative evidence explanation increase factor.

The support rating of a microstream numerically represents how complete is the contour or spectrogram support recorded in the *Uncertain-Support* SOUs on the hypothesis. This value is determined by adding the total weighted duration of found support and evaluating a sigmoid strength function to see how much strength (range 0.00 to 1.00) the amount of support provides. A set of found support contours is weighted by a fraction less than 1 only when they have a *Constraint* SOU. The negative evidence rating is determined similarly, with the total duration of all missing-support regions taking the place of the found support. Missing-support regions are weighted by fractions less than 1 when a *Possible-Inappropriate-Parameter* SOU is present for the regions. The negative evidence explanation increase factor shows how much the support rating would increase if all of the negative evidence for a hypothesis were replaced with reprocessed positive support once the negative evidence was explained away.

Noisebeds' support ratings are based on how close the feature vector from the observed spectrogram support approaches the mean vector of the noisebed model, using the same sigmoid-style function as in the microstream case to weight the distance. If the distance is very large, a support rating of zero is supplied, and a negative evidence rating of 1.0 is supplied.

The process of combining support information into a summary unit for the supported hypothesis is termed **summarization** in the RESUN framework. Summarization occurs after any KS creates an inference, in order to keep all hypotheses updated after a change to the

blackboard. In the SUT stream hypotheses are summarized as follows. A stream's support rating is the average of the support ratings of its component microstreams and noisebeds. A stream's negative evidence rating is the *maximum* of its components' negative evidence ratings, and its negative evidence explanation increase factor is generated by comparing the current support rating with the new average that would be obtained if the maximum negatively-rated component's evidence was found.

If a source hypothesis has stream support, it is summarized with an exact copy of its support stream's summary unit. If it has only spectral-band support, it is recursively summarized as in the case of the stream, with spectral band sets taking the place of microstreams.

4.2.3 Domain-Dependent Focusing Heuristics and Plans

At the PSM level of control in the SUT, the focusing heuristic that selects SOUs from the PSM for the ?psm-sou variable of the `Resolve-PSM-Uncertainty` plan (Figure 2.6) uses the following preference hierarchy:

1. hypotheses for SPA outputs with *Constraint-Inconsistency* SOUs.
2. hypotheses for tonal sounds from earlier data blocks with expectations for the current block, ordered by overall summary belief rating.
3. hypotheses for tonal sounds with no support, ordered by the sound's projected start time.
4. hypotheses for sounds with explainable negative evidence, ordered by overall summary belief rating.
5. hypotheses for sounds that could be explained as part of a script, ordered by overall summary belief rating.
6. unexplained or partially explained spectral bands
7. hypotheses for partially supported impulsive sounds. That is, the sound's wideband noisebed has not yet been sought for.

8. hypotheses for scripts that specify new expected sounds for the current block
9. hypotheses for impulsive sounds that have no support.
10. hypotheses for sounds expected for scripts in the *next* block.
11. the no-evidence SOU.

Note that "negative evidence" can be indicated either by an *Uncertain-Support* SOU summarizing the uncertainty in a hypothesis' supporting hypotheses, or by a *Support-Exclusion* SOU indicating the absence of the expected support.

In lieu of an exhaustive list of figures listing all of the SUT-dependent control plans, the following control strategy summary is provided. The basic control strategy of the SUT for each data block achieved by the PSM-SOU selection heuristic and the control plans is to:

1. apply the front end, producing peaks and shift hypotheses as the highest-level blackboard results
2. produce spectral bands from the observed peaks
3. apply the reprocessing-loop to *Constraint-Inconsistency* SOUs
4. if any sound hypotheses from the previous block are expected to extend into the current block, perform steps 5–9, then return to step 4.
5. generate initial (possibly competing) source hypotheses based on the bands
6. recursively generate stream and microstream and noisebed expectations for the sources
7. for microstream expectations, generate and execute contouring SPAs with parameter contexts based on the frequency width and energy variations of the expected microstreams; for noisebeds, do the same with STFT SPAs
8. search for support contours, spectrograms, and impulses

9. if any missing evidence from the search in step 7 appears diagnosable, execute the diagnosis KS, and the reprocessing component, if a diagnosis is found

10. repeat steps 6–8 for each sound hypothesis with partial-support in the current data block.

11. create expectations for the next data block, load in the next data block, and return to step 1.

The application of the contouring SPA with specialized control parameters for each microstream is referred to as *focused contouring* to distinguish it from the application of the SPA with default parameters that would occur in the front end. In Chapter 5, when the number of parameter contexts produced during reprocessing is counted, focused contouring contexts are counted only if they are produced during the execution of a reprocessing plan.

In the SUT experiments in this thesis, the IPUS capability for adapting the front end based on model-based expectations is not explored, so no plan for this component of IPUS is reported.

The interpretation sufficiency heuristic used by the SUT in the generic `Simplify Interpretation` control plan requires that 70% of the energy in the spectral-bands be explained before the system can advance to the next data block, unless no sound hypotheses can be credibly generated from the regions covered by unexplained spectral bands.

When a higher-level interpretation hypothesis has more than one hypothesis SOU that could be solved, the following preference hierarchy is enforced by focusing heuristics for the *Have-Extension-SOU-Resolved* subgoal of the "Resolve Extension SOU" control plan (Figure 2.13):

- **script:** *Partial-Support* SOUs for as-yet unsought component sources in the current data block.

- **source:**

    1. *Partial-Support* SOUs, indicating that only some of the possibly available spectral-band or stream or impulse support for the source has been sought.

2. *No-Support* SOUs, when the source is a newly-generated script-based expectation.
3. *Support-Exclusion* SOUs, indicating an impulsive sound's expected impulse was not found in the current block.
4. *Uncertain-Support* SOUs, if they indicate that explainable negative uncertainty exists for a supporting stream or impulse hypothesis.

- **stream:**

1. *No-Support* SOUs, when a stream expectation for a source has been generated top-down.
2. *Partial-Support* SOUs, indicating that not all the possible duration of the stream's microstreams has had evidence sought for in the current block, or that not all possible boundaries have been sought for.
3. *Support-Exclusion* SOUs, if they indicate explainable missing microstream or noise-bed support.
4. *Uncertain-Support* SOUs, if they indicate that explainable negative evidence uncertainty exists for a supporting microstream or noisebed hypothesis.

- **impulse:** *No-Support* SOUs, when an impulse expectation for a source has been generated top-down.

- **microstream:**

1. *No-Support* SOUs, when a stream expectation for a source has been generated top-down.
2. *Partial-Support* SOUs, indicating that not all the possible duration of the stream's microstreams has had evidence sought for in the current block, or that not all possible boundaries have been sought for.
3. *Support-Exclusion* SOUs, if they indicate explainable missing contour evidence for a region of the microstream.

4. *Uncertain-Support* SOUs, if they indicate that constraint uncertainty (e.g. contour support is too long in time or is too wide in frequency) exists for a region of the microstream.

- **noisebed:** *No-Support* SOUs, when an impulse expectation for a stream has been generated top-down.

Note that an *Uncertain-Support* SOU indicates explainable negative evidence if the support hypothesis it represents itself has a supporting inference annotated with a *Constraint* SOU or a *Possible-Inappropriate-Parameter* SOU. A *Support-Exclusion* SOU indicates explainable negative evidence if the negative inference for the SOU's missing support is annotated with a *Possible-Inappropriate-Parameter* SOU. An *Uncertain-Support* SOU indicates explainable negative evidence if the support hypothesis it represents itself has a supporting inference annotated with a *Constraint* SOU or a *Possible-Inappropriate-Parameter* SOU. A *Support-Exclusion* SOU indicates explainable negative evidence if the negative inference for the SOU's missing support is annotated with a *Possible-Inappropriate-Parameter* SOU.

SOUs on a hypothesis that are of the same type but "located" at different times along the extent of the hypothesis are handled in order starting from the one closest to the end of the current data block and continuing backwards in time toward the earliest one.

4.2.4 Discrepancy Diagnosis KS

The SUT discrepancy diagnosis KS, which is based on the work of Nawab *et al.*[Nawab *et al.*, 1987], models the reasoning of a signal processing expert and carries out a discrepancies-to-distortions inverse mapping. This diagnostic reasoning is captured within a means-ends analysis framework [Newell and Simon, 1969] using multiple levels of abstraction and a verification phase. Furthermore, the reasoning is carried out with a qualitative description of the various signal quantities involved in order to deal with uncertain and approximate information. Figure 4.3 outlines the plan-and-verify strategy of the diagnostic process.

The formal discrepancy diagnosis task is to generate a sequence of "distortion operators" that can explain the discrepancies between an *initial state* that represents the information assumed

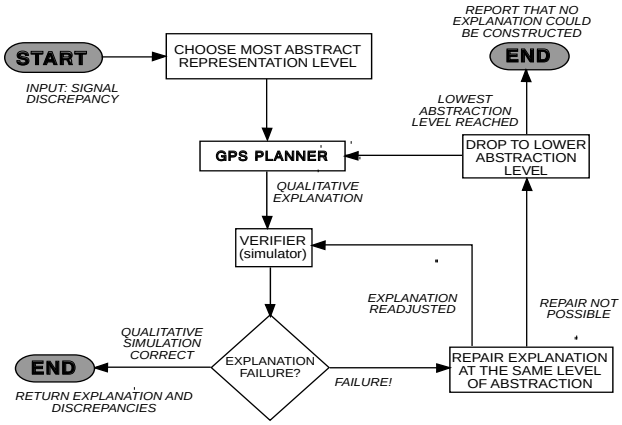


Figure 4.3. SUT Discrepancy Diagnosis KS Design.

as ground-truth and a *goal state* that represents the observed SPA output that is assumed to be somehow distorted. The diagnosis KS has a database containing operators that model various kinds of distortions that can result from improperly-tuned signal-processing control parameters. For example, in the context of the STFT algorithm, one of the distortion operators models a “frequency-resolution" distortion that occurs in the SPA output data when the window-length control parameter is not large enough to resolve two closely-spaced frequency components in a signal (see figure 4.4). The diagnosis process hypothesizes a sequence of operators, which when applied to the initial signal state will yield the distorted goal state. The search for the sequence is carried out using progressively more complex abstractions of the initial and final states, until finally an abstraction level is reached where an operator sequence can be generated using no more signal information than is available at that level. That is, the diagnosis process mimics the diagnostic reasoning of experts in that they first offer explanations (i.e., operator sequences) that are as uncomplicated as possible[Peng and Reggia, 1986].

Once a candidate sequence has been obtained, the diagnostic process enters into its *verify phase*. At this point, the diagnostic process “drops” to to the lowest abstraction level at which a description of the initial state is known. Verification proceeds as a degenerate case of the GPS algorithm at this lowest abstraction level. That is, no real “operator search” is carried out: the “search” algorithm simply selects operators in the order they appear in the candidate

operator sequence. This phase verifies that the pre- and post-conditions of each operator are met even when all information about the initial and final states is considered. If verification succeeds, the diagnosis process returns the candidate operator sequence as its final answer. If verification fails at some point, however, the diagnosis process attempts to "patch" the operator sequence by building a new sequence that eliminates the unmet conditions observed in the original sequence. This new sequence is then inserted into the original operator sequence and verification continues.

One issue not originally dealt with in [Nawab *et al.*, 1987] that arises in the IPUS framework is the problem of incorrect explanations. Sometimes the first explanation offered by the diagnosis process will not enable the reprocessing mechanism to eliminate a discrepancy. In these cases, IPUS may decide to reactivate the diagnostic process and provide the incorrect explanation as one that must not be returned again. To prevent the diagnosis process from repeating the same search it performed when it originally generated the incorrect explanation, the system stores with the explanation the search-tree context it was in when the explanation was produced. Then, the diagnosis process simply "starts up" from that point in the search space when it begins considering operators for a new explanation.

Another extension to the original work concerns the use of diagnostic knowledge to modify expectations for how future support evidence should appear under the current parameter settings. Each distortion operator contains a logical "support specification" of how data that is expected can appear distorted when processing parameters take on the current parameter values. When a distortion-operator sequence is specified, each operator's support-specification is combined to form a single specification that is used to annotate the expectation units for the hypothesis involved in the original discrepancy. This annotation serves to locally modify the high quality-level usually required by the system for all evidence for any expectation. That is, the specification permits the system to use less-clear evidence (*without* raising a discrepancy) for supporting its near-future expectations about the sources currently involved in the discrepancy.

4.2.4.1 Diagnostic Distortion Operators

The testbed instantiation of the IPUS diagnosis component models how SPA outputs can be distorted by poor parameter settings with a database of distortion operators. When applied to an abstract the description modified to contain the operator's distortion. The SUT discrepancy diagnosis KS uses these operators in a means-ends analysis framework [Nawab *et al.*, 1987] to "explain" discrepancies. The KS takes two inputs: an *initial state* representing anticipated correlates and a *goal state* representing the computed correlates. The formal task of the KS is to generate a distortion operator sequence mapping the initial state description onto the goal state description.

For an example of how a distortion operator is developed, consider the situation in Figure 4.4 where an STFT with an analysis window of W sample points is applied to a signal sampled at R samples per second. If the signal came from a scenario containing frequency tracks that approached closer than R/W Hz, Fourier theory predicts that the tracks will be merged in the STFT's computed correlates. Thus, the STFT's signal processing theory provides us with the concept of a "low frequency resolution" distortion process which can account for missing and unanticipated correlates in the STFT output. Table 4.5 lists all the distortion operators implemented in the SUT.

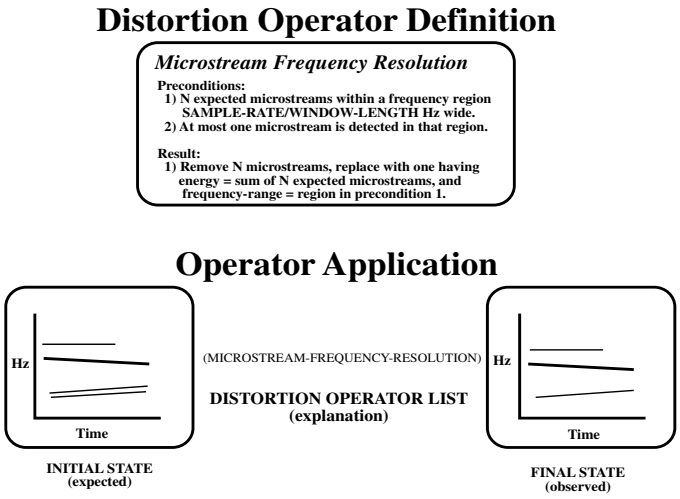


Figure 4.4. SUT Frequency-Resolution Distortion Operator.

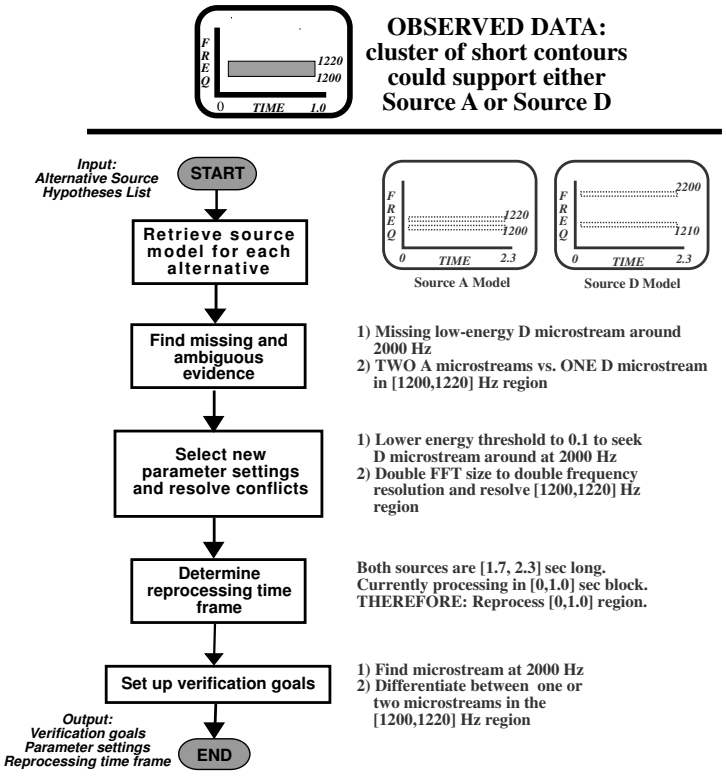


Figure 4.5. Sample Differential Diagnosis Execution.

4.2.5 Differential Diagnosis KS

One situation in which differential diagnosis can be used in IPUS occurs when a query to the source database returns more than one source model whose frequency components (or energy levels, or whatever other indexing feature is used) overlap the observed data. An abstract example of this situation in the auditory scene analysis domain appears in Figure 4.5 along with a possible execution trace for an IPUS differential diagnosis KS.

Spectral data from the current block in the $[1200, 1230]$ Hz range could support both sound A and sound D's existence. In such cases IPUS (under the RESUN framework) pursues a least-commitment interpretation strategy. For each retrieved model, an explanation hypothesis supported by an extension of the observed data is created, and on each support extension an alternative-extension-sou is recorded. These SOUs are left unresolved until the focusing heuristics deem its resolution appropriate to the current problem-solving context.

Table 4.5. SUT Distortion Operators.

DISTORTION	ASSOCIATED SPAS	DESCRIPTION
frequency resolution	STFT QSTFT LSTFT	Indicates that missing or unexpected tracks are caused by parameters providing inadequate frequency resolution for scenario.
time resolution	STFT QSTFT LSTFT	Indicates that missing chirps or extra-long merged tracks or bands are caused by parameters providing inadequate time resolution for scenario.
frequency thresholding	QSTFT LSTFT	Indicates that spectral data is missing because the QSTFT or LSTFT were not applied in the time-frequency region under question.
energy thresholding	MAX-PEAK THRESH-PEAK	Indicates that the SPAs' energy cutoff precludes finding a low-energy track.
peak thresholding	THRESH-PEAK	Indicates that a track is either missing or prematurely terminated because a new sound's higher-energy tracks keep the old sound's track peaks from being in the top N peaks selected from a spectrum.
wide neighborhood	THRESH-PEAK	Indicates that peaks are not found because they are sought for in a wide neighborhood; another nearby spectrum value was the local maximum.
track termination	CONTOUR MAX-PEAK THRESH-PEAK	Indicates that a track ends before reaching maximum duration due to the track's sound having started before the time the system started processing the acoustic stream.

The KS first compares the interpretation hypotheses to determine their overlapping regions. Any observed evidence in these regions is labelled "ambiguous." The KS then determines the hypotheses' discriminating regions (e.g., `Sound1`, and no other hypothesis, has a microstream at 2000 Hz). For each discriminating region where no evidence was observed, the KS posits an explanation for how the evidence could have gone undetected, assuming the hypothesized source was actually present. Using these explanations as indices into a plan database, the KS retrieves reprocessing plans and parameter values that should cause the missing evidence to appear. At this point the ambiguous evidence is considered. The KS seeks for multiple signal structures within each overlapping region (e.g., a region that contains data that could support one microstream of a hypothesis or two microstreams of another hypothesis), and selects processing plans to produce data with better structural resolution in the regions of overlap.

If the missing-evidence processing plan set and the ambiguous-evidence plan set intersect, the intersection forms the third element of the output triple. If the intersection is empty, the missing-evidence plan set forms the third element of the output triple. Finally, if the missing-evidence plan set is empty, the ambiguous-evidence plan set is returned. The rationale behind this hierarchy of plan set preference is that this ordering will return the most likely plans for producing evidence that could eliminate interpretations from further consideration. The region of mutual temporal overlap for the alternative hypotheses defines the reprocessing time region in the output triple, and the ambiguous and missing data that is handled by the reprocessing plan set defines the support evidence in the output triple. The output triple's reprocessing plan is then executed as in the reprocessing KS until either the parameter-value limits are exceeded or at least one of the pieces in the support evidence set is found after a reprocessing.

4.2.6 Reprocessing Strategies

For each distortion operator defined for the discrepancy diagnosis and differential diagnosis KSs, there is a control plan implementing a sequence of generic SPAs (i.e. they are specified with no predetermined control parameters) that will produce data at the level of abstraction at which

the distortion is observed. For all of the operators, these are the levels at which the "associated SPAs" in Table 4.5 produce output. In addition, there are reprocessing plans specified for the plausible two-operator explanations that can be produced by the diagnosis KSs. Explanations of three or more operators, while theoretically possible, have not been encountered during SUT test runs, and therefore were not considered in this version of the SUT.

Reprocessing plans are supplied for the following diagnoses that could be returned from either the discrepancy diagnosis and differential diagnosis KSs:

1. (end-of-data-boundary)
2. (energy-thresholding)
3. (frequency-resolution)
4. (time-resolution)
5. (peak-thresholding)
6. (source-termination)
7. (peak-thresholding energy-thresholding)
8. (time-resolution frequency-resolution)

Note that the last two possible explanations have no order imposed on their component operators.

In the interest of space only one involved example of the signal theory incorporated in the reprocessing plans is presented here. The discussion describes in detail one of the match heuristics for the spectrogram-output SPAs, and one of the variable-focusing heuristics for selecting reprocessing SPA control parameter values.

At points in reprocessing plans where more than one SPA could be used to produce data at a particular level of abstraction (e.g. according to Table 4.3 there are four SPAs that could produce output at the spectrogram level), match-focusing heuristics are defined to select

SPAs based on a cost criterion. In the basic SUT evaluated in Section 5.2, reprocessing-plan match heuristics favor the STFT SPA, unless a narrow frequency range of the spectrogram is being reprocessed that will require at most K frequency channels to be calculated, where $K <$ `FFT-Size`. K includes not just the number of frequency channels for the current microstream being reprocessed, but also at least three channels apiece for all other microstreams with diagnosable SOUs in the current block. In those cases the LSTFT (see Table 4.3 is selected. This preference is an attempt to minimize the total amount of multiplications and additions performed by the SUT's spectral analysis. It can be shown ([Oppenheim and Schafer, 1989], Chapter 9) that a partial spectrogram produced from repeated direct ($O(n^2)$ complexity) computation of the Discrete Fourier Transform at K channels will be faster than that produced from the FFT ($O(n\log(n))$ complexity) when fewer than $\log(N)$ frequency channels are required, with $N =$ `FFT-Size`. If at least 3 channels are necessary to determine whether a reprocessed spectrum value is a local maximum for inclusion in a microstream, the focusing heuristic conservatively estimates the number of frequency channels to be calculated for a block during reprocessing, assuming that every microstream with a diagnosable SOU will result in spectrogram reprocessing (an overly conservative assumption). This heuristic, therefore, can reduce reprocessing costs in portions of acoustic scenarios with few microstreams. In the SUT versions evaluated in Section 5.3, reprocessing-plan match-heuristics for one of the versions favor the QSTFT, while those for the other favor the STFT.

The signal-processing theoretic rules relating values of the reprocessing SPAs' control variables to desired SPA outputs are represented in focusing heuristics for the reprocessing plans. For example, there is a variable focusing heuristic for the STFT which selects a value for the `Window` parameter based on the frequency separation F between a microstream being reprocessed and its nearest neighbor in the current block. According to Fourier theory, an STFT with analysis window of length W applied to data sampled at R Hz will resolve spectral tracks no closer than R/W Hz. Thus, the focusing heuristic selects a value for `Window` that is at least R/F. Further, since the `FFT-Size` parameter must be a power of 2, the

`Window`-value heuristic selects as window length the lowest power of two X that is greater than R/F and less than the `FFT-Size` value, if that value has already been determined. If it has not been determined, simply the lowest power of two greater than R/F is selected. A variable-focusing heuristic for `FFT-Size` selects a power of two that is at at most twice the value of the desired `Window` value.

CHAPTER 5

EVALUATING THE SUT

This chapter presents three experiment suites to evaluate IPUS' performance within the sound understanding domain. The first section describes how the evaluation scenarios for the first two experiment suites were generated and defines the basic statistics measured for each experiment. The next three sections present the suites themselves. The first experiment suite explores how the SUT defined in Chapter 4 behaves with and without the reprocessing-loop components. The second suite examines the utility of approximate processing techniques in IPUS. The third suite explores the relationship between the exhaustiveness of the parameter search performed within the front end and the quality of the interpretation results. The final section summarizes the experimental results.

5.1 Basic Experiment Design

The design of the experiment suites involves two phases, each of which in turn has two library styles. This section first defines the experiment phases, and then discusses the library styles. In the first phase a version of the SUT is applied to 40 acoustic scenarios generated from the individual sounds in the SUT library, and in the second phase the same version of the SUT is applied to 15 five-second random acoustic scenarios.

5.1.1 Phase I Experiments

Each Phase I scenario contains a "single" sound. The term "single" is qualified because in some cases, such as clock-ticks or policecar sirens, the scenario will contain a script of several related sound instances. The scenarios are generated by randomly choosing an instance of the sound used to generate the acoustic models.[1] Regardless of content, each scenario signal is

[1] At least 5 instances were used for each sound's model. Appendix A provides more details on the construction of the library.

amplified so that all scenarios have the same average power. Phase 1 scenarios can range in length from 1 to 5 seconds, and are always an integer number of seconds long. If a library sound can last more than 5 seconds, the scenario for it is five seconds long, consisting entirely of that sound. If a library sound is less than 5 seconds long, the scenario for it lasts as long as the minimum integer number of seconds that spans the sound's duration.

Results from the Phase I experiments provide a minimum level of competency for a SUT version with respect to isolated sounds. This level of competency is useful for at least two reasons. First, it verifies the adequacy of the sound models for isolated sounds for the testbed version. Second, it establishes best-case recognition expectations for Phase II runs.

5.1.2 Phase II Experiments

Fifteen scenarios are generated once and used in the Phase II experiments for all suites. These scenarios are intended to produce anecdotal results that indicate how well the testbed version handles "complex environments." For this thesis a complex acoustic environment is one that has the potential for sounds that share some frequency content to be present at overlapping time periods. Thus, the following 5-step method was used to generate the scenarios. First, four "sounds" (scripts and/or isolated sounds) were randomly selected from the SUT library. Second, a random instance of each sound was selected from the corpus of instances used in the 40 single-source scenarios of Phase I. Third, start-times for each instance were randomly selected with uniform distribution within a 7-second base timeframe. Fourth, a 5-second window was randomly chosen within the base timeframe such that all four sound instances were included for at least their length or 1 second, whichever was shorter. If the start times precluded such a window, steps 3 and 4 were repeated until a 5-second window fitting this criterion was obtainable. Fifth, as with the scenarios in the first phase, each scenario was scaled so that all have the same average power. Figures 5.1 through 5.4 show the 15 scenarios generated for Phase II. Each scenario has at least four sound instances.

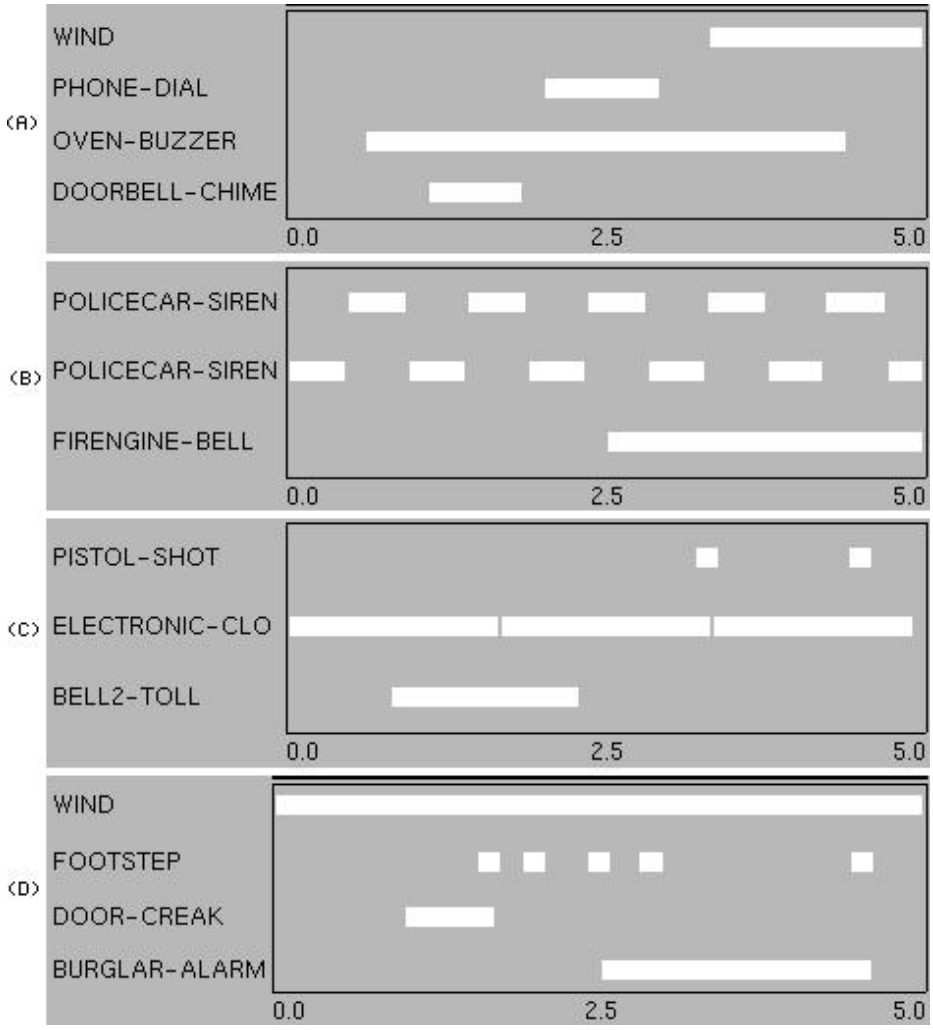


Figure 5.1. Phase II Scenarios, Part 1.

5.1.3 Library Styles

While the phases of each experiment suite help to illustrate a SUT version's ability to manipulate library models in simple and complex scenarios, the library styles for each phase help to show how library content influences the SUT version's interpretation search. There are two library styles in each experiment suite: **minimum** and **maximum**. When a scenario is run on a testbed version with a minimum-style library, the sound model library available to the testbed contains *only the models corresponding to the sounds in the scenario.* When the scenario is run with a maximum-style library, the library available to the testbed contains *all 40 sound*

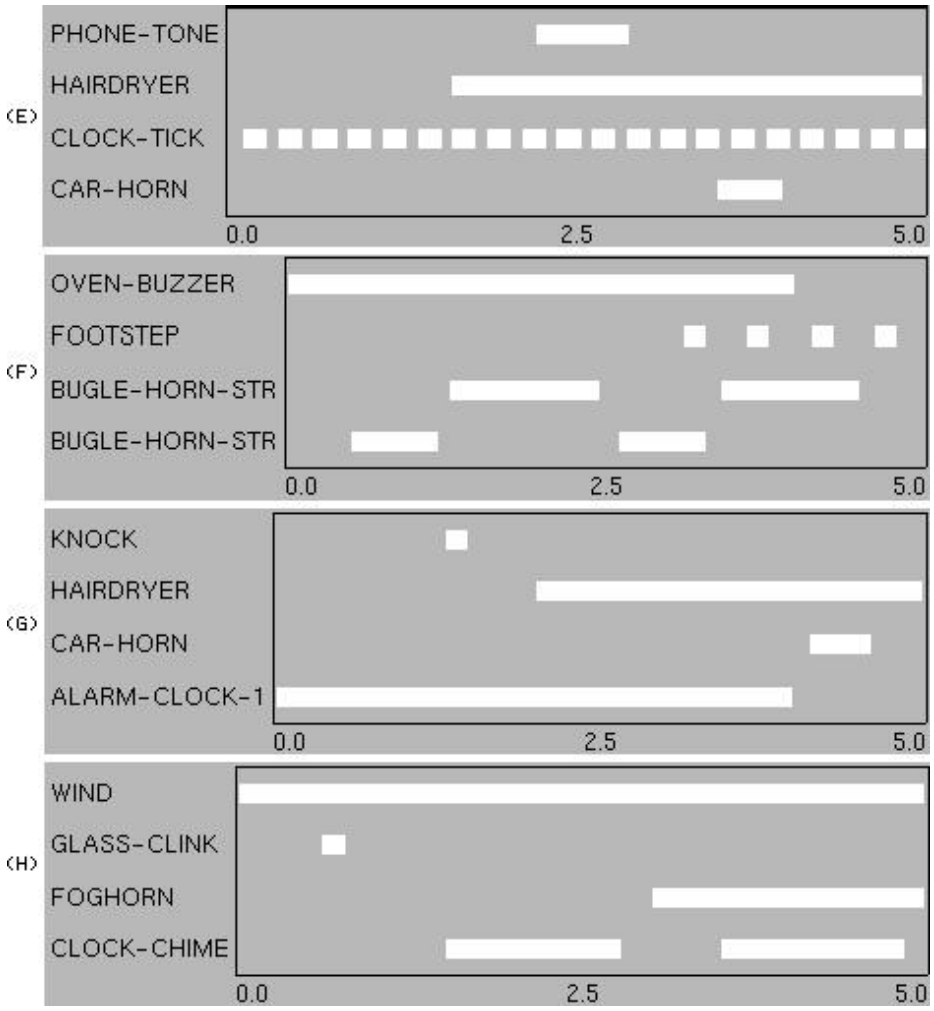


Figure 5.2. Phase II Scenarios, Part 2.

models. In both cases, the SUT version under study has no additional information about either the total number of sounds in the scenario, or about any correlation between the number of sound instances in the scenario and the models in the library.

5.1.4 Experiment Statistics

The hypothesis set taken as a scenario's final interpretation for both experiment phases is the set of all "answer" hypotheses (as defined in Section 4.2.2) whose summary ratings are at least 0.2 (out of a possible 1.0) and are greater than their negative-evidence ratings. Based on these final interpretations, each experiment suite reports the following statistics for the SUT versions' performance:

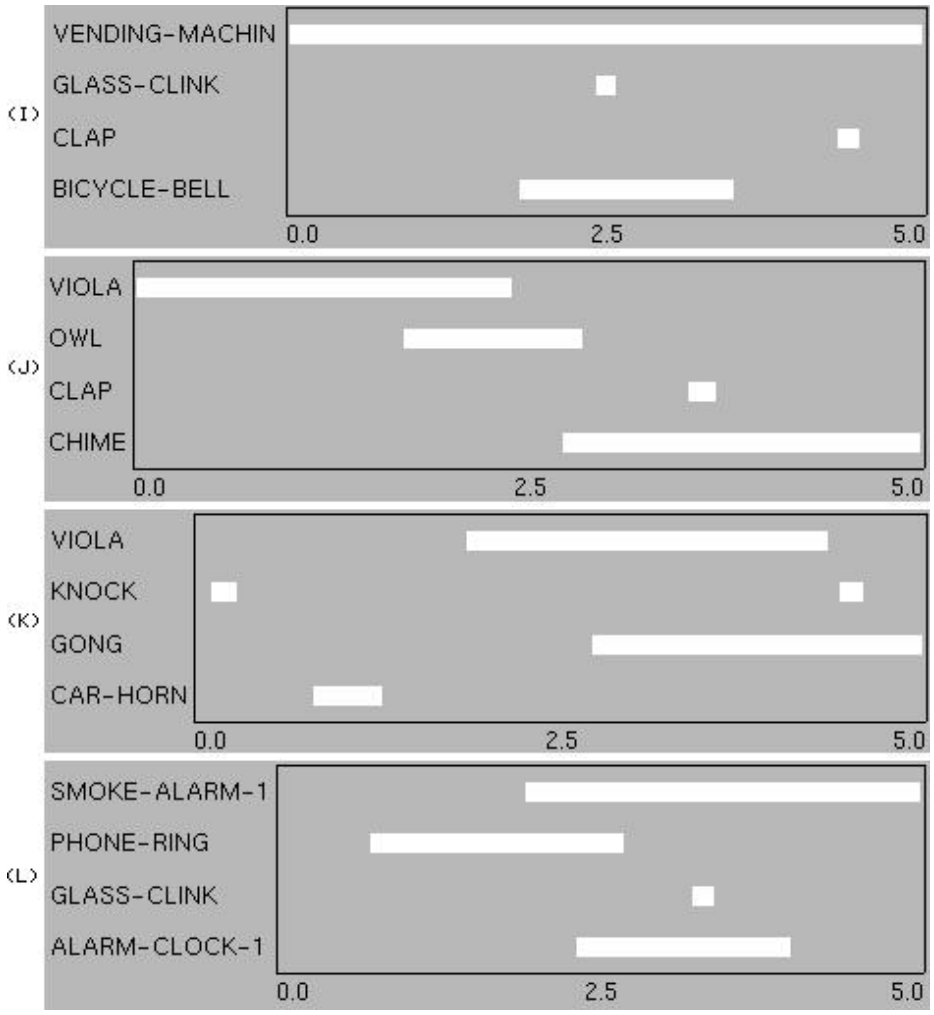


Figure 5.3. Phase II Scenarios, Part 3.

Hit Rate: the ratio of the number of correctly-identified scenario sound instances to the total number of sound instances in the entire run. An instance is considered identified if, after the scenario run, it is temporally overlapped by some answer hypothesis of the same type. If an answer hypothesis overlaps more than one instance, it is identified only with the instance it overlaps the most.

False Alarm Rate: the ratio of the number of answer hypotheses that do not identify a sound instance in the scenario to the total number of answer hypotheses (hits and false alarms) given for the run.

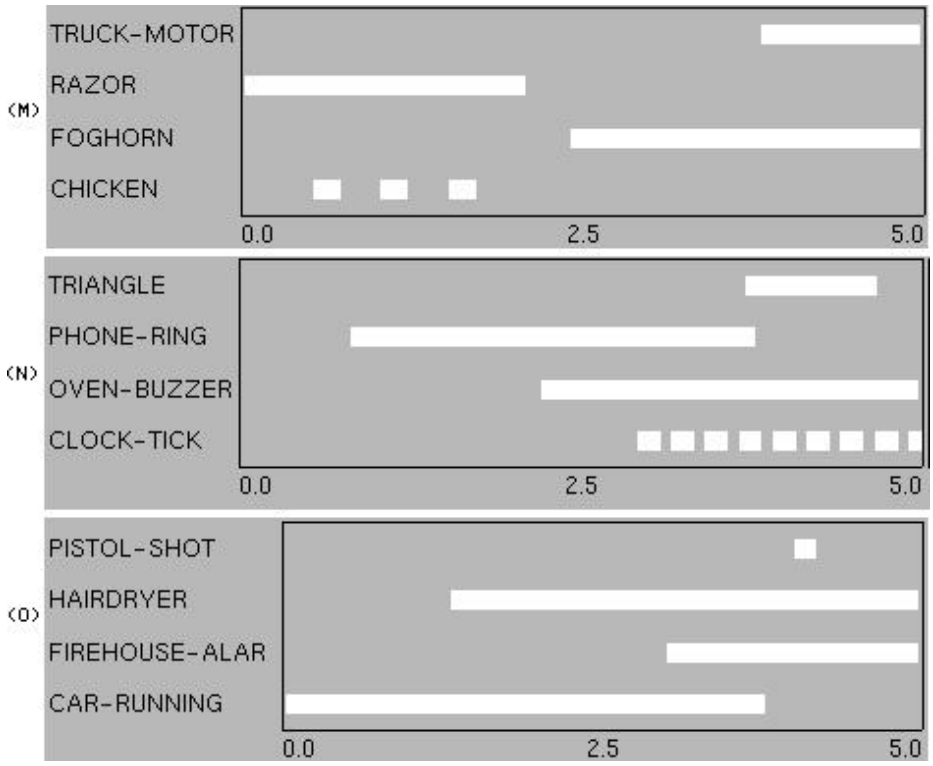


Figure 5.4. Phase II Scenarios, Part 4.

Track+ Rate: the ratio of the amount of time for which all "hit" sound instances were tracked to the total amount of time for which all sound instances in all runs lasted. A "hit" sound instance is considered to have been tracked for the amount of time its corresponding answer hypothesis overlaps it.

Overshooting: the ratio of the amount of time for which all answer hypotheses *do not* overlap their corresponding scenario sound instance to the total time covered by *all* answer hypotheses (hits and false alarms) in all scenarios.

False Tracks: the ratio of the time covered by all false alarm answer hypotheses to the total time covered by *all* answer hypotheses (hits and false alarms).

Track- Rate: *Overshooting* + *False Tracks*.

Answers: the number of reported answer hypotheses (both false alarm and hits) over the number of sound instances in all scenarios.

Nonanswers: the number of hypotheses from all scenarios considered but ultimately rejected as answers by the SUT over the number of sound instances in all scenarios.

Total Hyps: *Answers* + *Nonanswers*. The total number of sound hypotheses considered over the number of sounds in all scenarios.

Disc. Diag. Rate: the number of discrepancy diagnoses over all scenarios' sounds.

Diff. Diag. Rate: the number of differential diagnoses over all scenarios' sounds.

Param Contexts: for all blackboard hypothesis levels below *microstream* and *noisebed*, these are the average number of distinct reprocessing parameter contexts *per scenario*. Only focused contouring performed during a reprocessing plan's execution is considered as an instance of that SPA's reprocessing.

5.2 Suite 1: Effects of Reprocessing Loop

The centerpiece of the IPUS framework is its reprocessing-loop. The components in this loop and the signal processing knowledge they manipulate are what enable IPUS to integrate front-end parameter search with interpretation search. It is important, therefore, to evaluate what effect the reprocessing loop has on the quality of SUT interpretations, and what relationships exist between scenario complexity and reprocessing frequency.

The experiment suite in this section uses two versions of the SUT to examine the influence of the reprocessing loop on interpretations. The first version is the one described in Chapter 4, and the second version is identical to the first except that

- the variable focusing heuristic for ?psm-sou of the `Have-PSM-SOU-Resolved` subgoal at the PSM-contol-plan level (Figure 2.6) will not select uncertain answers with negative evidence.

- support-exclusion-sous are not selected by focusing-heuristics related to the `Resolve-Uncertain-Hypothesis` plan hierarchy.
- the rough diagnostic tests performed by interpretation KSs are disabled; missing or incomplete evidence is assumed to not be present.
- discrepancy-detection tests associated with the front-end SPAs are disabled.

Thus the second version does not have access to the signal processing theory in the reprocessing-loop, nor can it mitigate the effect of possibly explainable negative evidence.

The two versions were run in all four phase/library-style combinations. All scenarios were sampled at 16 KHz. Both versions started with the front end in Table 5.1. Refer to Tables 4.3 and 4.4 for a review of the parameters' purposes.

Table 5.1. Suite 1 Front End.

SPA	Parameter and Value
STFT	Window: 1024 data points Decimation: 128 data points FFT-Size: 2048 data points
THRESH-PEAK	Threshold: 0.03 Limit: 7 peaks Neighborhood: 3 channels
BAND	Threshold: 3 peaks Max-Dead-Time: 0.08 seconds
ENV	Order: 2
SHIFT	Rel-Diff: 0.22

The particular default front end was chosen so as to give the no-reprocessing SUT version the parameters and SPAs for determining microstream entropies (STFT) and for generating the number of peaks necessary for tracking the sounds with the most tracks (THRESH-PEAK). No single sound in the SUT library has more than 7 microstreams. No contouring parameters are provided since the control strategy in all testbed versions uses focused contouring to generate

support for microstreams. The 128-point decimation is provided so that the no-reprocessing version will have the correct spectral support available for noisebeds. The scenarios are all analyzed in 1-second blocks of data.

Table 5.2. Suite 1, Phase I: Results From 40 Isolated-Sound Runs.

	Minimum Library		Maximum Library	
Suite:	Reprocessing	No Reprocessing	Reprocessing	No Reprocessing
Hit Rate:	0.79 (0.85)	0.73 (0.8)	0.79 (0.85)	0.63 (0.65)
False Alarm Rate:	0	0	0.38	0.32
Track+ Rate:	0.66	0.64	0.65	0.54
Track- Rate:	0.03	0.05	0.18	0.15
Overshooting:	0.03	0.05	0.03	0.05
False Alarms:	0	0	0.15	0.10
Total Hyps:	1.04	1.45	7.32	8.18
Answers:	0.79	0.73	1.26	0.93
Nonanswers:	0.22	0.72	6.06	7.25
Disc. Diag. Rate:	2.51	0	5.82	0
Diff. Diag Rate:	0	0	0.03	0
Param Contexts:	1.53	0.00	4.10	0.00
Spectrogram:	0.50	0.00	0.95	0.00
Peak:	0.50	0.00	0.85	0.00
Boundary:	0.15	0.00	0.15	0.00
Contour:	0.38	0.00	1.15	0.00

Table 5.2 summarizes the results for the first phase's benchmarking runs. The data illustrate the basic competence of the SUT and the reprocessing loop's utility with respect to isolated instances of the sounds in the library. We first address two apparent inconsistencies in Table 5.2 concerning hit rates and differential diagnosis rates before discussing the primary question of the reprocessing loop's utility and cost.

At first glance one might expect the hit rates for the minimum library runs to be very close to or equal to 1.00. However, as mentioned in Section 5.1.1 several of the "single-source" scenarios actually contained more than one sound (e.g. the policecar siren note sequence, the clock-ticking tick sequence, etc.), and not all of these sound-sequence instances were correctly

matched to the sound models. In the 40 "single-source" scenarios there is in fact a total of 71 sound instances, and the total time duration of all instances is 110.45 seconds. If one were to determine hit rate on the basis of whether at least one instance in a sound-sequence was identified, the hit rate then becomes 0.85, accounting for 34 isolated sounds under minimum library conditions. The parenthesized values in the other hit rate entries also show this distinction in hit rate determination.

The remaining 6 sound instances that were not matched with their models under minimum library conditions with reprocessing were not identified because there were mismatches between observed contour energy levels and the microstream entropies specified for the sound models. The testbed KS that searched for contour support compared these expected entropies with those of any contours being considered as support for a microstream. Since the microstream model entropies are based on an average over the entire microstream (see Section 4.1.2.1), it is possible for local, short-time-duration contours that ought otherwise support the microstream to have out-of-bounds entropies. Interestingly, although these sounds appeared in at least one of the Phase II, minimum-library scenarios, at least one instance of each *was* recognized in the 15 complex scenarios because the low-energy, wideband frequency content of the other sounds modified the entropy of the found support contours. This relationship between local contour entropies and global microstream entropies is definitely a shortcoming in the modelling of sounds in the SUT; a better representation of the dynamics of microstream entropy needs to be developed. Under maximum library conditions, the entropy problem also holds for the same 6 sounds just discussed.

As expected, there is no differential diagnosis performed in the minimum library runs. However, the differential diagnosis rate for the maximum library runs appears rather low. In actuality this result shows the difference between implicit differential diagnosis occurring in the testbed's behavior and explicit executions of the differential diagnosis KS. Very few explicit differential diagnosis executions are being performed because the SUT effectively prunes many competing sounds in the course of performing discrepancy diagnosis and reprocessing for

missing evidence. By the time the SUT's behavior enters the fourth general control strategy phase (Section 2.3) during which explicit differential diagnosis occurs, the few sets of sounds in the current block that are still competing among themselves are ones whose microstream frequencies overlap too much for differential diagnosis to suggest reprocessing for higher frequency resolution or for entropy measuring.

With the two inconsistencies explained, an examination of the rest of Table 5.2 shows that the results support the intuitions that reprocessing should enhance signal recognition and tracking, and that increased library size should lead to more sounds being considered and more reprocessing work in verifying the sounds. That is, search in the interpretation state space and in the SPA parameter space increases in the face of library complexity. Between the minimum and maximum library runs for the reprocessing SUT there is a threefold increase in the number of reprocessing parameter contexts used. In addition, note that the overshoot tracking rate is lower for the SUT version that uses reprocessing; the detection of sound-termination and -onset times is improved.

However, there is a cost to the recognition and tracking improvements: an increase in false alarms and in false-alarm tracking. The false-alarm recognition and tracking rates for reprocessing translate to 34 additional sound instances tracked for a total of 12.7 seconds, while the false-alarm recognition and tracking rates for the non-reprocessed data translate to 21 additional sounds tracked for a total of 6.6 seconds. These higher false alarm rates for reprocessing can be attributed both to the SUT's sound-modelling framework and to the heuristic test used in the SUT at the end of the third general control strategy phase (Section 2.3) for determining the sufficiency of a data block's interpretation. As can be seen from the spectrograms in Appendix A, several long-term (3–5 second minimum duration) sounds (notably, `Alarm Clock 1`, `Car Running`, `Hairdryer`, `Oven Buzzer`, `Razor`, `Truck Motor`, `Vending Machine Hum`, and `Wind`), have moderately significant wideband spectral energy. Some of the spectrogram values in these sounds' wideband regions were labelled as narrowband peaks, and clustered into spectral bands. In the minimum-library runs,

these "wideband bands" do not lead to the retrieval of any false-alarms from the library, but in the maximum-library runs, they do. In fact, the long-term sounds just cited account for 15 of the reprocessing SUT's false alarms, and 9 of the no-reprocessing SUT's false alarms. In both SUTs, the heuristic test used at the end of the third general control strategy phase (Section 2.3) for determining the sufficiency of a data block's interpretation is the same: at least 70% of the energy in a block's spectral bands should be explained, if possible. While the no-reprocessing SUT accepts 9 sources because of extraneous spectral energy, the reprocessing SUT accepts not only the same 9 sources but also extends them and "hallucinates" another 10 sound instances, because its reprocessing capability permits it and its sufficiency heuristic and sound-models encourage it! Clearly the wideband energy of the cited sounds must be accounted for in improvements in sound modelling.

Even when the sound hypotheses resulting from model shortcomings are taken into account, the reprocessing SUT still produces 19 false alarms, 9 of which stem from the entropy-matching problem described under minimum library conditions.[2] The remaining 10 false alarms are attributable to "hallucinations" induced by the sufficiency heuristic. These false alarms, however, would probably not be removed by sound model improvements. They occur because when the SUT is forced to reach a threshold, there is no mechanism to prevent it from iterating this behavior over all sounds which overlap a few low-energy unexplained peaks:

- hypothesize a sound
- search for contour support and find none
- assume that support peaks are missing because they have (very) low energy
- perform discrepancy diagnosis

[2] That is, when the 6 sound instances were not recognized due to lack of contour evidence with the desired entropy, the testbed found 9 false alarms to account for their spectrograms.

- if the discrepancy diagnosis KS reports that indeed some distortion such as `energy-thresholding` or `peak-thresholding` is theoretically possible, reprocess with extremely low peak energy thresholds.

Eventually some sound model's relative microstream energies will permit such dogged reprocessing to "hallucinate" the presence of tracks.

Table 5.3. Suite 1, Phase II: Results From 15 Complex-Scenario Runs.

	Minimum Library		Maximum Library	
Suite:	Reprocessing	No Reprocessing	Reprocessing	No Reprocessing
Hit Rate:	0.61	0.46	0.59	0.47
False Alarm Rate:	0.02	0.02	0.39	0.40
Track+ Rate:	0.68	0.47	0.67	0.44
Track- Rate:	0.06	0.09	0.19	0.27
Overshooting:	0.04	0.07	0.05	0.08
False Tracks:	0.02	0.02	0.14	0.19
Total Hyps:	1.21	1.14	8.14	8.53
Answers:	0.56	0.46	0.86	0.79
Nonanswers:	0.65	0.68	7.28	7.74
Disc. Diag. Rate:	3.66	0	12.63	0
Diff. Diag. Rate:	0.01	0	0.03	0
Param Contexts:	9.40	0.00	29.14	0.00
Spectrogram:	2.00	0.00	3.20	0.00
Peak:	4.40	0.00	14.47	0.00
Boundary:	0.07	0.00	0.07	0.00
Contour:	2.93	0.00	11.40	0.00

Table 5.3 summarizes the results for the experiment suite's second phase: 15 complex-scenario runs. There is a total of 110 sound instances over all scenarios, with a total duration time of 114 seconds for all instances. As in the Phase I experiments the parameter context counts show an observable trend in extra reprocessing work being done as the library size increases, and the considered-hypotheses counts show a concomitant increase with library size. The table's tracking and recognition rates both show improvements due to the reprocessing loop.

The observation that hit rates reported for the runs are low, but improve up to a 60% threshold with reprocessing, might indicate that there is some interaction among sounds in the scenarios that is not being handled well by either the reprocessing strategies or the available testbed KSs. Inspection of the scenarios' interpretation traces bears out the latter culprit. Of the 110 total sound instances, 48 are impulsive (clock-tick sequences, etc.). The SUT's shift-detection SPA depends on relative-percent changes between points in the waveform-envelope to identify energy shifts. In these runs the percent threshold was too high for detecting shifts in the waveform envelope when they were superimposed on a narrowband sound's energy level. This leads to 25 impulsive sounds being missed because both the time-domain and the frequency-domain front-end SPAs do not generate correlates that might have indicated to discrepancy tests that signal information is being missed.

Since the complex scenarios were deliberately designed to have a greater incidence of sound interactions that violate SPA parameter-setting assumptions than single-source scenarios, one would expect that the interpretation of complex scenarios would benefit from reprocessing more than would the interpretation of single-source scenarios. In Phase I, though, the reprocessing loop improves the hit rate by $1 - (.79/.63) \approx .25$ whereas in Phase II the improvement is also $1 - (.59/.47) \approx .25$. This is only an apparent contradiction to the expectation, however, once the tracking rate improvements are taken into consideration. In Phase I the reprocessing loop improved the tracking rate by $1 - (.65/.54) \approx .20$ while in Phase II the reprocessing loop improved tracking by $1 - (.67/.44) \approx .52$. The justification for using tracking rate improvement instead of hit rate improvement to verify the expectation is that a hit only requires that *some* time-region of the sound be identified correctly. Thus, the SUT with no reprocessing can attain a somewhat higher hit rate when sounds that might otherwise interfere with each other's spectral signatures do not completely overlap each other in time. The tracking rate, on the other hand, indicates more reliably how much of each sound was correctly tracked *and* identified, and therefore ought to be used to verify the expectation that reprocessing should show greater benefit in complex scenarios.

The false-alarm rates of the reprocessing and non-reprocessing SUT versions are nearly the same in Phase II, while in Phase I's runs the reprocessing SUT version had a false-alarm rate that was marginally higher than that of the non-reprocessing version. In that case the claim was made that the interpretation sufficiency heuristic forced the reprocessing SUT to "hallucinate" more sounds in simple scenarios in an attempt to meet the heuristic's minimum explained-energy requirement. In the Phase II results the nearly equivalent false-alarm rates lend support to the claim that in complex scenarios the sufficiency heuristic at the worst does not lead to a higher incidence of "hallucination," and at the best forces the system to consider hypotheses and to reprocess for undistorted evidence that might otherwise be ignored. Although more complete experimentation is necessary to fully substantiate the position, one could argue that these results show the IPUS architecture is so strongly oriented toward handling complex signals that the framework overanalyzes simple scenarios to the detriment of its performance.

5.3 Suite 2: Approximate Front Ends

As discussed in Chapter 1, approximate processing [Decker *et al.*, 1990] refers to the deliberate limitation of search processes in order to trade off certainty for reduced execution time. Approximate SPAs are algorithms whose processing time can be limited in order to trade off precision in their output correlates for reduced execution time. The availability of approximate SPAs permits the formulation of SUT control strategies that first use approximate SPAs to generate a rough picture of the environment that is refined only where the front-end correlates' interpretations are too uncertain. Refinement is achieved by reprocessing these limited signal portions with SPAs that produce correlates having greater precision. These non-approximate SPAs would ordinarily be quite expensive if applied to the entire signal, but when they are applied only in restricted signal regions their costs become manageable. An issue to address, however, is whether the reliance on an approximate front end significantly increases the amount of interpretation search performed by the SUT.

This section explores how an approximate SPA such as the Quantized Short Time Fourier Transform (QSTFT) [Nawab and Dorken, 1995] may be used within IPUS, and presents a

limited answer concerning the nature of the relationship between reduced time in the front end and the amount of interpretation search. The QSTFT is an SPA that can compute an estimate of a signal's STFT using an order of magnitude fewer addition-operations than the STFT and *no* multiplication-operations. This performance is achieved in general by applying the $O(n^2)$ version of the Fourier Transform algorithm with precomputed Fourier coefficients to a signal quantized to the set (-1, 0, 1); the interested reader is referred to Nawab's work [Nawab and Dorken, 1995] for more details. When compared to STFTs with `FFT-Size`'s up to 256 points, the evaluation of the basic QSTFT for the entire frequency-domain spectrum takes fewer additions, and no multiplications. A version of the QSTFT that only computes the time-frequency spectrogram regions surrounding the estimated highest-energy frequency can remain competitive with the complete-spectrogram STFT up to *FFT-Size*s of 1024.

In this experiment suite, two versions of the SUT are compared. The first version (termed the "precise" version) has the same front end as those explored in the first suite, except that the STFT's `FFT-Size` parameter is defaulted to 512 points, and the analysis `Window` parameter is defaulted to 256 points.[3] The second version (the "approximate" version) replaces the STFT SPA in the front end with a QSTFT whose `FFT-Size` parameter is defaulted to 512 and whose computed frequency region over each waveform time slice covers 1000 Hz on either side of the frequency with the estimated highest energy. This information is summarized in Tables 5.4 and 5.5. This section only considers Phase II experiment evaluation, presented in Table 5.6.

An examination of Table 5.6 shows that both reduced frequency-resolution systems maintain recognition rates comparable to those obtained by the SUT from Section 5.2. Indeed, the reduced-resolution SUTs obtained slightly higher recognition rates because they did not use the entropy measures of instances of the six problematic sounds mentioned earlier for verification. The reason for this is that the spectral evidence for the sounds came from a context that did not match that required for verification of the entropies, and the

[3]That is, this version's front end produces spectrograms with one-quarter the frequency resolution capability of the version in Section 5.2.

Table 5.4. Suite 2 "Precise" Front End

SPA	Parameter and Value
STFT	Window: 256 data points Decimation: 0 data points FFT-Size: 512 data points
THRESH-PEAK	Threshold: 0.03 Limit: 7 peaks Neighborhood: 3 channels
BAND	Threshold: 3 peaks Max-Dead-Time: 0.08 seconds
ENV	Order: 2
SHIFT	Rel-Diff: 0.22

Table 5.5. Suite 2 "Approximate" Front End.

SPA	Parameter and Value
QSTFT	QWindow: 256 data points QDecimation: 0 data points QFFT-Size: 512 data points Freq-Radius: 1000 Hz
THRESH-PEAK	Threshold: 0.20 Limit: 7 peaks Neighborhood: 3 channels
BAND	Threshold: 3 peaks Max-Dead-Time: 0.08 seconds
ENV	Order: 2
SHIFT	Rel-Diff: 0.22

CONTOUR-MICROSTREAM KS did not automatically verify the results during initial contouring. However, one must note that the reduced front-end resolution did exact a toll in the interpretation search space: a slight increase in the false-alarm rate and nearly double (for the approximate SUT) the number of hypotheses considered per scenario. The effect of this doubling on system execution time will be discussed later in the section.

The reduced-frequency resolution SUTs showed slight reductions in the amount of track overshooting as compared with the higher-resolution SUT in Section 5.2. This is due to the

Table 5.6. Suite 2, Phase II: Results for 15 Complex Scenario Runs with Low-Resolution Front End.

	Minimum Library		Maximum Library	
Suite:	Approximate	Precise	Approximate	Precise
Hit Rate:	0.60	0.65	0.56	0.61
False Alarm Rate:	0.08	0.11	0.42	0.43
Track+ Rate:	0.72	0.77	0.67	0.65
Track- Rate:	0.06	0.08	0.24	0.22
Overshooting:	0.02	0.02	0.04	0.04
False Tracks:	0.04	0.06	0.20	0.18
Total Hyps:	2.23	1.66	16.48	14.10
Answers:	0.65	0.73	0.97	1.07
Nonanswers:	1.58	0.93	15.51	12.03
Disc. Diag. Rate:	7.90	4.46	35.30	23.16
Diff. Diag. Rate:	0	0	0.13	0.27
Param Contexts:	35.67	11.01	66.59	31.08
Spectrogram:	9.53	2.47	13.70	7.45
Peak:	22.87	5.27	33.42	12.33
Boundary:	0.07	0.07	0.07	0.07
Contour:	4.20	3.40	19.40	13.23

fact that, as predicted by Fourier theory, the SUTs in this suite had front ends with better time resolution than that provided by the SUT in Section 5.2.

When interpreting scenarios with a minimum library, the approximate SUT required only 32% of the mathematical operations (additions and multiplications) that the precise SUT required, and experienced a 5% loss in hit rate. When interpreting scenarios with a maximum library, the approximate SUT required 41% of the mathematical operations that the precise SUT required, with the same difference in hit rate. Additionally, under maximum-library conditions the approximate SUT required 9% of the operations required by the SUT version tested in Section 5.2 under the same conditions. These observations lend support to the claim that the IPUS architecture can apply approximate SPAs effectively enough to reap computational savings at moderate interpretation expense. An examination of the sounds that were missed by the approximate SUT reveals that all of the sounds were missed because they had only one or two microstreams and they fell outside the frequency window produced by

the front-end QSTFT. These missed sounds indicate that the fault-discrepancy tests which compare time-domain energy and frequency-domain energy of selected peaks might need to incorporate heuristic tests with higher energy thresholds.

Note that although the number of spectrogram reprocessing parameter contexts executed per sound instance for the approximate SUT is higher than that for the precise SUT, the majority of those extra contexts were reapplications of the QSTFT in very narrow-band spectral regions outside of the region originally produced by the front end. Hence, the computational cost of the extra QSTFTs was minimal.

5.4 Suite 3: Effects of Front-End Complexity

One can define a continuum of practical signal interpretation systems based on the relative complexities of their front ends and their interpretation components [4]. At one end of the continuum are systems (Type I) with front ends that are designed to produce correlates that are optimal in some sense in order to minimize the search complexity of their interpretation component. At the other end of the continuum are systems (Type II) with high search-complexity interpretation components that are designed to analyze correlates that are imprecise in order to minimize the complexity of their front end.

The experiment in this section anecdotally compares the front-end complexity, interpretation search complexity, and recognition rates of the SUT version from Section 5.2 with those of a Type I benchmark system. For this experiment a Type I version of the sound understanding testbed (SUT-1) was designed, having an optimized SPA for spectral analysis in the front end. SUT-1 also had its reprocessing disabled under the assumption that its front end would produce "optimal" evidence. With respect to Tables 4.3 and 4.4, SUT-1's full complement of front-end SPAs includes the ENV, SHIFT, ATF, THRESH-PEAK, BAND, and CONTOUR

[4] The continuum could have been generalized to a complexity **plane** with one axis representing front-end complexity and the other representing interpretation complexity. This plane would include both systems with complex interpretation components *and* complex front ends and systems with simple interpretation components *and* simple front ends. However, the former class of systems tends not to handle complex environments well, while the latter class of systems tends to have overall complexities that preclude online usage. Accordingly these are not covered in the thesis.

SPAs with fixed parameters tuned to environmental constraints. We briefly digress to justify the optimality of the ATF.

Jones and Parks [Jones and Parks, 1990] developed the ATF as an SPA that produces spectrograms (or, more generally, time-frequency representations) with maximum concentration of signal energy in time-frequency. Such representations are desirable since,

> "...concentrated components in general overlap or interfere with other nearby components as little as possible, and yield a "sharp" representation. Maximum concentration also implies that signals are confined as closely as possible to their proper support in time-frequency, which gives the interpreter more confidence in the time-frequency representation."([Jones and Parks, 1990], p. 2129)

As mentioned previously in Section 1.2, the length of the analysis window used by a time-frequency SPA such as the STFT determines a tradeoff between the time- and frequency-resolution in its output spectrogram. Figures 5.5a and 5.5b show the patterns of the amount of resolution (scale) offered by fixed-parameter instances of two commonly-used time-frequency SPAs: the STFT and wavelet analysis (WA) [Rioul and Vetterli, 1991]. In these figures the "scale" dimension represents resolution detail, or density of frequency sample points. Figure 5.5a shows the uniform resolution usually obtained by the STFT, while Figure 5.5b shows one possible nonuniform resolution that could be achieved by wavelet analysis.[5] In both cases regions with higher scale values indicate that those portions of the SPA's time-frequency representation have energy values from more-finely sampled points along the time-frequency plane.

The goal of the ATF SPA is to produce a composite spectrogram, or Generalized Short-Time Fourier Transform, where each point in the time-frequency plane has the energy value produced by an analysis window that maximizes the "peakiness," as measured by 2-dimensional kurtosis,

[5] The interested reader should note that Figure 5.5b shows the type of time-independent frequency resolution pattern that the human auditory system exhibits. The arguments at the end of Section 1.2 refer to this pattern.

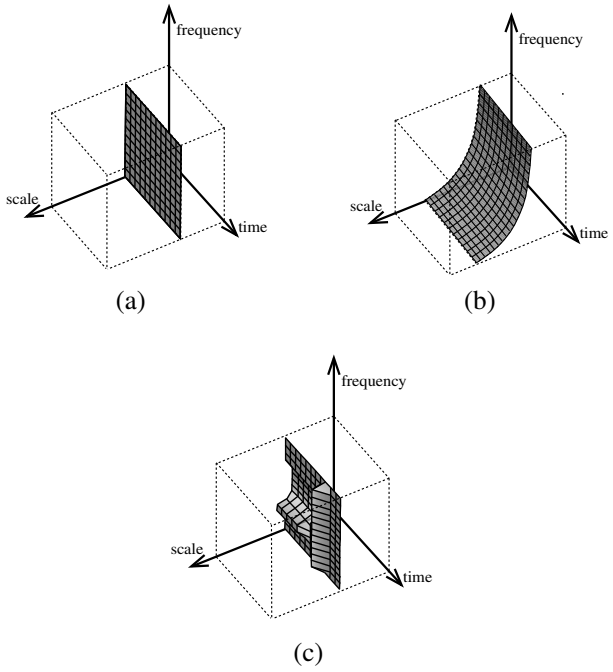


Figure 5.5. Sample Frequency Resolution Patterns.

of the spectrogram in the region immediately around that point. Kurtosis is a measure used in statistics to quantify how "pointy" a set of data values is, and is defined as

$$\alpha_4 = \frac{m_4}{(m_2)^2},$$

where m_2 and m_4 are the second and fourth moments about the mean of the distribution. The method by which the ATF SPA achieves this concentration can be summarized as follows. The SPA performs an exhaustive search of a list of given STFT analysis window lengths, generating each window's associated spectrogram. For each point in a spectrogram, the SPA then applies a Gaussian localizing window to the region around the point to weight the nearby points more than the distant points, and generates a kurtosis measure of the point's weighted neighborhood. Finally, the SPA compares the local kurtosis measured at each point in each window's spectrogram, and reports the energy value associated with the maximum kurtosis measure. Figure 5.5c shows the type of context-sensitive resolution attainable by the ATF through exhaustive search of the STFT parameter space.

Figure 5.6 shows an acoustic scenario whose interpretation can benefit from use of the ATF SPA in the front end. Figures 5.7a and 5.7b show the spectrogram obtained from short-window and long-window STFT analysis, respectively, of the acoustic scenario in Figure 5.6, while Figure 5.7d shows the scenario's sharper ATF spectrogram produced by adaptive use of the window lengths shown in Figure 5.7c. Darker regions in the spectrograms indicate higher energy, while in Figure 5.7c dark regions indicate 256-point analysis windows, grey regions indicate 512-point analysis windows, and light regions indicate 1024-point analysis windows. The 256-point window produces wide tracks that, in an environment where many of the expected frequency ranges of sounds' tracks could overlap each other, can have too much ambiguity in sound-model matches. However, the same window is desirable for producing the sharp evidence for the footsteps, pistol shot, and the time-varying doorcreak. The 1024-point window produces narrow tracks necessary for minimizing match ambiguity among competing narrowband sound hypotheses for the wind, the burglar alarm, and the steady behavior of the door creak, but "smears out" the time-dependent features of the door creak, footsteps, and pistol shot. The ATF spectrogram gathers sharp evidence for both the short-time and narrowband features of the scenario sounds, minimizing the work that an interpretation component would have to do to identify tracks of sounds.

Of course, there is a price for the ATF's quality: the exhaustive parameter-space search increases the front end's complexity. Jones and Parks [Jones and Parks, 1990] show that the number of real multiplications is of order $O(\frac{NM(\log N/S+\log M)}{S})$, where N is the length of the signal being analyzed, M is the points taken in one FFT analysis step of the STFT, and S is the maximum STFT analysis window length considered.

In comparing the performance of Section 5.2's SUT with that of SUT-1, we note that the ATF's time requirements made it infeasible to apply SUT-1 to all of the 15 Phase 2 scenarios. Instead, we offer the following observations after applying it to four scenarios with a full library. When interpreting scenarios containing low background noise sounds, such as (K) and (L) in Figure 5.3, the SUT-1 produced better tracking results (0.00 overshooting,

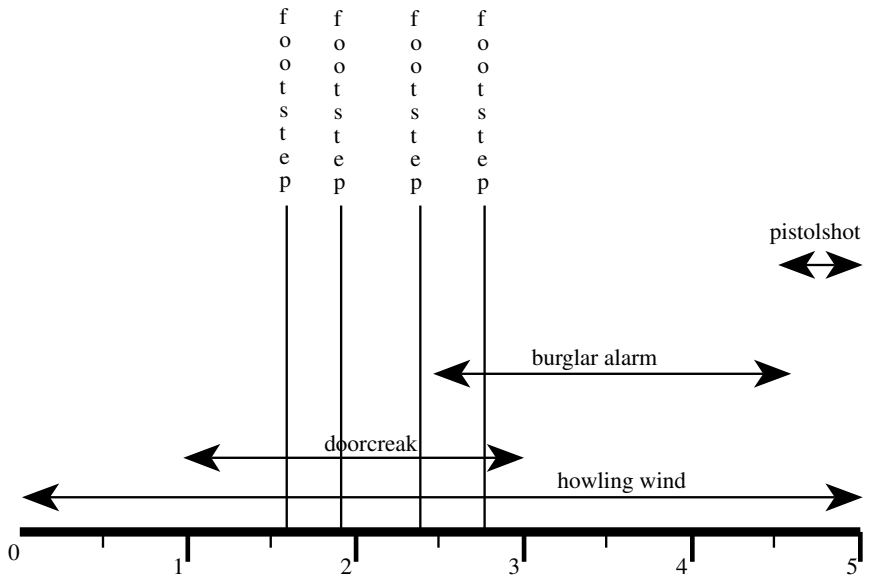


Figure 5.6. ATF Example Scenario.

0.04 false-alarm tracking) and higher hit rates (100% in this case, due to reduced confusion from fewer frequency- and time- resolution problems) than the standard SUT, at the cost of $O(10^3)$ times more mathematical operations. However, when the ATF-based system analyzed scenarios containing sounds with significant background wideband noise, such as (G) in Figure 5.2 and (M) in Figure 5.3, it produced higher false-alarm rates (4 more sounds were "hallucinated") than the SUT from Section 5.2 because its emphasis on producing sharp spectrograms created spurious peak tracks in the "noisier" wideband spectral regions around the hairdryer, truck-motor, and foghorn in the scenarios.

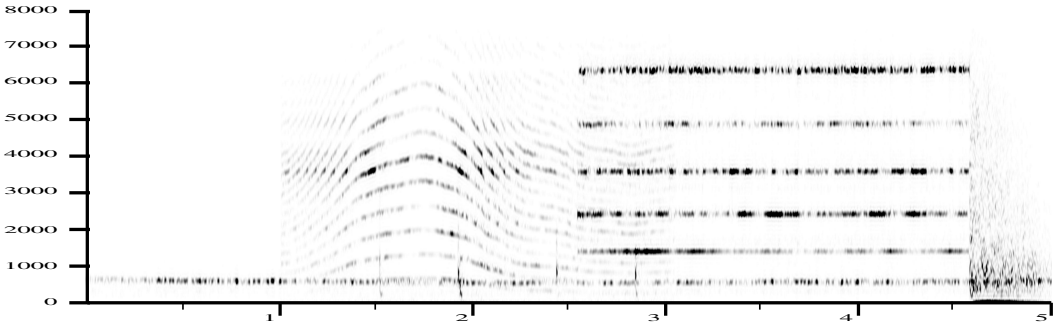
5.5 Summary

The experimental results of Section 5.2 demonstrate (1) that the IPUS architecture can be used to construct perceptual systems with reasonable interpretation performance in scenarios in real-world complex environments, and (2) that reprocessing and signal-processing theory can contribute to significant improvement in the interpretation of scenarios from complex environments. In support of the first claim, we note that the SUT discussed in Section 5.2 achieved a 59% hit rate and 67% tracking rate when interpreting the Phase II complex scenarios

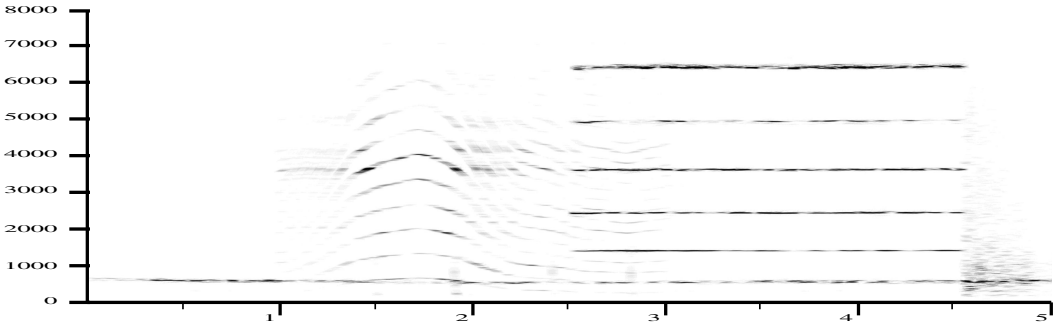
with access to a full sound library (Table 5.3). The second claim is supported by the observation that with a full library and complex scenarios, the SUT version with the reprocessing loop enabled outperformed the SUT version with the reprocessing loop disabled. Specifically, the reprocessing loop gained a 25% increase in hit rate and a 52% increase in tracking (Table 5.3).

The experimental results of Section 5.3 indicate the potential utility of using approximate front ends in perceptual systems to reduce computation for complex environments. With reduced cost in its front end, the approximate SUT version showed similar (though slightly lower) performance to that of the SUT version that used only non-approximate SPAs. Although more hypotheses were considered by the approximate SUT (Table 5.6) than by the "precise" SUT, many of the alternatives were removed from consideration after search for only 1 supporting microstream was performed (i.e. minimal extra work was done). The conclusion is tentative, however, because consistent timing of the interpretation search could not be characterized beyond a count of considered hypotheses. Clock-time measures for the highlevel interpretation work would be helpful in making the conclusion more definitive. However, they could not be determined reliably in the thesis experiments because the experiments were conducted on several computers with different load-sharing, and because the high-level interpretation KS's did not have optimal coding.

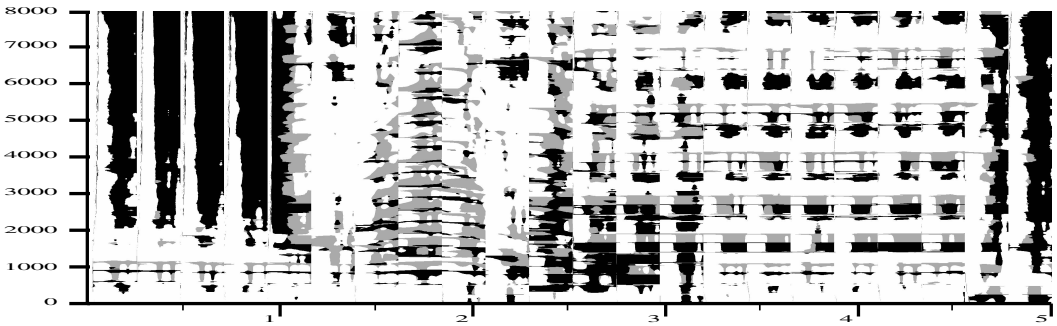
The results of Section 5.4 lend support to the claim that although it is possible for systems with fixed front ends to outperform IPUS in interpretation quality, such systems can require significantly greater front-end computational costs than IPUS-based systems and demand greater effort in constructing object models to account for all the detail their front ends are capable of producing.



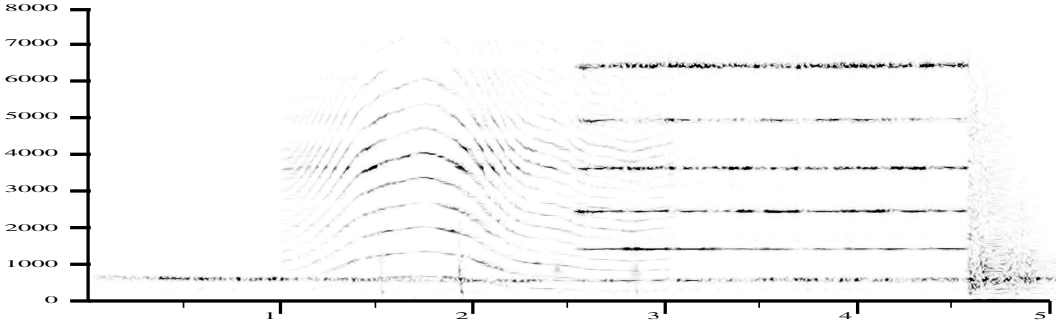
(a) 256-point window spectrum



(b) 1024-point window spectrum



(c) ATF window recommendations



(d) ATF spectrum

Figure 5.7. Comparison of ATF and STFT.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

This chapter's initial section first recapitulates the research issues and goals of the thesis and then summarizes the results of the experiments in Chapter 5 with respect to their support for the thesis contributions. The final section concludes with future research issues raised by the work.

6.1 Conclusions and Contributions

Complex environments can produce signal-to-noise ratios that vary unpredictably over time, can contain perceptual objects that mutually interfere with each others' signal signature, and can have perceptual objects that have arbitrary time-dependent behaviors. Within the traditional paradigm for designing signal understanding systems, the approach for handling the interactions created by these properties generally leads to exhaustive analysis of the environment and intricate tailoring of front ends to each possible combination of perceptual objects in the environment. While feasible for significantly constrained environments, this approach often adds a combinatorially explosive set of SPAs to a system's front end when used to build systems for interpreting complex environments.

This thesis makes two major arguments. The first argument is that the processing of signals from complex environments benefits from a new view of signal interpretation as the product of two interacting search processes, where the first search process involves the dynamic, context-dependent selection of signal features and interpretation hypotheses, and the second search process involves the dynamic, context-dependent selection of appropriate SPAs for extracting correlates to support the features. The second argument is that signal interpretation architectures can use explicit representation of the theoretical relationships between SPA

parameters and SPA outputs to effectively structure bidirectional interaction between these dual searches. (See Chapter 1.)

The IPUS architecture was designed to demonstrate the feasibility of the approach advocated in these arguments. In addition to showing how the basic components of the architecture's "reprocessing loop" (discrepancy detection, diagnosis, and reprocessing) provide conceptual hooks for organizing and applying signal processing theory to the dual-search interpretation paradigm, this thesis shows how the generic architecture exploits support machinery such as (1) explicit representation of uncertainty in interpretations and SPA outputs, and (2) processing-contexts and context-mapping to constrain both search processes and fuse evidence from multiple front ends. It also shows how the generic architecture can support effective application of specialized and approximate algorithms. (See Chapter 2.)

Most important to the evaluation of the architecture, however, this thesis shows through the instantiation of the Sound Understanding Testbed (See Chapter 4.) that IPUS can in fact be instantiated in a real-world problem domain (auditory scene analysis). The experimental evaluation results of Chapter 5 show support for the claims:

- the knowledge represented by the signal processing theory of a domain can play a significant role in improving the quality of interpretations of signals from complex environments. Significantly, the utility of the reprocessing loop increases with increasing scenario complexity. (See Section 5.2.)

- the IPUS architecture can effectively apply approximate processing techniques to trade off reductions in front-end and reprocessing complexity for moderate increases in interpretation search. (See Section 5.3.)

Though admittedly anecdotal and incomplete, the results of Section 5.4 indicate that although it is possible to produce systems with fixed front ends that can outperform IPUS in interpretation quality, such systems can require significantly greater front end computational costs than IPUS-based systems and demand greater effort in constructing object models to account for all the detail their front ends produce.

To summarize, this thesis makes the following research contributions:

- a generic architecture for designing perceptual systems for complex environments that represents a significant departure from conventional systems,
- the context-mapping and processing-context support mechanisms in Chapter 2 provides a framework for fusing correlates obtained from disparate front ends' analysis,
- through demonstrating the IPUS architecture's applicability in the real-world problem of auditory scene analysis in Chapter 5:
  - a demonstration of the role of reprocessing in improving the quality of interpretations, as shown by the relative performance improvements in Section 5.2.
  - a demonstration of the applicability and potential role of approximate processing techniques in IPUS, as shown by the tradeoffs between front-end complexity and interpretation search in Section 5.3.
- as will be seen in Section 6.2, a platform for future exploration of how to computationally approximate theories of auditory perception.

6.2 Future Research

The research program in this thesis points to several areas where new issues need to be explored. These issues arise from both the abstract interpretation approach developed in IPUS as well as the application domain studied in the validation of the architecture:

- **Preventative Front-End Adaptation.** The IPUS architecture has the ability to adapt front ends in anticipation of potential distortions among the signal signatures of objects. Such adaptation would have reduced the amount of reprocessing seen in the Phase 2 experiments of Chapter 5 by preventing the SUT from repeating preventable distortions along multi-block sounds. Since predicted distortions can quickly require a combinatorial set of conflicting adaptations (e.g. a need for better time resolution for sound A

requires a shorter STFT analysis window, while a need for better frequency resolution for co-occuring sounds B and C requires a longer window), however, a nontrivial framework for identifying the most probable and the most damaging subset of predictable distortions to be handled at the expense of others will be needed.

- **Memory Management.** The issue of memory management of reprocessed data and old data in the blackboard is important to long-term operation of IPUS-based systems. After a typical run on five seconds of data, the SUT's blackboard database often contained in excess of 7000 hypotheses, consuming 30 megabytes of memory. The IPUS reprocessing component currently saves all reprocessing results, and only the waveform level's contents are purged of data from 3 seconds before the current block's time. Clearly a continuously running interpretation system based on IPUS must eventually purge its blackboard database, or else available system memory would soon be exhausted. The issues to be explored include determining what levels of data should be considered for purging, how often the blackboard level should be purged, and how references made during interpretation or during reprocessing to purged data should be handled: should attempts be made to reconstruct purged evidence, or should the purged evidence be treated like any other missing evidence in IPUS and represented with `Support-Exclusion` SOUs?

- **Architectural Concerns.** The false-alarm results and their explanations from Section 5.2 indicate that either the IPUS architecture's reliance on an interpretation- sufficiency heuristic or the particular heuristic itself in the SUT instantiation might predispose a system to "hallucinate" in simple scenarios. Although the fixed-threshold nature of the heuristic makes it more likely that merely a more adaptive heuristic is needed, more work needs to be done on deciding this architectural question.

- **Sound Modelling.** The experiments of Chapter 5 show that the sound modelling framework does not adequately represent enough facets of acoustic environments, such

as wideband, longterm noisebeds. Work needs to be done on developing a compact, time-varying representation of the longterm wideband components of sounds that supports detection of situations where two or more sounds are superimposed.

- **Exploration of Acoustic Interpretation Strategies.** The combination of PSM SOU-selection heuristics and hypothesis SOU-selection heuristics defined in Chapter 4 represent only one of many possible control strategies that could be evaluated for the SUT. Furthermore, the decision to execute the complete reprocessing loop to confirm diagnoses is only one possible system behavior. More work needs to be done on finding cues for indicating how reliable the SUT diagnoses are, and on the utility of deferring actual reprocessing when highly reliable diagnoses are considered. Strategies where only part of the region covered by a hypothesis SOU is reprocessed also need to be explored, as they can lead to significant computational cost reductions for the reprocessing component.

APPENDIX A

THE SUT SOUND LIBRARY

The sound models in this library were derived from at least five instances for each sound; in the case of impulsive sounds the number of instances is often more. Each sound instance was captured in a signal stream at most 5 seconds long and sampled at 16 KHz. With the exception of the impulsive sounds "Knock," and "Clap," all sounds were extracted from a commercial tape provided by Auditec of St. Louis, Incorporated [Auditec 1989]. Note that relative energies for each sound's tracks are represented as ranges defining the highest and lowest steady-state simultaneous energy ratio measured between the a track and the selected baseline track for the source. Only if a sound was strictly a decaying one (e.g. a gong or chime) was the relative energy range determined from a non-steady region.

A.1 Alarm Clock 1

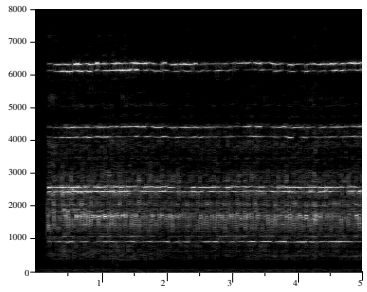


Figure A.1. Alarm Clock 1's Spectrogram.

Table A.1. Alarm Clock 1's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	6297—6367	[1.00, 1.00]	{64.9, 13.5}	{4.7, 1.4}
2	6093—6157	[0.05, 0.95]	{78.1, 17.8}	{4.4, 1.3}
3	4397—4439	[0.25, 0.80]	{68.3, 23.5}	{4.5, 1.1}
4	4093—4141	[0.15, 0.80]	{65.9, 14.3}	{4.6, 0.9}
5	2585—2602	[0.20, 0.80]	{81.1, 21.2}	{4.8, 1.1}
6	2453—2477	[0.80, 0.93]	{58.8, 9.31}	{5.7, 1.2}
7	929— 953	[0.04, 0.35]	{71.6, 11.9}	{11.0, 2.1}

notes: This is an analog, bell+ringer clock. There is a lot of variability in the strikes of the ringer on the bell, as seen in the amplitude entropies. The sound's range of durations is arbitrarily set to [1.5, 10.0] seconds in the IPUS library.

A.2 Alarm Clock 2

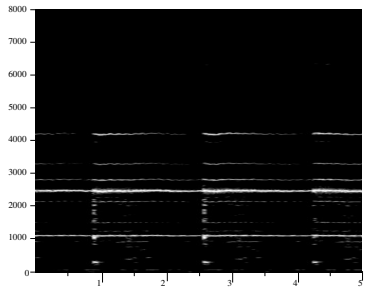


Figure A.2. Alarm Clock 2's Spectrogram.

Table A.2. Alarm Clock 2's Track Information.

Track	Freq. Range	Rel. Energy
1	2450—2510	[1.00, 1.00]
2	4190—4240	[0.22, 0.45]
3	2800—2850	[0.18, 0.32]
4	1110—1140	[0.09, 0.35]
5	3300—3330	[0.13, 0.20]
6	1510—1530	[0.01, 0.10]
7	3950—3990	[0.06, 0.09]

notes: This is an electronic alarm clock with each "ring" being a 1.6-second 7-track stream. Energies are higher at the start of the stream, and drop 50% (linearly) by the end of the stream. The sound's range of durations is arbitrarily set to [3.2, 10] seconds. In the SUT library this sound is represented as a script containing a sequence of rings abutting each other in time.

A.3 Bell Toll

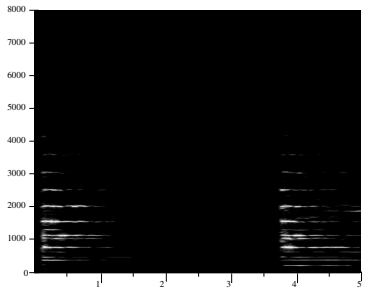


Figure A.3. Bell Toll's Spectrogram.

Table A.3. Bell Toll's Track Information.

Track	Freq. Range	Rel. Energy	Duration
1	765— 790	[0.05, 0.40]	[0.32, 0.45]
2	1120—1150	[0.06, 0.52]	[0.80, 0.94]
3	1555—1575	[1.00, 1.00]	[0.71, 0.80]
4	2010—2040	[0.03, 0.17]	[0.70, 0.83]
5	2500—2540	[0.05, 0.19]	[0.12, 0.22]

notes: Each toll stream is a harmonic set with $f_0 \approx 112$ Hz. The tracks listed are the most prominent harmonics. In order of decreasing energy, the harmonics represented here are 14, 10, 7, 18, and 22. Each toll is [0.8 1.0] seconds long, measured from strike time to when signal energy decays to the background energy level just prior to the strike. The *start* and *end* times for each track are relative to the start of a bell toll. Most of the time `Track 1` has the maximum energy of all tracks; however, around 0.2 seconds after the start of a toll `Track 2`'s energy grows to 5 times tk1's energy, and then decays similarly to all other tracks.

A.4 Bicycle Bell

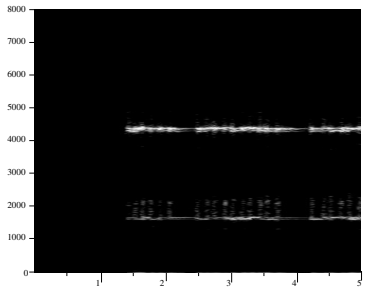


Figure A.4. Bicycle Bell's Spectrogram.

Table A.4. Bicycle Bell's Track Information.

Track	Freq. Range	Rel. Energy
1	4310—4390	[1.00, 1.00]
2	1640—1690	[0.06, 0.27]

notes: There are two phases in this source. The first phase, called the "active" phase, covers the period over which the bell is being struck, and can last an indeterminate period of time. From wideband analysis, each strike (vertical bar in the spectrogram) is 0.08 second long and is separated from the next strike by a minimum of 0.05 seconds. The second phase in the source is the "decay" phase, where the reverberations of the bell exponentially decay to background energy levels. This phase's length depends on the intensity of the last strike, but in the collected data this length has been found to be approximately 1.0 second.

A.5 Bugle

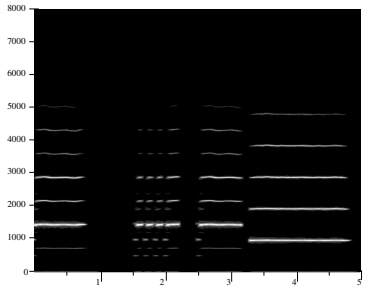


Figure A.5. Bugle's Spectrogram.

Table A.5. Bugle Note-1's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	700— 734	[0.01, 0.04]	{7.3, 0.3}	{8.9, 1.7}
2	1420—1453	[1.00, 1.00]	{18.1, 7.4}	{9.8, 0.5}
3	2148—2164	[0.05, 0.44]	{37.4, 13.7}	{5.2, 0.6}
4	2859—2875	[0.11, 0.57]	{82.2, 21.7}	{3.1, 0.2}

Table A.6. Bugle Note-2's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	929— 977	[1.00, 1.00]	{10.8, 2.2}	{3.4, 3.2}
2	1906—1930	[0.28, 0.61]	{11.3, 2.7}	{6.2, 0.5}
3	2867—2883	[0.04, 0.37]	{32.1, 23.6}	{3.1, 0.2}
4	3828—3844	[0.01, 0.19]	{32.0, 21.4}	{2.7, 0.5}

notes: These bugle notes are referred to as "coach horn" in the Auditec tape index. The figures are for the highest-energy tracks in the two longest notes in the sequence pictured above. Note 1 occurs in the 0.3–1.3 time range and again in the 2.5–3.15 time range in the spectrogram shown above. Note 2 occurs in the 3.3–5.0 time range. Other tracks shown

in the figure have only 1% of the energy of those for which data is given, and since they do not overlap other sounds' tracks, it was decided not to include them in the sound model.

A.6 Burglar Alarm

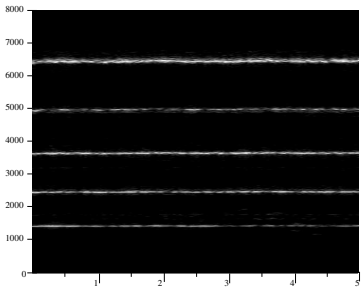


Figure A.6. Burglar Alarm's Spectrogram.

Table A.7. Burglar Alarm's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	6413—6468	[0.03, 0.40]	{49.9, 9.4}	{4.8, 0.9}
2	4875—5023	[0.04, 0.90]	{74.8, 15.0}	{7.1, 2.3}
3	3625—3665	[0.38, 0.77]	{67.6, 12.1}	{5.4, 1.1}
4	2445—2485	[0.42, 0.90]	{80.4, 12.8}	{5.2, 1.2}
5	1414—1446	[1.00, 1.00]	{69.3, 15.6}	{6.3, 3.0}

A.7 Car Engine

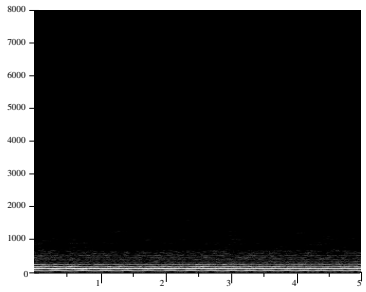


Figure A.7. Car Engine's Spectrogram.

Table A.8. Car Engine's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	249—283	[0.02, 0.09]	{30.8, 5.2}	{27.5, 10.1}
2	187—219	[0.04, 0.10]	{29.3, 5.0}	{46.8, 5.7}
3	117—140	[1.00, 1.00]	{7.9, 2.7}	{3.2, 3.8}
4	61— 70	[0.10, 0.40]	{20.4, 3.3}	{154.1, 3.2}

notes: This is the sound of a car's engine from inside the car.

A.8 Car Horn

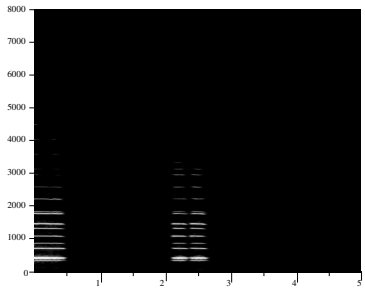


Figure A.8. Car Horn's Spectrogram.

Table A.9. Car Horn's Track Information.

Track	Freq. Range	Rel. Energy
1	445— 460 (f_0^1)	[1.00, 1.00]
2	1780—1805 ($4f_0^1$)	[0.10, 0.35]
3	735— 750 ($2f_0^2$)	[0.02, 0.10]
4	1100—1125 ($3f_0^2$)	[0.02, 0.10]
5	360— 382 (f_0^2)	[0.08, 0.25]
6	1470—1500 ($4f_0^2$)	[0.08, 0.16]

notes: The observed beep durations fall in the range [0.2, 0.50] seconds. The source has approximately 20 significant tracks that come from 2 harmonic sets. The first set has $f_0^1 \approx 450$ Hz, and contributes its first and fourth harmonics to the source. The second set has $f_0^2 \approx 370$ Hz, and contributes its first 4 harmonics to the source.

A.9 Chicken

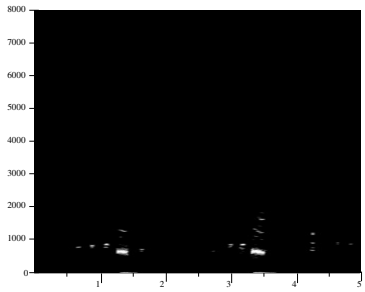


Figure A.9. Chicken's Spectrogram.

Table A.10. Chicken's Track Information.

Track	Freq. Range	Rel. Energy
1	chirp 670— 610	[1.00, 1.00]
2	chirp 1350—1220	[0.10, 0.30]

notes: The five chicken clucks used to generate this model have durations lying in the range [0.13, 0.20] seconds.

A.10 Chime

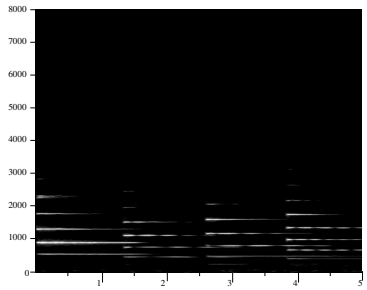


Figure A.10. Chime's Spectrogram.

Table A.11. Chime's Track Information.

Track	Freq. Range	Energy	Duration
1	560— 590	[0.01, 0.12]	[1.4, 1.6]
2	900— 935	[1.00, 1.00]	[1.4, 1.6]
3	1300—1340	[0.05, 0.53]	[0.9, 1.0]
4	1780—1820	[0.01, 0.17]	[0.5, 0.6]
5	2275—2324	[0.01, 0.24]	[0.2, 0.25]

notes: Each chime note is the result of a single strike followed by [1.2 2.5] seconds of reverberation. The tracks shown in the table are for the first note.

A.11 Clap

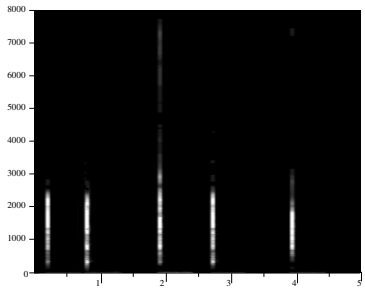


Figure A.11. Clap's Spectrogram.

noisebed 1: mean: [2.1933, 0.4440, 0.2067, 0.2268]

cov:
$$\begin{bmatrix} 7.5131s-2 & -3.4630s-3 & -1.0287s-2 & -6.3474s-3 \\ -3.4630s-3 & 8.9997s-4 & 4.2013s-4 & 3.6960s-4 \\ -1.0287s-2 & 4.2013s-4 & 4.5016s-3 & 2.7282s-3 \\ -6.3474s-3 & 3.6960s-4 & 2.7282s-3 & 2.0249s-3 \end{bmatrix}$$

notes: The feature values are listed in the order they are described at the beginning of this section. Ten isolated claps were used to generate these feature values.

A.12 Clock Chime

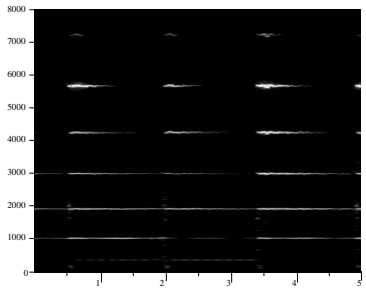


Figure A.12. Clock Chime's Spectrogram.

Table A.12. Clock Chime's Track Information.

Track	Freq. Range	Rel. Energy	Duration
1	5650—5685	[1.00, 1.00]	[0.5, 0.55]
2	1920—1950	[0.03, 0.31]	[1.9, 2.0]
3	1040—1065	[0.03, 0.22]	[1.4, 1.5]
4	2290—3025	[0.05, 0.29]	[0.9, 1.0]
5	4240—4280	[0.08, 0.25]	[0.4, 0.5]

notes: Each isolated chime's reverberations last approximately 2.0 seconds, with `Track 2` lasting the longest of the harmonic set. The track with greatest energy is `Track 1`; however, the other tracks last longer and have slight rises in energy as `Track 1` decays.

A.13 Clock Ticks

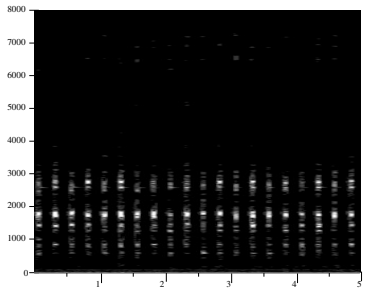


Figure A.13. Clock Tick's Spectrogram.

track 1: 1730—1770 Hz

noisebed 1: mean: [2.3224, 0.3689, 0.1056, 0.1299]

cov:
$$\begin{bmatrix} 3.3037s-2 & -1.5382s-3 & -4.7619s-3 & -2.5159s-3 \\ -1.5382s-3 & 2.1002s-3 & -3.6221s-5 & 2.5113s-4 \\ -4.7619s-3 & -3.6221s-5 & 3.0816s-3 & 2.8253s-3 \\ -2.5159s-3 & 2.5113s-4 & 2.8253s-3 & 3.1981s-3 \end{bmatrix}$$

notes: Each tick is 0.1 seconds long (practically no variation), and there is 0.15 seconds between ticks. There is a significant noisebed throughout the narrow time-slice that each tick lasts. However, over all examined ticks, there was a consistent peak at the "track" frequencies noted above. Twenty clock ticks were used to generate the noisebed feature values.

A.14 Cuckoo Clock (Cuckoo + Hour Chime)

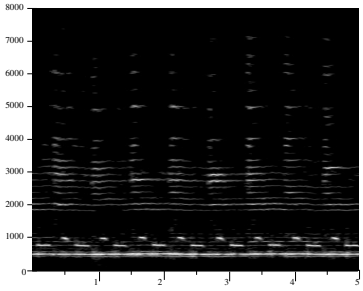


Figure A.14. Cuckoo Clock's Spectrogram.

Table A.13. Cuckoo Clock's Track Information.

Track	Freq. Range	Rel. Energy
1	960—1020	[0.05, 0.35]
2	750— 800	[0.25, 0.75]
3	495— 525	[1.00, 1.00]
4	2000—2050	[0.02, 0.13]
5	2740—2790	[0.01, 0.19]

notes: This sound is actually the result of two simultaneous sources. `Track 1` and `Track 2` are produced by the cuckoo-call, while `Track 3`, `Track 4`, and `Track 5` are produced by a clock-chime. Each "cuckoo-chime" combination lasts [0.56, 0.63] seconds (i.e. time between successive strikes to the clock-chime). The final chime's reverberations last 0.3 seconds beyond the 0.6-second chime-length. This model was developed from 12 "cuckoo-chime" instances.

A.15 Doorbell Chime

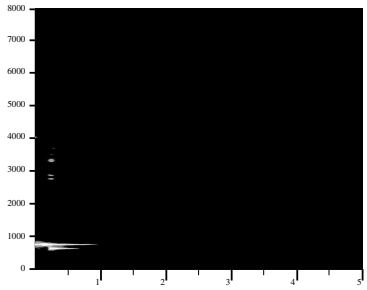


Figure A.15. Doorbell Chime's Spectrogram.

Table A.14. Doorbell Chime's Track Information.

Track	Freq. Range	Rel. Energy	Start Time
1	740— 767	[1.00, 1.00]	[0.00, 0.00]
2	615— 649	[0.21, 0.90]	[0.20, 0.25]

notes: The short "tracks" above `Track 1`'s frequency have extremely low energy and appear prominent only because of the image-enhancement process used to generate the spectrogram shown here. They are not included in the sound model.

A.16 Door Creak

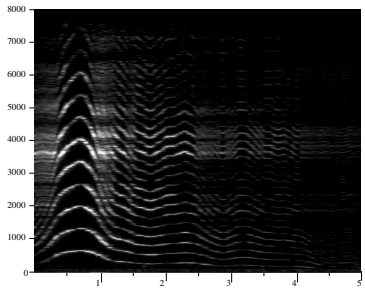


Figure A.16. Door Creak's Spectrogram.

Table A.15. *Door Creak's Track Information.*

Track	Freq. Range	Rel. Energy
1	300— 680— 330	[0.80, 1.20]
2	625—1350— 670	[0.80, 1.20]
3	870—2030—1000	[0.80, 1.20]
4	1100—2600—1400	[0.80, 1.20]
5	1400—3375—1800	[0.80, 1.20]
6	1600—4050—2000	[1.00, 1.00]

notes: All attack lengths fall in the range [0.3, 0.4] seconds, while all decay lengths fall into the range [0.3, 0.4] seconds. The frequencies shown are derived from the "knee" in the first 1.5 seconds of spectrogram data.

A.17 Firengine Bell

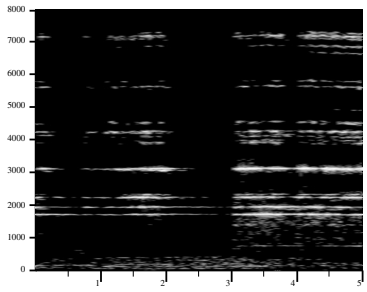


Figure A.17. Firengine Bell's Spectrogram.

Table A.16. Firengine Bell's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	3070—3149	[0.50, 1.00]	{80.1, 22.9}	{5.0, 1.5}
2	2242—2266	[0.06, 0.53]	{80.7, 16.3}	{6.4, 1.2}
3	1945—1969	[0.30, 0.82]	{84.9, 25.6}	{5.7, 0.8}
4	1719—1750	[1.00, 1.00]	{72.2, 23.1}	{6.6, 1.8}
5	3304—3329	[0.02, 0.33]	{92.0, 14.4}	{6.5, 0.6}
6	4210—4235	[0.01, 0.20]	{84.9, 18.3}	{4.9, 0.5}

notes: This sound is produced by a bell being struck. `Track 1` is the most energetic track. The source has no natural duration bounds; for the purposes of this database the bounds on active duration are set to be [3.0, 6.0] seconds. The reverberation decay ranges from 0.5 to 1.0 seconds.

A.18 Firehouse Alarm

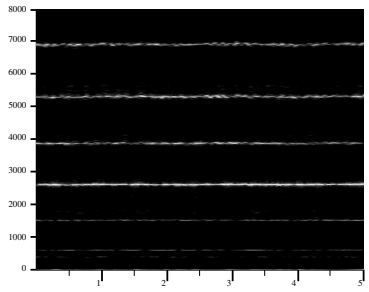


Figure A.18. Firehouse Alarm's Spectrogram.

Table A.17. Firehouse Alarm's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	601— 618	[0.07, 0.49]	{78.4, 12.5}	{15.8, 2.8}
2	1515—1539	[0.08, 0.52]	{65.6, 12.5}	{11.7, 2.4}
3	2617—2633	[1.00, 1.00]	{75.7, 15.9}	{5.1, 0.8}
4	3882—3907	[0.37, 0.90]	{78.7, 17.8}	{4.5, 0.9}
5	5312—5352	[0.20, 0.73]	{65.6, 13.3}	{4.8, 1.0}
6	6914—6954	[0.18, 0.82]	{75.2, 23.3}	{4.0, 1.1}

notes: The source's tracks have too much variability in energy to determine useful rel. energy ratios. `Track 1` generally is the lowest-energy track. The tracks form the following harmonics from a harmonic set with f_0 =[186, 188] Hz: 3, 8, 14, 21, 28, and 37. The sound's range of durations is nominally set to [3.0, 10.0] seconds.

A.19 Foghorn

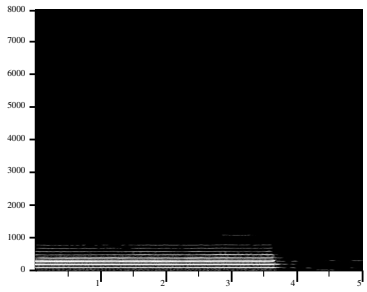


Figure A.19. Foghorn's Spectrogram.

Table A.18. Foghorn's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	257— 336	[1.00, 1.00]	{12.4, 6.7}	{9.7, 13.2}
2	171— 235	[0.50, 0.60]	{13.6, 7.5}	{5.3, 12}
3	492— 516	[0.10, 0.30]	{32.1, 9.4}	{14.3, 5.5}
4	593— 610	[0.20, 0.40]	{32.8, 8.1}	{14.9, 2.4}
5	687— 711	[0.03, 1.00]	{46.2, 24.7}	{13.8, 1.6}
6	789— 805	[0.03, 1.00]	{32.9, 6.2}	{11.9, 2.0}

notes: Each horn blast lasts 3.8 seconds. A horn blast is a harmonic stream with $f_0 = 97$ Hz. The energy ratios are quite stable; there is a slow rise in energy in `Track 5` and `Track 6` as the horn finishes sounding, which accounts for their wide energy ranges. The tracks indicated here represent the following harmonics: 2, 3, 5, 6, 7, and 8.

A.20 Footsteps

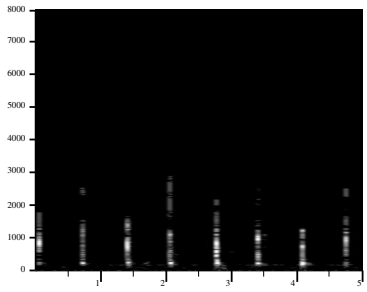


Figure A.20. Footstep's Spectrogram.

noisebed 1: mean: [1.2753, 0.2762, 0.1619, 0.2129]

cov:
$$\begin{bmatrix} 4.5466\times10^{-2} & -6.0995\times10^{-3} & -1.0380\times10^{-2} & -3.3756\times10^{-3} \\ -6.0995\times10^{-3} & 5.1067\times10^{-3} & 2.6788\times10^{-3} & 1.0639\times10^{-3} \\ -1.0380\times10^{-2} & 2.6788\times10^{-3} & 6.1067\times10^{-3} & 4.5021\times10^{-3} \\ -3.3756\times10^{-3} & 1.0639\times10^{-3} & 4.5021\times10^{-3} & 6.1556\times10^{-3} \end{bmatrix}$$

notes: Each footstep is at most 0.3 seconds in duration. In the IPUS sound library we use the source model "footfall" to represent a single footstep. The library uses the acoustic script "footsteps" to represent the concept of a sequence of footfalls. Eight footfalls were used to generate the noisebed feature values.

A.21 Glass Clink

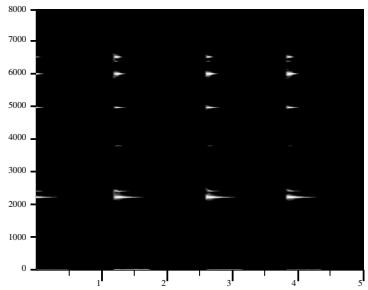


Figure A.21. Glass Clink's Spectrogram.

Table A.19. Glass Clink's Track Information.

Track	Freq. Range	Rel. Energy
1	2225—2243	[1.0, 1.0]
2	4960—4990	[0.5, 1.0]

noisebed 1: mean: [4.9140, 0.06986, 0.03265, 0.02894]

cov:
$$\begin{bmatrix} 1.7730\times10^{-1} & 9.7419\times10^{-3} & 4.6897\times10^{-4} & -1.2236\times10^{-3} \\ 9.7419\times10^{-3} & 7.9156\times10^{-4} & -2.0913\times10^{-5} & -1.1002\times10^{-4} \\ 4.6897\times10^{-4} & -2.0913\times10^{-5} & 3.1226\times10^{-5} & 1.8969\times10^{-5} \\ -1.2236\times10^{-3} & -1.1002\times10^{-4} & 1.8969\times10^{-5} & 3.0866\times10^{-5} \end{bmatrix}$$

notes: Each glass-clink is [0.15, 0.2] seconds in duration. Ten isolated glass clinks were used to generate the feature values. The spectrogram tracks not listed had low and highly variable energies.

A.22 Gong

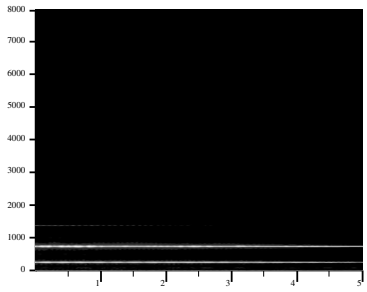


Figure A.22. Gong's Spectrogram.

Table A.20. Gong's Track Information.

Track	Freq. Range	Rel. Energy	Duration
1	743— 757	[1.00, 1.00]	[6.30, 7.00]
2	251— 273	[0.30, 0.40]	[6.30, 7.00]
3	1375—1390	[0.002, 0.01]	[1.20, 1.43]

notes: The majority of the sound's spectral energy is in the first two tracks listed. However, `Track 3` is included in the model because its energy is strong enough to interfere with the energy measurement of other sounds' tracks that overlap its frequency region. The sound's duration range is [6.3, 7.1] seconds.

A.23 Hairdryer

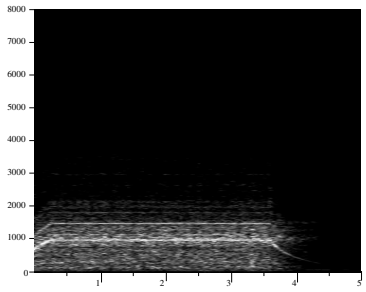


Figure A.23. Hairdryer's Spectrogram.

Table A.21. Hairdryer's Track Information.

Track	Freq. Range	Rel. Energy
1	980—1015	[1.00, 1.00]
2	1475—1510	[0.33, 0.54]
3	1810—1840	[0.05, 0.20]

notes: The hairdryer displays distinct phases of activity corresponding to it being turned-on, running and being turned-off. They are known as the attack or onset, steady and decay phases. Onset behavior is visible in `Track 1`, where the frequency rises from 210 Hz to 1015 Hz over a period of 0.5 seconds. `Track 1` similarly displays decay behavior, characterized by a slowly decaying exponential, starting at 1015 Hz and ending at 210 Hz. In the IPUS database this sound's *overall* range of durations (encompassing attack, steady, and decay behaviors) is set to [3.0, 10.0] seconds.

A.24 Knock

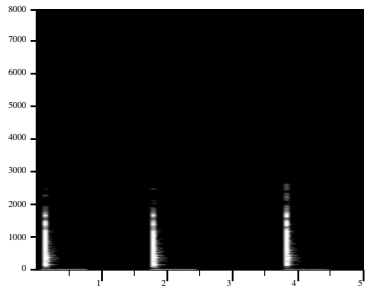


Figure A.24. Knock's Spectrogram.

noisebed 1: mean: [1.2257, 0.2551, 0.2364, 0.3248]

cov:
$$\begin{bmatrix} 1.1641\times10^{-3} & -5.0455\times10^{-5} & -1.0276\times10^{-3} & -6.2427\times10^{-4} \\ -5.0455\times10^{-5} & 2.8732\times10^{-4} & 5.8748\times10^{-6} & -4.4741\times10^{-5} \\ -1.0276\times10^{-3} & 5.8748\times10^{-6} & 9.3234\times10^{-4} & 5.8904\times10^{-4} \\ -6.2427\times10^{-4} & -4.4741\times10^{-5} & 5.8904\times10^{-4} & 4.0269\times10^{-4} \end{bmatrix}$$

notes: Ten instances of the knock sound were used to generate the noisebed feature values.

A.25 Oven Buzzer

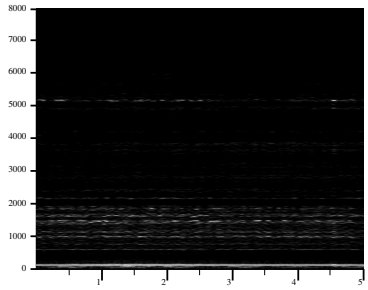


Figure A.25. Oven Buzzer's Spectrogram.

Table A.22. Oven Buzzer's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	116— 133	[1.00, 1.00]	{9.0, 3.3}	{0.1, 0.1}
2	593— 602	[0.02, 0.09]	{44.5, 11.8}	{10.5, 3.4}
3	1000—1016	[0.02, 0.06]	{55.8, 11.4}	{14.8, 1.9}

notes: This sound's expected duration range in the SUT library is arbitrarily set to [3.0, 10.0].

A.26 Owl

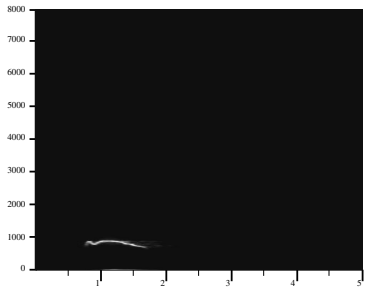


Figure A.26. Owl's Spectrogram.

Table A.23. Owl Track Information.

Track	Freq. Range	Rel. Energy
1	780— 880	[1.00, 1.00]

A.27 Pistol Shot

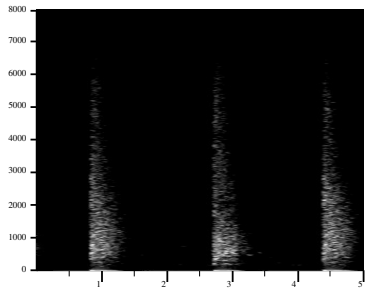


Figure A.27. Pistol Shot's Spectrogram.

noisebed 1: mean: [1.6869, 0.5350, 0.04049, 0.04070]

cov:
$$\begin{bmatrix} 3.7361\times10^{-2} & 3.0124\times10^{-3} & -1.0742\times10^{-3} & -3.0123\times10^{-3} \\ 3.0124\times10^{-3} & 1.6050\times10^{-1} & 3.7407\times10^{-3} & 1.2623\times10^{-3} \\ -1.0742\times10^{-3} & 3.7407\times10^{-3} & 1.6393\times10^{-4} & 1.3829\times10^{-4} \\ -3.0123\times10^{-3} & 1.2623\times10^{-3} & 1.3829\times10^{-4} & 2.8300\times10^{-4} \end{bmatrix}$$

notes: This sound has extreme wide-band energy, with each shot stream including a frequency decay of approximately 0.3 seconds. Five isolated pistol-shot instances were used to generate the noisebed feature values shown here.

A.28 Policecar Siren

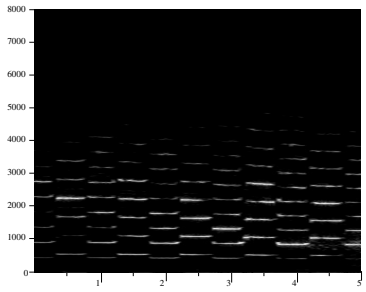


Figure A.28. Policecar Siren's Spectrogram.

Table A.24. Policecar Siren Note-1's Track Information.

Track	Freq. Range	Rel. Energy
1	555— 570	[0.05, 0.33]
2	1110—1150	[0.001, 0.03]
3	1674—1710	[0.07, 0.35]
4	2250—2281	[1.00, 1.00]
5	2813—2851	[0.08, 0.30]
6	3375—3414	[0.06, 0.32]

Table A.25. Policecar Siren Note-2's Track Information.

Track	Freq. Range	Rel. Energy
1	446— 468	[0.11, 0.34]
2	907— 947	[0.78, 1.10]
3	1360—1406	[0.10, 0.32]
4	1821—1843	[1.00, 1.00]
5	2274—2304	[0.25, 0.60]
6	3180—3210	[0.19, 0.57]
7	3634—3680	[0.07, 0.61]

notes: This sound has two alternating streams (notes). One is defined by a harmonic set with $f_0 \approx 460$, and the other is defined by a harmonic set with $f_0 \approx 560$. Both notes have durations of [0.45 0.50] seconds. Each note is represented as a separate sound model in the SUT library, and a script defines the repetitive behavior of the overall sound.

A.29 Razor

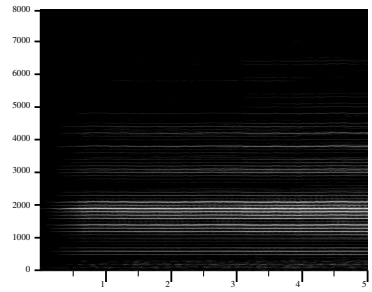


Figure A.29. Razor's Spectrogram.

Table A.26. Razor's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	1890—1910	[1.00, 1.00]	{27.0, 20.3	{4.8, 0.6}
2	2090—2110	[0.60, 0.64	{35.6, 21.9}	{2.3, 1.8}
3	1790—1810	[0.56, 0.60]	{30.8, 21.2}	{5.1, 0.4}
4	1690—1710	[0.42, 0.46]	{32.4, 17.6}	{4.3, 1.1}
5	1590—1610	[0.38, 0.42]	{32.0, 20.1}	{2.2, 2.2}
6	1390—1410	[0.38, 0.42]	{22.6, 19.3}	{5.6, 1.8}

notes: The attack and decay durations are [0.5, 0.6] seconds long. The signal is best described as a harmonic set with $f_0 = 100$ Hz. The signal has *very* steady relative energy ratios among its tracks. However, wideband (short-time) analysis reveals that the tracks all have sinusoidal amplitude modulation. The range of durations for the overall length of a razor sound is arbitrarily set to [5.0, 11.0] seconds in the SUT library.

A.30 Smoke Alarm 1

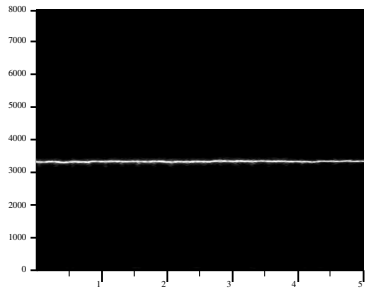


Figure A.30. Smoke Alarm-1's Spectrogram.

Table A.27. Smoke Alarm-1's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	3336—3352	[1.00, 1.00]	{39.6, 23.7}	{3.5, 1.2}
2	6660—6710	[0.003, 0.01]	—	—

notes: `Track 1`'s narrow peak widens in sideband energy every 0.25 seconds. At the same time, `Track 2`'s energy dips to a local minimum. This sound's duration range is nominally set to [3.0, 10.0] seconds. No entropy ranges are included for `Track 2` because its low energy and high variability led to entropy ranges that were so wide as to be useless.

A.31 Smoke Alarm 2

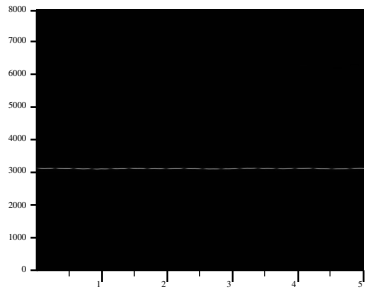


Figure A.31. Smoke Alarm-2's Spectrogram.

Table A.28. Smoke Alarm-2's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	3125—3133	[1.00, 1.00]	{58.9, 6.4}	{2.3, 0.8}
2	6227—6280	[0.02, 0.20]	{51.2, 16.1}	{4.5, 1.1}

notes: According to short-time analysis, every 0.7 seconds, the total signal energy drops. At these times, `Track 1`'s energy drops by a factor of 10, while `Track 2`'s energy drops by a factor of 2. The high relative energy of `Track 2` occurs at this point. The low-energy dip lasts for 0.05 seconds. This sound's duration range is nominally set to [3.0, 10.0] seconds.

A.32 Telephone Dial

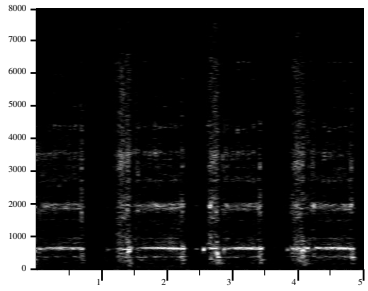


Figure A.32. Telephone Dial's Spectrogram.

Table A.29. Telephone Dial's Track Information.

Track	Freq. Range	Rel. Energy
1	630— 690	[1.00, 1.00]
2	1930—2000	[0.02, 0.38]

noisebed 1: start-time = [0.25, 0.3]

mean: [1.5481, 0.2088, 0.06939, 0.07352]

covariance:
$$\begin{bmatrix} 1.9684\times10^{-2} & 1.2455\times10^{-2} & -1.0637\times10^{-3} & -1.0697\times10^{-3} \\ 1.2455\times10^{-2} & 2.9754\times10^{-2} & -9.3511\times10^{-4} & -9.5440\times10^{-4} \\ -1.0637\times10^{-3} & -9.3511\times10^{-4} & 2.2955\times10^{-4} & 2.4747\times10^{-4} \\ -1.0697\times10^{-3} & -9.5440\times10^{-4} & 2.4747\times10^{-4} & 2.6835\times10^{-4} \end{bmatrix}$$

noisebed 2: start-time = 0.1 seconds before the end of the dial,[0.25, 0.3]

mean: [1.2605, 0.2114, 0.08285, 0.08847]

covariance:
$$\begin{bmatrix} 3.3743\times10^{-2} & -6.1750\times10^{-4} & -1.5244\times10^{-3} & -1.0827\times10^{-3} \\ -6.1750\times10^{-4} & 1.0062\times10^{-3} & 9.8642\times10^{-5} & 7.7836\times10^{-5} \\ -1.5244\times10^{-3} & 9.8642\times10^{-5} & 2.1811\times10^{-4} & 2.0461\times10^{-4} \\ -1.0827\times10^{-3} & 7.7836\times10^{-5} & 2.0461\times10^{-4} & 1.9815\times10^{-4} \end{bmatrix}$$

notes: This is the sound of a rotary phone being dialed. The sound has two impulses: the first representing the impact of the finger and the stopper at the end of the clockwise

dial rotation, and the second at the end of the sound, representing the end of the counterclockwise dial rotation. The dial sounds range in duration from 0.5 seconds for dials from the digit 1 to 2.5 seconds for dials from the digit 0.

A.33 Telephone Ring

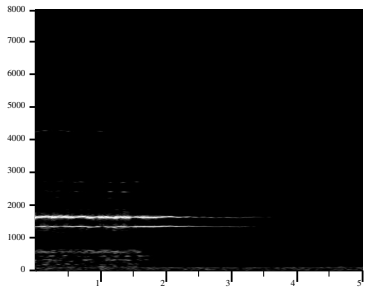


Figure A.33. Telephone Ring's Spectrogram.

Table A.30. Telephone Ring's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	585— 649	[0.005 0.04]	{44.9, 7.5}	{89.5, 6.7}
2	1351—1367	[0.04 0.26]	{50.1, 13.0}	{8.7, 2.1}
3	1633—1656	[1.00 1.00]	{76.5, 15.6}	{8.2, 1.5}

notes: For each ring, the striker actively hits the bell for 1.7 seconds; reverberations last 3.3 seconds after that. `Track 1` is the highest-energy track. `Track 3` has energy diffused throughout its range.

A.34 Telephone Tone

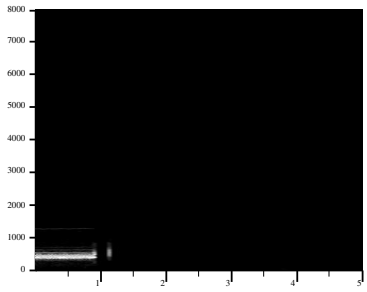


Figure A.34. Telephone Tone's Spectrogram.

Table A.31. Telephone Tone's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	391— 414	[1.00 1.00]	{13.6, 0.8}	{17.4, 4.7}
2	469— 492	[0.19 0.42]	{13.1, 1.1}	{16.7, 4.6}
3	1270—1290	[0.01 0.30]		

notes: The sound's range of durations is [0.8, 2.0] seconds in the SUT library.

A.35 Triangle

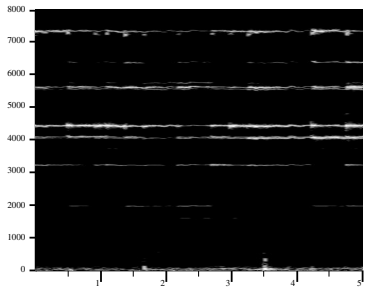


Figure A.35. Triangle's Spectrogram.

Table A.32. Triangle's Track Information.

Track	Freq. Range	Rel. Energy
1	4415—4440	[1.00, 1.00]
2	7297—7320	[0.10, 0.20]
3	4047—4085	[0.20, 0.40]
4	5571—5609	[0.15, 0.40]

notes: The above model was generated from five isolated triangle-strike instances. The durations all fall in the range [1.0,1.2] seconds.

A.36 Truck Motor

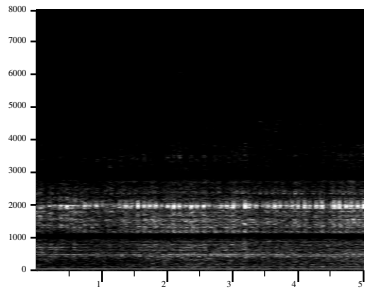


Figure A.36. Truck Motor's Spectrogram.

Table A.33. Truck Motor's Track Information.

Track	Freq. Range	Rel. Energy
1	1970—2010	[1.00, 1.00]
2	475— 507	[0.03, 0.26]

notes: `Track 1` is really a series of discontinuous frequency spikes. Wideband analysis shows that each burst on the spectrogram shown here is a double spike with 0.02 seconds between each spike in the pair, and with each spike lasting approximately 0.01 seconds. The distance between the last spike of one pair and the first spike of the next is 0.05 seconds. The three time values yield a pair period of approximately 0.1 seconds. The sound's range of durations is arbitrarily set to [5.0, 30.0] seconds.

A.37 Vending Machine Hum

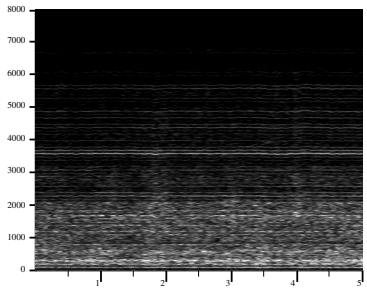


Figure A.37. Vending Machine Hum's Spectrogram.

Table A.34. Vending Machine Hum's Track Information

Track	Freq. Range	Rel. Energy
1	282— 328	[1.00 1.00]
2	1672—1728	[0.29 0.83]
3	3560—3601	[0.05 0.32]

notes: Although the narrowband tracks are well-defined, note that this source has a significant noisebed from 0 to 8000 Hz. The sound's range of durations is arbitrarily set to [5.0, 30.0] seconds.

A.38 Viola

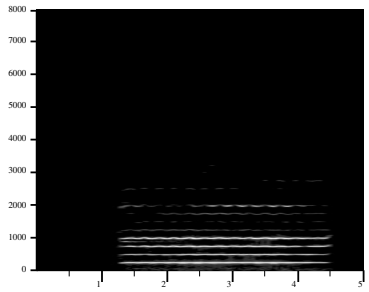


Figure A.38. Viola's Spectrogram.

Table A.35. Viola's Track Information.

Track	Freq. Range	Rel. Energy	Ampl. Entropy	Freq. Entropy
1	242— 275	[0.78 1.10]	{25.9, 3.9}	{14.6, 12.0}
2	491— 525	[0.70 1.06]	{52.9, 14.8}	{22.5, 3.5}
3	749— 783	[1.00 1.00]	{51.0, 6.7}	{18.5, 2.5}
4	991—1033	[0.61 0.90]	{26.6, 7.7}	{19.3, 1.4}

notes: The signal has the harmonic set $f_0 \approx 260$ Hz. Its duration range in the IPUS database is [2.4, 3.0] seconds.

A.39 Wind

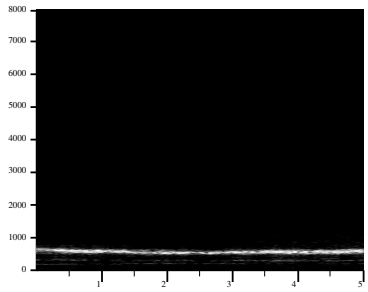


Figure A.39. Wind's Spectrogram.

Table A.36. Wind's Track Information.

Track	Freq. Range	Ampl. Entropy	Freq. Entropy
1	625— 766	{71.8, 14.6}	{46.9, 12.6}

notes: The spectral energy in `Track 1` meanders throughout the track's frequency region. The range of durations for the overall length of this sound is arbitrarily set to [5.0, 30.0] seconds in the IPUS database.

APPENDIX B

SUT TRACE

To provide a concrete example of how the IPUS SUT behaves when processing data, this appendix presents a trace of the SUT during one of its experiment runs reported in Chapter 5. The run's scenario is shown in Figure B.1, and contains three types of sounds: two notes of a policecar siren and a firengine bell. For this run the testbed's sound library contained all 40 sounds reported in Appendix A, and the testbed's default front end was the one used in Suite 1 (see Section 5.2).

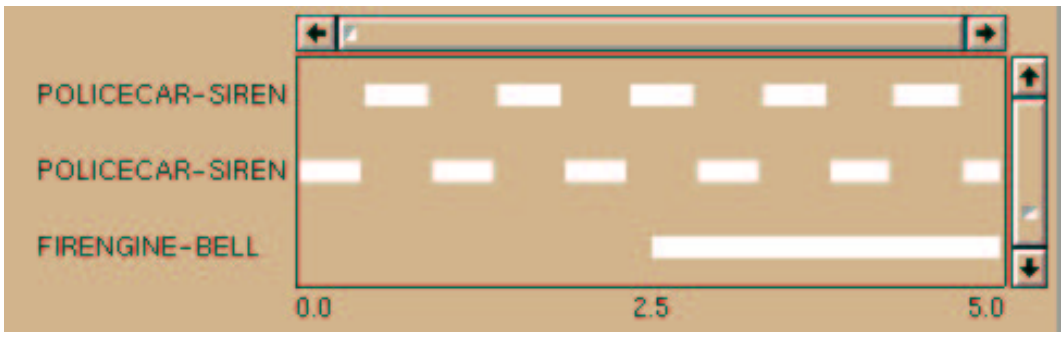


Figure B.1. Trace Scenario.

The major distortion processes that are expected to require reprocessing in the scenario are (1) time-resolution problems in determining the endpoints of the policecar siren notes and the firengine bell's start-time, (2) peak-thresholding problems in picking enough peaks to track all the sounds' microstreams, and (3) frequency-resolution problems in separating the microstreams of the firengine bell and the policecar siren notes. The trace shows that for some of the low-energy tracks of POLICECAR-SIREN-STREAM1, an energy-thresholding problem was also assumed by the SUT. Note that diagnosis and reprocessing are local with respect to

the individual notes in the policecar siren script; no similarities across script components are exploited in generating diagnoses or in searching for support.

The following conventions were adopted in selecting what details to show and in formatting the trace output. The trace only reports

1. decisions about what sound hypotheses are selected for consideration on the basis of observed spectral bands,

2. initial attempts at searching for contour support for sounds' microstream hypotheses, and their results, including discrepancies

3. violation discrepancies involving detection of time-domain events at block edges.

4. attempts at diagnosing missing evidence,

5. reprocessing efforts at uncovering missing evidence, and

6. when sound hypotheses are disbelieved.

The start of each of these events is labelled with four asterisks. The OBJECT message in the diagnosis alerts indicates the microstream whose negative-evidence SOU is being diagnosed, and the REGION message indicates where in time and frequency the SOU exists.

The table at the end of the trace shows all answer and nonanswer hypotheses, their rating, their negative evidence rating, their proposed time-periods of existence, and the number of processing contexts that were used to support or disconfirm them.

To aid in following the trace, a screen dump of the SUT's status report window is shown at the end of each data block's analysis. As shown in Figure B.2, the panes on the window show blackboard hypotheses for each testbed abstraction level except the waveform, spectrogram, and script levels. For most time-frequency items such as peaks and microstreams, the display representation should be self-explanatory, although note that for peaks and spectral bands lighter object colors indicate greater spectral energy. On the Time-Domain Events pane, boundary events appear as stylized steps indicating a step-up or step-down in waveform envelope energy,

while impulses appear as “lollipops” indicating spikes in waveform envelope energy. Between block traces, the updated window will show the previously-analyzed waveform envelope, plus the envelope of the next waveform block to be analyzed.

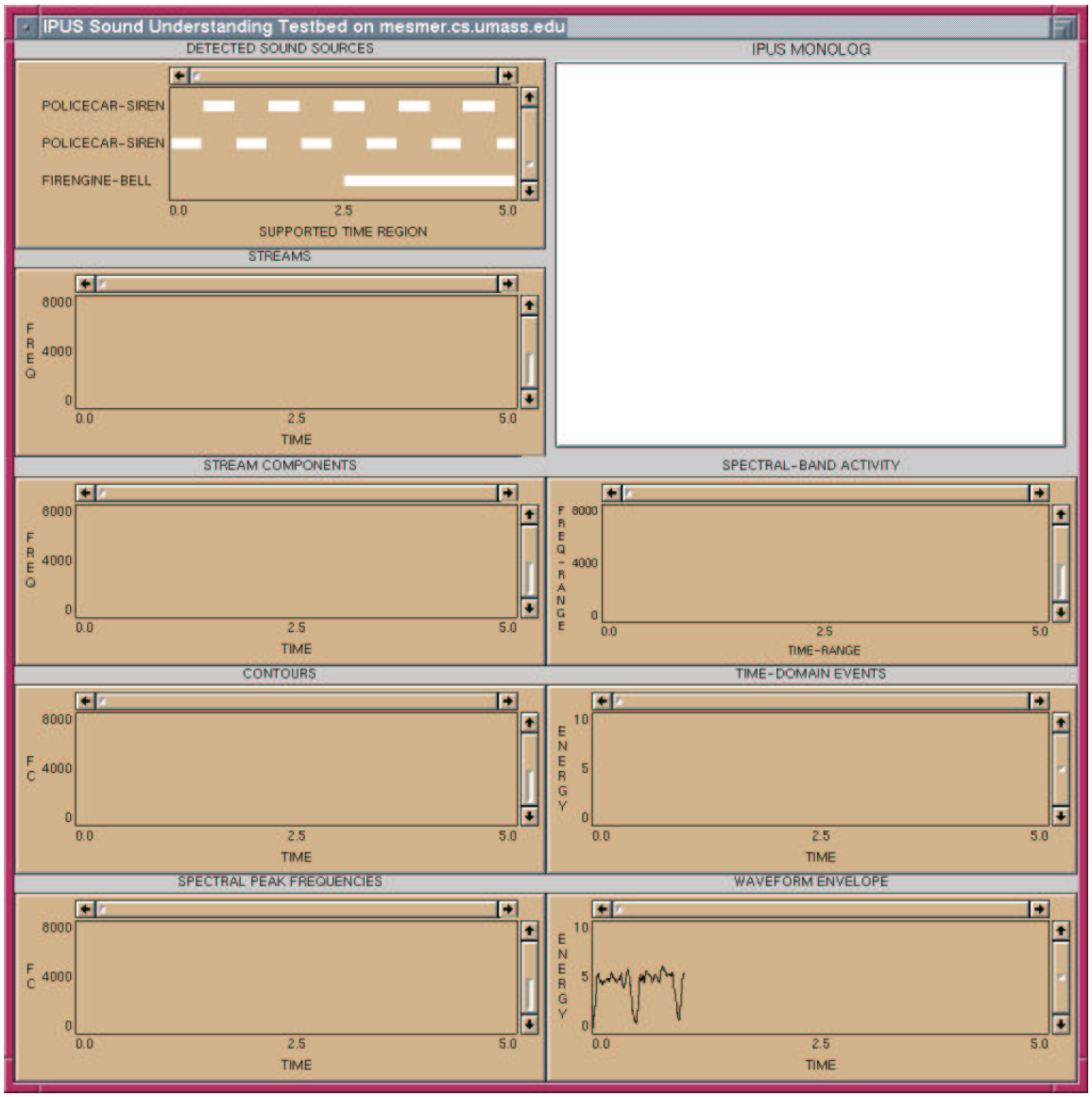


Figure B.2. Trace Start.

```
Started Scenario 2 at 30 Jun 1996 12:44:00

**** Processing of [0.000 1.000} time period started.
     Explained energy ratio from previous block: 0.0S0
     1.000 seconds of new data read


**** Violation discrepancy: BOUNDARY-DETECTION output.
     <BOUNDARY-EXT.0003> at block edge.
     Discrepancy Explanation proposed:
         (END-OF-DATA-BOUNDARY)
        ---Reprocessing Postponed until data in time
           [0.96 1.96] is available.


**** Out of the possible explanations:
       WIND,POLICECAR-SIREN-STREAM1
       POLICECAR-SIREN-STREAM2,
     for unexplained spectral bands in time [0.00 1.00],
     the following were initially selected:
       POLICECAR-SIREN-STREAM2, POLICECAR-SIREN-STREAM1
     to initially create
     <POLICECAR-SIREN-STREAM1#001>
     <POLICECAR-SIREN-STREAM2#001>


**** Attempting to confirm POLICECAR-SIREN-STREAM1#001's
     mstream at [2250 2281] Hz in time [0.00 0.50].
     Contours found: <contour.0005> <contour.0004>
                     <contour.0003> <contour.0002>

     4 contours confirm
     <policecar-siren.st1.mst4.0020 [2250 2281] Hz>
     in time ([0.05 0.40})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#001's
     mstream at [2813 2851] Hz in time [0.00 0.50].
     Contours found: <contour.0017> <contour.0016>
                     <contour.0015> <contour.0014>

     4 contours confirm
     <policecar-siren.st1.mst5.0025 [2813 2851] Hz>
     in time ([0.00 0.45})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#001's
```

```
     mstream at [3375 3414] Hz in time [0.00 0.50].
     Contours found: <contour.0024> <contour.0023>
                     <contour.0022>

     3 contours confirm
     <policecar-siren.st1.mst6.0031 [3375 3414] Hz>
     in time ([0.00 0.45})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#001's
     mstream at [550 580] Hz in time [0.00 0.42].
     Contours found: <contour.0030> <contour.0029>
                     <contour.0028>

     3 contours confirm
     <policecar-siren.st1.mst1.0036 [550 580] Hz>
     in time ([0.00 0.48})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#001's
     mstream at [1674 1710] Hz in time [0.00 0.50].
     Contours found: <contour.0036> <contour.0035>
                     <contour.0034>

     3 contours confirm
     <policecar-siren.st1.mst3.0038 [1674 1710] Hz>
     in time ([0.00 0.45})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#001's
     mstream at [1110 1150] Hz in time [0.08 0.35].
     Contours found: <contour.0041> <contour.0040>

     2 contours confirm
     <policecar-siren.st1.mst2.0041 [1110 1150] Hz>
     in time ([0.08 0.42})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#001's
     mstream at [1110 1150] Hz in time [0 0.08] [0.42 0.50].
     ---> mstream at [1110 1150] Hz unconfirmed

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [907 947] Hz in time [0.41 0.97].
     Contours found: <contour.0046> <contour.0045>
                     <contour.0044>

     3 contours confirm
     <policecar-siren.st2.mst2.0072 [907 947] Hz>
```

```
     in time ([0.45 0.94})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [1821 1843] Hz in time [0.44 0.93].
     Contours found: <contour.0052> <contour.0051>
                     <contour.0050>

     3 contours confirm
     <policecar-siren.st2.mst4.0074 [1821 1843] Hz>
     in time ([0.45 0.88})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [2274 2304] Hz in time [0.44 0.93].
     Contours found: <contour.0060> <contour.0059>
                     <contour.0058> <contour.0057>
                     <contour.0056>

     5 contours confirm
     <policecar-siren.st2.mst5.0083 [2274 2304] Hz>
     in time ([0.46 0.91})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [3180 3210] Hz in time [0.44 0.82].
     Contours found: <contour.0069> <contour.0068>
                     <contour.0067> <contour.0066>

     4 contours confirm
     <policecar-siren.st2.mst6.0087 [3180 3210] Hz>
     in time ([0.46 0.88})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [3634 3680] Hz in time [0.44 0.82].
     Contours found: <contour.0075>
                     <contour.0074>

     2 contours confirm
     <policecar-siren.st2.mst7.0094 [3634 3680] Hz>
     in time ([0.54 0.77})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [446 468] Hz in time [0.45 0.78].
     Contour found: <contour.0078>

     1 contour confirms
     <policecar-siren.st2.mst1.0097 [446 468] Hz>
```

```
     in time ([0.45 0.85})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [1360 1406] Hz in time [0.44 0.97].
     Contours found: <contour.0083> <contour.0082>
                     <contour.0081> <contour.0080>

     4 contours confirm
     <policecar-siren.st2.mst3.0099 [1360 1406] Hz>
     in time ([0.45 0.75} [0.80 0.91})

**** Trying to solve negative evidence in
     <policecar-siren.st2.mst7.0095 [3634 3680] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: <policecar-siren.st2.mst7.0095 [3634 3680] Hz>)
    REGION: T:[0.43 0.56] F:[3610 3704] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-ENERGY-THRESHOLDING)



**** Searching for reprocessing plans for
     <EXPLANATION (MS-ENERGY-THRESHOLDING)> with the goals
     ((HAVE-HYPOTHESES-SUPPORT
        (<policecar-siren.st2.mst7.0095 [3634 3680] Hz>)))


**** Reprocessing started in frequency [3610 3704]
     during time period [0.44 0.54] for Discr. Diag.
     <FRONT-END.0002> with
     *STFT-ABSOLUTE-NOISE-THRESHOLD* = 0.0258209S0
     *NUM-PEAKS-STFT-SPECTRUM* = 1
     *STFT-CONTOUR-FREQ-RADIUS* = (46 . 46)
     *STFT-CONTOUR-ENERGY-RADIUS* = (0.5 . 0.5)
     *PEAK-NEIGHBOURHOOD* = 1
     in effect.
     No microstreams found.
     Reprocessing completed. Global context restored.


**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [3634 3680] Hz in time [0.82 0.97].
     ---> mstream at [3634 3680] Hz unconfirmed
```

```
**** Explaining <POLICECAR-SIREN-STREAM2#001 0024 (0.83)> of
     <POLICECAR-SIREN-STREAM2#001> as
     #<POLICECAR-SIREN#001 (SCRIPTEXT.0001)>

**** Trying to solve negative evidence in
     <policecar-siren.st1.mst2.0043 [1110 1150] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st1.mst2.0043 [1110 1150] Hz>)
    REGION: T:[0.40 0.52] F:[1086 1174] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Explaining <POLICECAR-SIREN-STREAM1#001 0021 (0.84)> of
     <POLICECAR-SIREN-STREAM1#001> as
     #<POLICECAR-SIREN#001 (SCRIPTEXT.0002)>
```

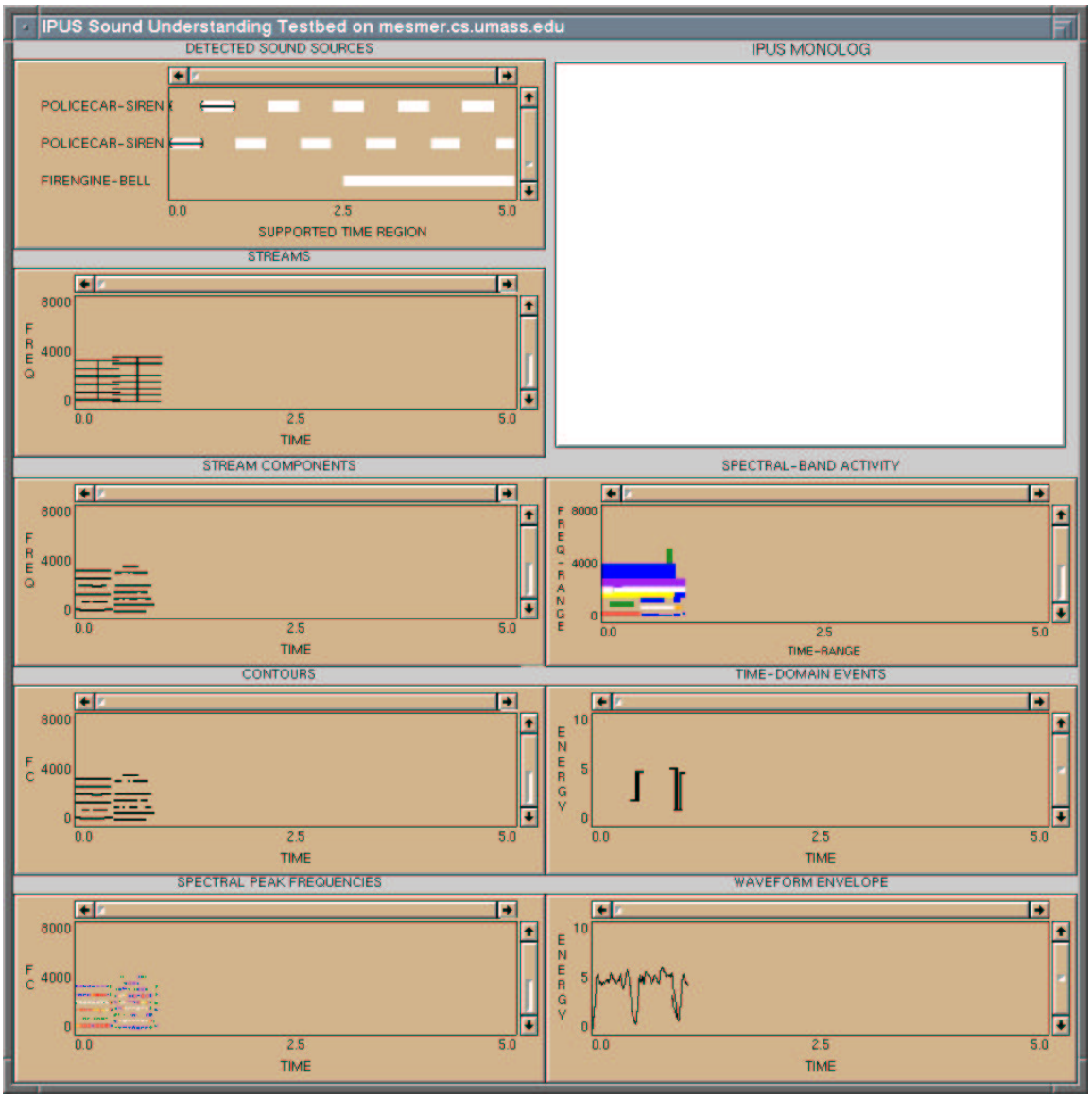


Figure B.3. Post Block-1 Status.

```
**** Processing of [1.000 2.000} time period started.
     Explained energy ratio from previous block: 0.78S0
     1.000 seconds of new data read


**** Creating Script Expectations
     <POLICECAR-SIREN-STREAM1#002>
     <POLICECAR-SIREN-STREAM2#002>

**** Searching for reprocessing plans for
     <EXPLANATION (END-OF-DATA-BOUNDARY)> with the goals
     ((HAVE-HYPOTHESES-SUPPORT (<BOUNDARY-EXT.0003>)))


**** Reprocessing started in frequency [0 8000]
     during time period [0.87 1.04] for Violation Discr.
     <FRONT-END.0003> with
     *MINIMUM-RELATIVE-DIFF* = 0.2
     in effect.

   Synthesizing confirmed BOUNDARY:
    <BOUNDARY-EXT.0004>
     Reprocessing completed. Global context restored.


**** Trying to solve negative evidence in
     <policecar-siren.st2.mst7.0102 [3634 3680] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st2.mst7.0102 [3634 3680] Hz>)
    REGION: T:[0.80 0.99] F:[3610 3704] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-ENERGY-THRESHOLDING)



**** Searching for reprocessing plans for
     <EXPLANATION (MS-ENERGY-THRESHOLDING)> with the goals
     ((HAVE-HYPOTHESES-SUPPORT
         (<policecar-siren.st2.mst7.0102 [3634 3680] Hz>)))


**** Reprocessing started in frequency [3610 3704]
     during time period [0.82 0.97] for Discr. Diag.
```

```
     <FRONT-END.0004> with
     *STFT-ABSOLUTE-NOISE-THRESHOLD* = 0.0258209S0
     *NUM-PEAKS-STFT-SPECTRUM* = 1
     *STFT-CONTOUR-FREQ-RADIUS* = (46 . 46)
     *STFT-CONTOUR-ENERGY-RADIUS* = (0.5 . 0.5)
     *PEAK-NEIGHBOURHOOD* = 4
     in effect.
     No microstreams found.
     Reprocessing completed. Global context restored.


**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [446 468] Hz in time [0.85 0.97].
     Contours found: <contour.0089>

     1 contour confirms
     <policecar-siren.st2.mst1.0103 [446 468] Hz>
     in time ([0.85 0.93})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [2274 2304] Hz in time [0.91 0.97].
     ---> mstream at [2274 2304] Hz unconfirmed

**** Attempting to confirm POLICECAR-SIREN-STREAM2#001's
     mstream at [3180 3210] Hz in time [0.88 0.97].
     ---> mstream at [3180 3210] Hz unconfirmed

**** Attempting to confirm POLICECAR-SIREN-STREAM1#002's
     mstream at [2250 2281] Hz in time [0.92 1.43].
     Contours found: <contour.0095> <contour.0094>
                     <contour.0093> <contour.0092>

     <contour.0095> <contour.0094> have incorrect energy

     <contour.0093> <contour.0092>
     confirm
     <policecar-siren.st1.mst4.0132 [2250 2281] Hz>
     in time ([1.13 1.34})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#002's
     mstream at [2813 2851] Hz in time [0.92 1.37].
     Contours found: <contour.0103> <contour.0102>
                     <contour.0101>

     3 contours confirm
```

```
     <policecar-siren.st1.mst5.0150 [2813 2851] Hz>
     in time ([0.93 1.18} [1.13 1.38})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#002's
     mstream at [3375 3414] Hz in time [0.92 1.37].
     Contour found: <contour.0107>

     1 contour confirms
     <policecar-siren.st1.mst6.0156 [3375 3414] Hz>
     in time ([0.93 1.38})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#002's
     mstream at [550 580] Hz in time [0.92 1.37].
     Contours found: <contour.0114> <contour.0113>
                     <contour.0112> <contour.0111>
                     <contour.0110> <contour.0109>

    <contour.0114>: limitation due to length.

     <contour.0113> <contour.0112>
     <contour.0111> <contour.0110>
     <contour.0109> confirm
     <policecar-siren.st1.mst1.0161 [550 580] Hz>
     in time ([0.93 1.35})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#002's
     mstream at [1674 1710] Hz in time [0.92 1.37].
     Contours found: <contour.0124> <contour.0123>
                     <contour.0122> <contour.0121>
                     <contour.0120>

     <contour.0124>: limitation due to length.

     <contour.0123> <contour.0122>
     <contour.0121> <contour.0120>
     confirm
     <policecar-siren.st1.mst3.0165 [1674 1710] Hz>
     in time ([0.93 1.35})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#002's
     mstream at [1110 1150] Hz in time [0.92 1.37].
     Contours found: <contour.0130> <contour.0129>

     2 contours confirm
     <policecar-siren.st1.mst2.0170 [1110 1150] Hz>
```

```
     in time ([1.02 1.13} [1.16 1.37})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#002's
     mstream at [907 947] Hz in time [1.39 1.89].
     Contours found: <contour.0136> <contour.0135>
                     <contour.0134> <contour.0133>

     4 contours confirm
     <policecar-siren.st2.mst2.0209 [907 947] Hz>
     in time ([1.40 1.69} [1.64 1.88})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#002's
     mstream at [1821 1843] Hz in time [1.39 1.89].
     Contours found: <contour.0143> <contour.0142>
                     <contour.0141>

     3 contours confirm
     <policecar-siren.st2.mst4.0211 [1821 1843] Hz>
     in time ([1.40 1.62} [1.58 1.83})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#002's
     mstream at [2274 2304] Hz in time [1.39 1.89].
     Contours found: <contour.0150> <contour.0149>
                     <contour.0148> <contour.0147>

     4 contours confirm
     <policecar-siren.st2.mst5.0220 [2274 2304] Hz>
     in time ([1.40 1.67} [1.62 1.85})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#002's
     mstream at [3180 3210] Hz in time [1.39 1.89].
     Contours found: <contour.0158> <contour.0157>
                     <contour.0156> <contour.0155>

     4 contours confirm
     <policecar-siren.st2.mst6.0226 [3180 3210] Hz>
     in time ([1.40 1.51} [1.54 1.82})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#002's
     mstream at [3634 3680] Hz in time [1.39 1.89].
     Contours found: <contour.0164> <contour.0163>

     2 contours confirm
     <policecar-siren.st2.mst7.0233 [3634 3680] Hz>
     in time ([1.50 1.70})
```

```
**** Attempting to confirm POLICECAR-SIREN-STREAM2#002's
     mstream at [446 468] Hz in time [1.39 1.89].
     Contour found: <contour.0167>

     1 contour confirms
     <policecar-siren.st2.mst1.0238 [446 468] Hz>
     in time ([1.40 1.86})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#002's
     mstream at [1360 1406] Hz in time [1.39 1.89].
     Contours found: <contour.0172> <contour.0171>
                     <contour.0170> <contour.0169>

     4 contours confirm
     <policecar-siren.st2.mst3.0240 [1360 1406] Hz>
     in time ([1.40 1.85})

**** Trying to solve negative evidence in
     <policecar-siren.st1.mst2.0173 [1110 1150] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st1.mst2.0173 [1110 1150] Hz>)
    REGION: T:[0.90 1.03] F:[1086 1174] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Out of the possible explanations:
       WIND,
     for unexplained spectral bands in time [1.00 2.00],
     the following were initially selected:
       WIND
     to initially create
       <WIND#001>


**** Attempting to confirm WIND#001's
     mstream at [525 766] Hz in time [1.70 2.00].
     Contour found: <contour.0177>

     1 contour confirms
     <wind.st1.mst1.0245 [525 766] Hz>
     in time ([1.70 1.99})
```

```
**** Attempting to confirm WIND#001's
     mstream at [525 766] Hz in time [1.40 1.70].
     ---> mstream at [525 766] Hz unconfirmed

**** Trying to solve negative evidence in
     <wind.st1.mst1.0246 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0246 [525 766] Hz>)
    REGION: T:[1.38 1.72] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm WIND#001's
     mstream at [525 766] Hz in time [1.10 1.40].
     Contour found: <contour.0179>

     1 contour confirms
     <wind.st1.mst1.0246 [525 766] Hz>
     in time ([1.11 1.46})

**** Attempting to confirm WIND#001's
     mstream at [525 766] Hz in time [0.80 1.10].
     Contour found: <contour.0181>

     1 contour confirms
     <wind.st1.mst1.0247 [525 766] Hz>
     in time ([0.90 1.16})

**** Trying to solve negative evidence in
     <wind.st1.mst1.0248 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0248 [525 766] Hz>)
    REGION: T:[0.78 0.91] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm WIND#001's
     mstream at [525 766] Hz in time [0.47 0.80].
```

```
---> mstream at [525 766] Hz unconfirmed
```

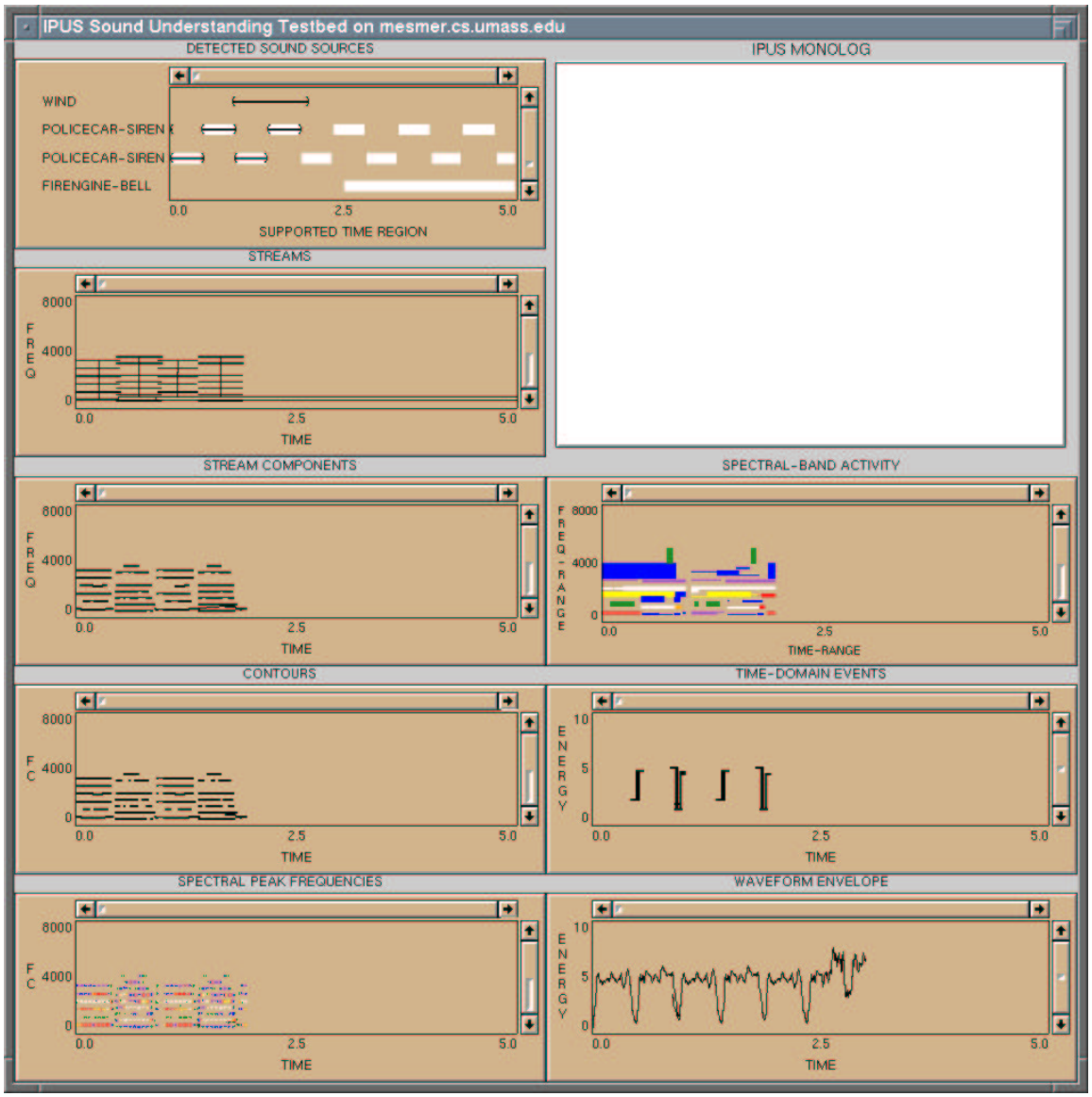


Figure B.4. Post Block-2 Status.

```
**** Processing of [2.000 3.000} time period started.
     Explained energy ratio from previous block: 0.82S0

     1.000 seconds of new data read


**** Creating Script Expectations
     <POLICECAR-SIREN-STREAM1#003>
     <POLICECAR-SIREN-STREAM2#003>

**** Attempting to confirm WIND#001's
     mstream at [525 766] Hz in time [1.99 3.00].
     Contours found: <contour.0184> <contour.0183>

     2 contours confirm
     <wind.st1.mst1.0249 [525 766] Hz>
     in time ([2.00 2.35} [2.78 2.99})

**** Trying to solve negative evidence in
     <wind.st1.mst1.0250 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0250 [525 766] Hz>)
    REGION: T:[2.34 2.80] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL

    Source WIND#001 is now disbelieved.



**** Attempting to confirm POLICECAR-SIREN-STREAM1#003's
     mstream at [2250 2281] Hz in time [1.86 2.37].
     Contours found: <contour.0188> <contour.0187>

     2 contours confirm
     <policecar-siren.st1.mst4.0270 [2250 2281] Hz>
     in time ([2.06 2.29})


**** Attempting to confirm POLICECAR-SIREN-STREAM1#003's
     mstream at [2813 2851] Hz in time [1.86 2.31].
     Contours found: <contour.0201> <contour.0200>
                     <contour.0199> <contour.0198>
```

```
     4 contours confirm
     <policecar-siren.st1.mst5.0288 [2813 2851] Hz>
     in time ([1.86 2.13} [2.08 2.32})


**** Attempting to confirm POLICECAR-SIREN-STREAM1#003's
     mstream at [3375 3414] Hz in time [1.86 2.31].
     Contour found: <contour.0206>

     1 contour confirms
     <policecar-siren.st1.mst6.0294 [3375 3414] Hz>
     in time ([1.86 2.32})


**** Attempting to confirm POLICECAR-SIREN-STREAM1#003's
     mstream at [550 580] Hz in time [1.86 2.31].
     Contours found: <contour.0210> <contour.0209>
                     <contour.0208>

     <contour.0210>: limitation due to length.

     <contour.0209> <contour.0208>
     confirm <policecar-siren.st1.mst1.0299 [550 580] Hz>
     in time ([1.86 2.27})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#003's
     mstream at [1674 1710] Hz in time [1.86 2.31].
     Contours found: <contour.0216> <contour.0215>
                     <contour.0214> <contour.0213>

     4 contours confirm
     <policecar-siren.st1.mst3.0303 [1674 1710] Hz>
     in time ([1.88 2.16} [2.11 2.34})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#003's
     mstream at [1110 1150] Hz in time [1.86 2.31].
     Contour found: <contour.0223> <contour.0222>
                    <contour.0221>

     3 contours confirm
     <policecar-siren.st1.mst2.0308 [1110 1150] Hz>
     in time ([2.00 2.06} [2.03 2.30})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#003's
     mstream at [907 947] Hz in time [2.33 2.83].
```

```
     Contours found: <contour.0230> <contour.0229>
                     <contour.0228> <contour.0227>

     4 contours confirm
     <policecar-siren.st2.mst2.0345 [907 947] Hz>
     in time ([2.34 2.69} [2.64 2.82})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#003's
     mstream at [1821 1843] Hz in time [2.33 2.83].
     Contours found: <contour.0237> <contour.0236>
                     <contour.0235>

     3 contours confirm
     <policecar-siren.st2.mst4.0347 [1821 1843] Hz>
     in time ([2.34 2.56} [2.51 2.77})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#003's
     mstream at [2274 2304] Hz in time [2.33 2.83].
     Contours found: <contour.0245> <contour.0244>
                     <contour.0243> <contour.0242>
                     <contour.0241>

     5 contours confirm
     <policecar-siren.st2.mst5.0356 [2274 2304] Hz>
     in time ([2.34 2.78})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#003's
     mstream at [3180 3210] Hz in time [2.33 2.83].
     Contours found: <contour.0253> <contour.0252>
                     <contour.0251>

     3 contours confirm
     <policecar-siren.st2.mst6.0362 [3180 3210] Hz>
     in time ([2.35 2.72})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#003's
     mstream at [3634 3680] Hz in time [2.33 2.83].
     Contours found: <contour.0259> <contour.0258>
                     <contour.0257>

     3 contours confirm
     <policecar-siren.st2.mst7.0369 [3634 3680] Hz>
     in time ([2.43 2.75})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#003's
```

```
     mstream at [446 468] Hz in time [2.33 2.83].
     Contour found: <contour.0266> <contour.0265>
                    <contour.0264> <contour.0263>

     4 contours confirm
     <policecar-siren.st2.mst1.0374 [446 468] Hz>
     in time ([2.34 2.74})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#003's
     mstream at [1360 1406] Hz in time [2.33 2.83].
     Contours found: <contour.0275> <contour.0274>
                     <contour.0273> <contour.0272>

     4 contours confirm
     <policecar-siren.st2.mst3.0376 [1360 1406] Hz>
     in time ([2.34 2.80})


**** Out of the possible explanations:
       WIND,FIRENGINE-BELL
     for unexplained spectral bands in time [2.00 3.00],
     the following were initially selected:
       FIRENGINE-BELL
     to initially create
     <FIRENGINE-BELL#001>


**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3070 3149] Hz in time [2.58 2.93].
     Contour found: <contour.0280>

     1 contour confirms
     <firengine-bell.st1.mst3.0397 [3070 3149] Hz>
     in time ([2.58 2.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1719 1750] Hz in time [2.58 2.93].
     Contour found: <contour.0282>

     1 contour confirms
     <firengine-bell.st1.mst6.0405 [1719 1750] Hz>
     in time ([2.58 2.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1945 1969] Hz in time [2.58 2.93].
```

```
     Contour found: <contour.0284>

     1 contour confirms
     <firengine-bell.st1.mst5.0410 [1945 1969] Hz>
     in time ([2.58 2.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [2242 2266] Hz in time [2.00 2.93].
     Contours found: <contour.0291> <contour.0290>
                     <contour.0289> <contour.0288>

     <contour.0291> <contour.0290>
     <contour.0289> have incorrect energy

     <contour.0288> confirms
     <firengine-bell.st1.mst4.0417 [2242 2266] Hz>
     in time ([2.26 2.86})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3304 3329] Hz in time [2.58 2.93].
     Contour found: <contour.0293>

     <contour.0293> has incorrect energy

     ---> mstream at [3304 3329] Hz unconfirmed

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [4210 4235] Hz in time [2.70 3.00].
     Contour found: <contour.0294>

     1 contour confirms
     <firengine-bell.st1.mst1.0422 [4210 4235] Hz>
     in time ([2.70 2.99})

**** Trying to solve negative evidence in
     <firengine-bell.st1.mst2.0423 [3304 3329] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst2.0423 [3304 3329] Hz>)
    REGION: T:[2.56 2.94] F:[3288 3345] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL
```

```
**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3304 3329] Hz in time [2.28 2.58].
     Contours found: <contour.0297> <contour.0296>

     <contour.0297> <contour.0296>
     have incorrect energy

     ---> mstream at [3304 3329] Hz unconfirmed

**** Trying to solve negative evidence in
     <firengine-bell.st1.mst4.0420 [2242 2266] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst4.0420 [2242 2266] Hz>)
    REGION: T:[1.98 2.27] F:[2226 2282] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [4210 4235] Hz in time [2.40 2.70].
     Contour found: <contour.0298>

     1 contour confirms
     <firengine-bell.st1.mst1.0426 [4210 4235] Hz>
     in time ([2.51 2.75})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3070 3149] Hz in time [2.28 2.58].
     Contour found: <contour.0300>

     1 contour confirms
     <firengine-bell.st1.mst3.0432 [3070 3149] Hz>
     in time ([2.53 2.64})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1945 1969] Hz in time [2.28 2.58].
     Contour found: <contour.0302>

     1 contour confirms
     <firengine-bell.st1.mst5.0435 [1945 1969] Hz>
     in time ([2.50 2.64})

**** Attempting to confirm FIRENGINE-BELL#001's
```

```
     mstream at [1719 1750] Hz in time [2.28 2.58].
     Contour found: <contour.0304>

     <contour.0304> has incorrect energy
     ---> mstream at [1719 1750] Hz unconfirmed

**** Trying to solve negative evidence in
     <firengine-bell.st1.mst2.0436 [3304 3329] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst2.0436 [3304 3329] Hz>)
    REGION: T:[2.26 2.59] F:[3288 3345] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3304 3329] Hz in time [1.98 2.28].
     Contour found: <contour.0305>

     <contour.0305> has incorrect energy
     ---> mstream at [3304 3329] Hz unconfirmed



**** Trying to solve negative evidence in
     <firengine-bell.st1.mst5.0440 [1945 1969] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst5.0440 [1945 1969] Hz>)
    REGION: T:[2.26 2.51] F:[1929 1985] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Trying to solve negative evidence in
     <firengine-bell.st1.mst1.0439 [4210 4235] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst1.0439 [4210 4235] Hz>)
    REGION: T:[2.38 2.53] F:[4194 4251] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL
```

```
**** Trying to solve negative evidence in
     <firengine-bell.st1.mst3.0434 [3070 3149] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst3.0434 [3070 3149] Hz>)
    REGION: T:[2.26 2.54] F:[3054 3165] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Trying to solve negative evidence in
     <firengine-bell.st1.mst6.0441 [1719 1750] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst6.0441 [1719 1750] Hz>)
    REGION: T:[2.26 2.59] F:[1703 1766] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Trying to solve negative evidence in
     <firengine-bell.st1.mst2.0436 [3304 3329] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst2.0436 [3304 3329] Hz>)
    REGION: T:[1.96 2.29] F:[3288 3345] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3304 3329] Hz in time [1.68 1.98].
     Contour found: <contour.0310>

     <contour.0310> has incorrect energy
     ---> mstream at [3304 3329] Hz unconfirmed

**** Trying to solve negative evidence in
     <firengine-bell.st1.mst4.0438 [2242 2266] Hz>
```

```
    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst4.0438 [2242 2266] Hz>)
    REGION: T:[1.68 2.02] F:[2226 2282] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-FREQUENCY-RESOLUTION)



**** Searching for reprocessing plans for
     (MS-FREQUENCY-RESOLUTION) with the goals
     ((HAVE-HYPOTHESES-SUPPORT
          (<firengine-bell.st1.mst4.0438 [2242 2266] Hz>
           <policecar-siren.st2.mst5.0224 [2274 2304] Hz>)))


**** Reprocessing started in frequency [2234 2305]
     during time period [1.34 2.36] for Discr. Diag.
     Attempt to assign 16384 to *FFT-SIZE*
     [constraint violation, reprocessing fails]
     Reprocessing completed. Global context restored.


**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1719 1750] Hz in time [2.00 2.28].
     ---> mstream at [1719 1750] Hz unconfirmed

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3070 3149] Hz in time [2.00 2.28].
     ---> mstream at [3070 3149] Hz unconfirmed

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1945 1969] Hz in time [2.00 2.28].
     ---> mstream at [1945 1969] Hz unconfirmed

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [4210 4235] Hz in time [2.00 2.40].
     Contour found: <contour.0312>

     <contour.0312> has incorrect energy
     ---> mstream at [4210 4235] Hz unconfirmed
```

```
**** Trying to solve negative evidence in
     <firengine-bell.st1.mst6.0445 [1719 1750] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst6.0445 [1719 1750] Hz>)
    REGION: T:[1.98 2.29] F:[1703 1766] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Trying to solve negative evidence in
     <firengine-bell.st1.mst5.0447 [1945 1969] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst5.0447 [1945 1969] Hz>)
    REGION: T:[1.98 2.29] F:[1929 1985] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Trying to solve negative evidence in
     <firengine-bell.st1.mst1.0446 [4210 4235] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst1.0446 [4210 4235] Hz>)
    REGION: T:[1.98 2.42] F:[4194 4251] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Trying to solve negative evidence in
     <firengine-bell.st1.mst3.0443 [3070 3149] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst3.0443 [3070 3149] Hz>)
    REGION: T:[1.98 2.29] F:[3054 3165] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [2242 2266] Hz in time [0.47 1.70].
     Contours found: <contour.0316> <contour.0313>
```

```
     <contour.0316> <contour.0313>
     have incorrect energy

     ---> mstream at [2242 2266] Hz unconfirmed


**** Out of the possible explanations:
       WIND,
     for unexplained spectral bands in time [1.00 2.00],
     the following were initially selected:
       WIND
     to initially create
     <WIND#002>


**** Attempting to confirm WIND#002's
     mstream at [525 766] Hz in time [2.78 2.93].
     Contour found: <contour.0318>

     1 contour confirms
     <wind.st1.mst1.0448 [525 766] Hz>
     in time ([2.78 2.93})

**** Attempting to confirm WIND#002's
     mstream at [525 766] Hz in time [2.48 2.78].

     ---> mstream at [525 766] Hz unconfirmed

**** Trying to solve negative evidence in
     <wind.st1.mst1.0449 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0449 [525 766] Hz>)
    REGION: T:[2.47 2.80] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm WIND#002's
     mstream at [525 766] Hz in time [2.18 2.48].
     Contour found: <contour.0321>

     1 contour confirms
     <wind.st1.mst1.0449 [525 766] Hz>
```

```
     in time ([2.19 2.35})

**** Trying to solve negative evidence in
     <wind.st1.mst1.0450 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0450 [525 766] Hz>)
    REGION: T:[2.34 2.50] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm WIND#002's
     mstream at [525 766] Hz in time [1.88 2.18].
     Contour found: <contour.0323>

     1 contour confirms
     <wind.st1.mst1.0450 [525 766] Hz>
     in time ([1.90 2.24})

**** Attempting to confirm WIND#002's
     mstream at [525 766] Hz in time [1.58 1.88].
     1 short contours found
     ---> mstream at [525 766] Hz unconfirmed

**** Trying to solve negative evidence in
     <wind.st1.mst1.0451 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0451 [525 766] Hz>)
    REGION: T:[1.57 1.90] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm WIND#002's
     mstream at [525 766] Hz in time [0.47 1.58].
     Contour found: <contour.0326>

     <contour.0326> has incorrect energy
     ---> mstream at [525 766] Hz unconfirmed
```

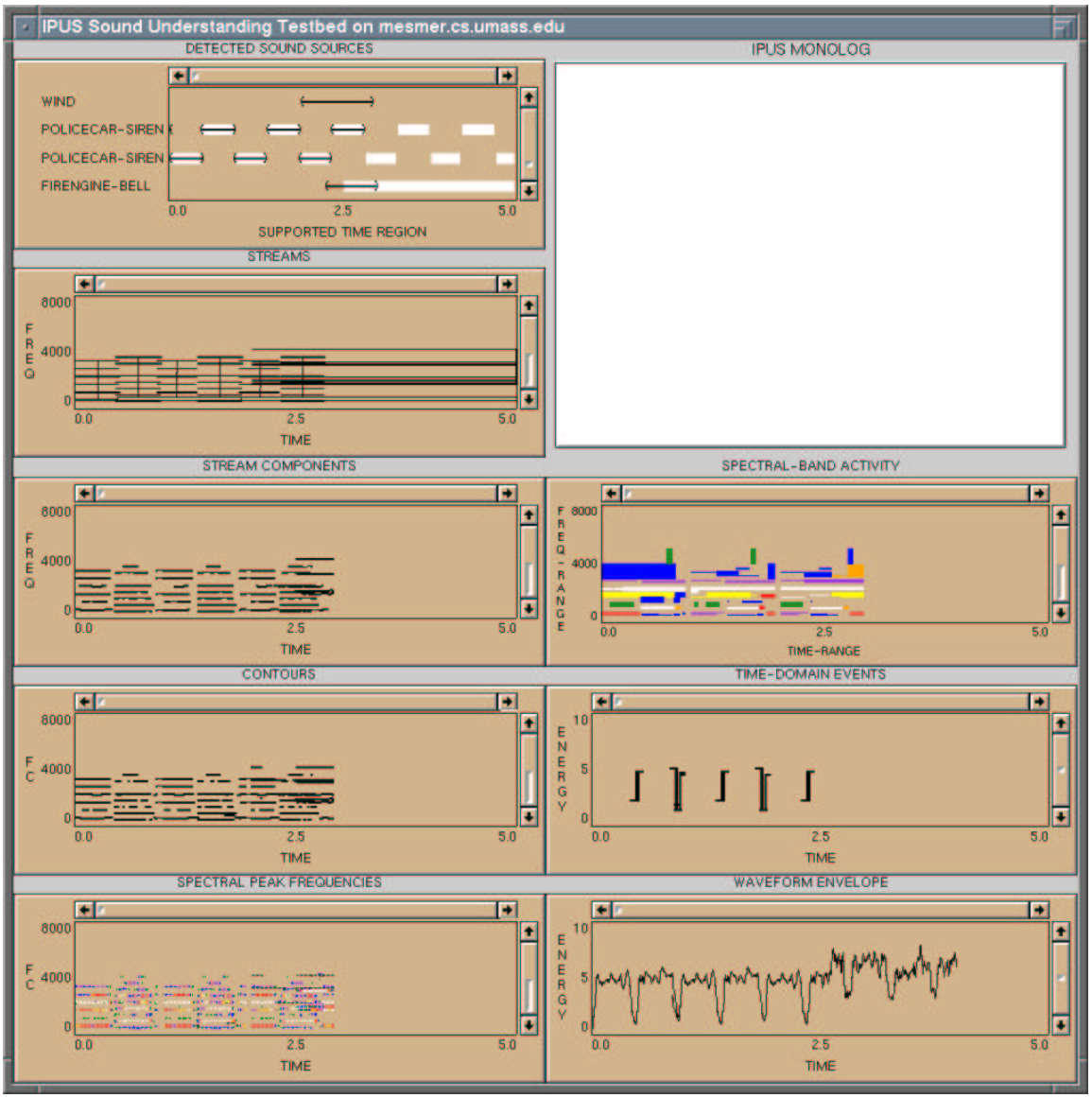


Figure B.5. Post Block-3 Status.

```
**** Processing of [3.000 4.000} time period started.
     Explained energy ratio from previous block: 0.79S0
     1.000 seconds of new data read


**** Creating Script Expectations
     <POLICECAR-SIREN-STREAM1#004>
     <POLICECAR-SIREN-STREAM2#004>

**** Attempting to confirm WIND#002's
     mstream at [525 766] Hz in time [2.93 4.00].
     Contours found: <contour.0328> <contour.0327>

     2 contours confirm
     <wind.st1.mst1.0452 [525 766] Hz>
     in times ([2.93 3.30} [3.72 3.99})


**** Trying to solve negative evidence in
     <wind.st1.mst1.0453 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0453 [525 766] Hz>)
    REGION: T:[3.29 3.74] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


    Source WIND#002 is now disbelieved.



**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3304 3329] Hz in time [2.93 4.00].
     Contour found: <contour.0331>

     ---> mstream at [3304 3329] Hz unconfirmed

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1719 1750] Hz in time [2.99 4.00].
     Contour found: <contour.0332>

     1 contour confirms
     <firengine-bell.st1.mst6.0445 [1719 1750] Hz>
     in time ([3.00 3.99})
```

```
**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3070 3149] Hz in time [2.99 4.00].
     Contours found: <contour.0336> <contour.0335>
                     <contour.0334>

     3 contours confirm
     <firengine-bell.st1.mst3.0458 [3070 3149] Hz>
     in time ([3.46 3.99} [3.00 3.35})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [4210 4235] Hz in time [2.99 4.00].
     Contours found: <contour.0341> <contour.0340>

     2 contours confirm
     <firengine-bell.st1.mst1.0463 [4210 4235] Hz>
     in time ([3.00 3.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1945 1969] Hz in time [2.99 4.00].
     Contour found: <contour.0344>

     1 contour confirms
     <firengine-bell.st1.mst5.0464 [1945 1969] Hz>
     in time ([3.00 3.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [2242 2266] Hz in time [2.93 4.00].
     Contours found: <contour.0353> <contour.0352>
                     <contour.0351> <contour.0349>

     <contour.0353> <contour.0352>
     have incorrect energy

     <contour.0351> <contour.0349>
     confirm
     <firengine-bell.st1.mst4.0461 [2242 2266] Hz>
     in time ([3.11 3.24} [3.32 3.69})

**** Trying to solve negative evidence in
     <firengine-bell.st1.mst3.0460 [3070 3149] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst3.0460 [3070 3149] Hz>)
    REGION: T:[3.34 3.48] F:[3054 3165] E:[0 10000.00]
```

```
    Discrepancy Explanation Proposed: NIL


**** Attempting to confirm POLICECAR-SIREN-STREAM1#004's
     mstream at [2250 2281] Hz in time [2.80 3.31].
     Contours found: <contour.0365> <contour.0364>

     <contour.0365> has incorrect energy

     <contour.0364> confirms
     <policecar-siren.st1.mst4.0488 [2250 2281] Hz>
     in time ([3.11 3.22})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#004's
     mstream at [2813 2851] Hz in time [2.80 3.25].
     Contours found: <contour.0374> <contour.0373>
                     <contour.0372>

     3 contours confirm
     <policecar-siren.st1.mst5.0505 [2813 2851] Hz>
     in time ([2.80 2.93} [2.91 3.26})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#004's
     mstream at [3375 3414] Hz in time [2.80 3.25].
     Contours found: <contour.0381> <contour.0380>
                     <contour.0379> <contour.0378>

     4 contours confirm
     <policecar-siren.st1.mst6.0511 [3375 3414] Hz>
     in time ([2.80 2.99} [3.00 3.24})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#004's
     mstream at [550 580] Hz in time [2.80 3.25].
     Contours found: <contour.0388> <contour.0387>
                     <contour.0386>

     <contour.0388>: limitation due to length.

     <contour.0387> <contour.0386>
     confirm
     <policecar-siren.st1.mst1.0516 [550 580] Hz>
     in time ([2.80 3.21})
```

```
**** Attempting to confirm POLICECAR-SIREN-STREAM1#004's
     mstream at [1674 1710] Hz in time [2.80 3.25].
     Contours found: <contour.0393> <contour.0392>
                     <contour.0391>

     3 contours confirm
     <policecar-siren.st1.mst3.0520 [1674 1710] Hz>
     in time ([3.02 3.16} [3.18 3.27})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#004's
     mstream at [1110 1150] Hz in time [2.80 3.25].
     Contour found: <contour.0399> <contour.0398>
                    <contour.0397>

     3 contours confirm
     <policecar-siren.st1.mst2.0525 [1110 1150] Hz>
     in time ([2.91 2.99} [3.06 3.27})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#004's
     mstream at [907 947] Hz in time [3.27 3.77].
     Contours found: <contour.0406> <contour.0405>
                     <contour.0404> <contour.0403>

     4 contours confirm
     <policecar-siren.st2.mst2.0562 [907 947] Hz>
     in time ([3.27 3.62} [3.58 3.77})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#004's
     mstream at [1821 1843] Hz in time [3.27 3.77].
     Contours found: <contour.0413> <contour.0412>
                     <contour.0411>

     3 contours confirm
     <policecar-siren.st2.mst4.0564 [1821 1843] Hz>
     in time ([3.27 3.50} [3.45 3.70})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#004's
     mstream at [2274 2304] Hz in time [3.27 3.77].
     Contours found: <contour.0421> <contour.0420>
                     <contour.0419> <contour.0418>
                     <contour.0417>

     5 contours confirm
     <policecar-siren.st2.mst5.0573 [2274 2304] Hz>
     in time ([3.29 3.53} [3.50 3.74})
```

```
**** Attempting to confirm POLICECAR-SIREN-STREAM2#004's
     mstream at [3180 3210] Hz in time [3.27 3.77].
     Contour found: <contour.0429> <contour.0428>
                    <contour.0427>

     3 contours confirm
     <policecar-siren.st2.mst6.0579 [3180 3210] Hz>
     in time ([3.30 3.40} [3.42 3.66})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#004's
     mstream at [3634 3680] Hz in time [3.27 3.77].
     Contours found: <contour.0435> <contour.0434>
                     <contour.0433>

     3 contours confirm
     <policecar-siren.st2.mst7.0586 [3634 3680] Hz>
     in time ([3.37 3.48} [3.43 3.64})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#004's
     mstream at [446 468] Hz in time [3.27 3.77].
     Contours found: <contour.0442> <contour.0441>
                     <contour.0440> <contour.0439>

     4 contours confirm
     <policecar-siren.st2.mst1.0584 [446 468] Hz>
     in time ([3.27 3.53} [3.53 3.67})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#004's
     mstream at [1360 1406] Hz in time [3.27 3.77].
     Contours found: <contour.0450> <contour.0449>
                     <contour.0448> <contour.0447>

     4 contours confirm
     <policecar-siren.st2.mst3.0585 [1360 1406] Hz>
     in time ([3.27 3.61} [3.62 3.74})

**** Trying to solve negative evidence in
     <policecar-siren.st1.mst3.0523 [1674 1710] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st1.mst3.0523 [1674 1710] Hz>)
    REGION: T:[2.78 3.03] F:[1650 1734] E:[0 10000.00]

    Discrepancy Explanation Proposed:
```

```
        (MS-FREQUENCY-RESOLUTION)


**** Searching for reprocessing plans for
     <EXPLANATION (MS-FREQUENCY-RESOLUTION)> with the goals
     ((HAVE-HYPOTHESES-SUPPORT
         (<policecar-siren.st1.mst3.0523 [1674 1710] Hz>
          <firengine-bell.st1.mst6.0465 [1719 1750] Hz>)))


**** Reprocessing started in frequency [1671 1750]
     during time period [2.40 3.42] for Discr. Diag.
     Attempt to assign 16384 to *FFT-SIZE*
     [constraint violation, reprocessing fails]
     Reprocessing completed. Global context restored.


**** Trying to solve negative evidence in
     <policecar-siren.st1.mst2.0528 [1110 1150] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st1.mst2.0528 [1110 1150] Hz>)
    REGION: T:[2.78 2.93] F:[1086 1174] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-PEAK-THRESHOLDING MS-ENERGY-THRESHOLDING)



**** Searching for reprocessing plans for
     (MS-PEAK-THRESHOLDING MS-ENERGY-THRESHOLDING)
     with the goals
     ((HAVE-HYPOTHESES-SUPPORT
       (<policecar-siren.st1.mst2.0528 [1110 1150] Hz>)))


**** Reprocessing started in frequency [1086 1174]
     during time period [2.80 2.91] for Discr. Diag.
     <FRONT-END.0005> with
     *STFT-ABSOLUTE-NOISE-THRESHOLD* = 0.001408428S0
     *NUM-PEAKS-STFT-SPECTRUM* = 1
     *STFT-CONTOUR-FREQ-RADIUS* = (40 . 40)
     *PEAK-NEIGHBOURHOOD* = 3
     in effect.
     Contours Found!
```

```
     Synthesizing the following mstreams: {
     (<policecar-siren.st1.mst2.0528 [1110 1150] Hz>)}
     Reprocessing completed. Global context restored.


**** Out of the possible explanations:
       WIND,
     for unexplained spectral bands in
     time [3.00 4.00], the following
     were initially selected:
       WIND
     to initially create:
       <WIND#003> <WIND#004>


**** Attempting to confirm WIND#004's
     mstream at [525 766] Hz in time [3.70 4.00].
     Contour found: <contour.0457>

     1 contour confirms
     <wind.st1.mst1.0599 [525 766] Hz>
     in time ([3.70 3.99})

**** Attempting to confirm WIND#004's
     mstream at [525 766] Hz in time [3.40 3.70].
     ---> mstream at [525 766] Hz unconfirmed

**** Trying to solve negative evidence in
     <wind.st1.mst1.0600 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0600 [525 766] Hz>)
    REGION: T:[3.38 3.72] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm WIND#004's
     mstream at [525 766] Hz in time [3.10 3.40].
     Contour found: <contour.0459>

     1 contour confirms
     <wind.st1.mst1.0600 [525 766] Hz>
     in time ([3.11 3.30})
```

```
**** Trying to solve negative evidence in
     <wind.st1.mst1.0601 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0601 [525 766] Hz>)
    REGION: T:[3.29 3.42] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Attempting to confirm WIND#004's
     mstream at [525 766] Hz in time [2.80 3.10].
     Contour found: <contour.0461>

     1 contour confirms
     <wind.st1.mst1.0601 [525 766] Hz>
     in time ([2.82 3.10})

**** Attempting to confirm WIND#004's
     mstream at [525 766] Hz in time [2.50 2.80].
     ---> mstream at [525 766] Hz unconfirmed

**** Trying to solve negative evidence in
     <wind.st1.mst1.0602 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0602 [525 766] Hz>)
    REGION: T:[2.48 2.82] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Attempting to confirm WIND#004's
     mstream at [525 766] Hz in time [0.47 2.50].
     Contours found: <contour.0468> <contour.0467>

     <contour.0468> <contour.0467>
     have incorrect energy

     ---> mstream at [525 766] Hz unconfirmed
```

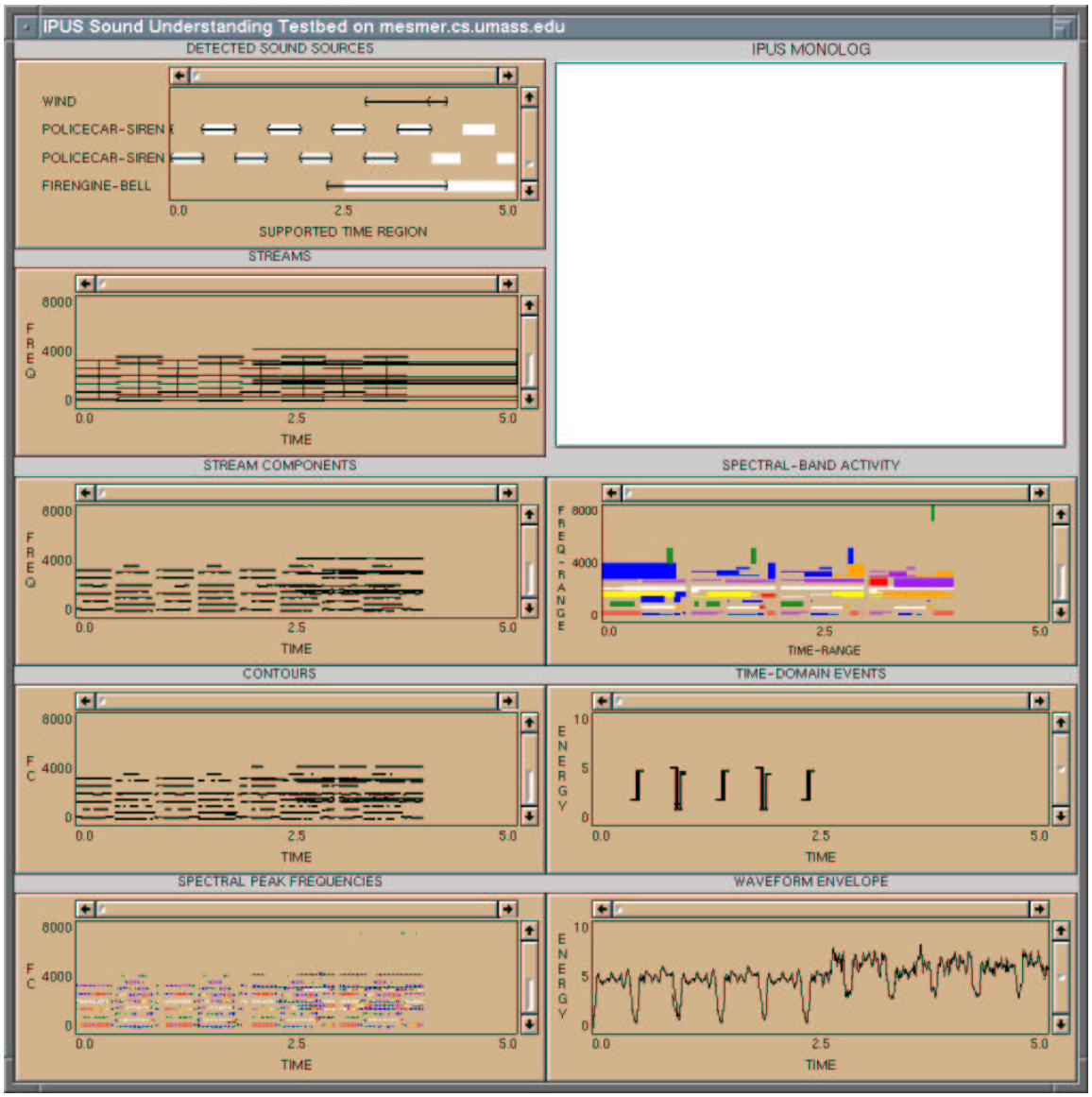


Figure B.6. Post Block-4 Status.

```
**** Processing of [4.000 5.000} time period started.
     Explained energy ratio from previous block: 0.71S0
     1.000 seconds of new data read

**** Creating Script Expectations
     <POLICECAR-SIREN-STREAM1#005>
     <POLICECAR-SIREN-STREAM2#005>


**** Trying to solve negative evidence in
     <firengine-bell.st1.mst2.0462 [3304 3329] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst2.0462 [3304 3329] Hz>)
    REGION: T:[2.91 4.02] F:[3288 3345] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Trying to solve negative evidence in
     <firengine-bell.st1.mst4.0468 [2242 2266] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst4.0468 [2242 2266] Hz>)
    REGION: T:[3.67 4.02] F:[2226 2282] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Trying to solve negative evidence in
     <firengine-bell.st1.mst4.0468 [2242 2266] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst4.0468 [2242 2266] Hz>)
    REGION: T:[2.91 3.13] F:[2226 2282] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3304 3329] Hz in time [4.00 5.00].
     Contour found: <contour-ext.0469>
```

```
     1 contour confirms
     <firengine-bell.st1.mst2.0469 [3304 3329] Hz>
     in time ([4.00 4.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [2242 2266] Hz in time [4.00 5.00].
     Contours found: <contour-ext.0477> <contour-ext.0476>
                     <contour-ext.0475> <contour-ext.0474>
                     <contour-ext.0473>

     <contour-ext.0473> has incorrect energy

     <contour-ext.0477> <contour-ext.0476>
     <contour-ext.0475> <contour-ext.0474>
     confirm
     <firengine-bell.st1.mst4.0606 [2242 2266] Hz>
     in time ([4.00 4.43} [4.64 4.75} [4.88 4.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [3070 3149] Hz in time [3.99 5.00].
     Contour found: <contour-ext.0482>

     1 contour confirms
     <firengine-bell.st1.mst3.0607 [3070 3149] Hz>
     in time ([4.00 4.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [4210 4235] Hz in time [3.99 5.00].
     Contour found: <contour-ext.0484>

     1 contour confirms
     <firengine-bell.st1.mst1.0614 [4210 4235] Hz>
     in time ([4.00 4.99})

**** Attempting to confirm FIRENGINE-BELL#001's
     mstream at [1719 1750] Hz in time [3.99 5.00].
     Contours found: <contour-ext.0488> <contour-ext.0487>
                     <contour-ext.0486>

     3 contours confirm
     <firengine-bell.st1.mst6.0616 [1719 1750] Hz>
     in time ([4.30 4.51} [4.00 4.14} [4.56 4.99})

**** Attempting to confirm FIRENGINE-BELL#001's
```

```
     mstream at [1945 1969] Hz in time [3.99 5.00].
     Contours found: <contour-ext.0494> <contour-ext.0493>
                     <contour-ext.0492>

     3 contours confirm
     <firengine-bell.st1.mst5.0615 [1945 1969] Hz>
     in time ([4.50 4.69} [4.00 4.29} [4.74 4.99})


**** Trying to solve negative evidence in
     <firengine-bell.st1.mst4.0617 [2242 2266] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst4.0617 [2242 2266] Hz>)
    REGION: T:[4.42 4.66] F:[2226 2282] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Trying to solve negative evidence in
     <firengine-bell.st1.mst4.0617 [2242 2266] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst4.0617 [2242 2266] Hz>)
    REGION: T:[4.74 4.90] F:[2226 2282] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Trying to solve negative evidence in
     <firengine-bell.st1.mst6.0619 [1719 1750] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst6.0619 [1719 1750] Hz>)
    REGION: T:[4.13 4.32] F:[1703 1766] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL



**** Trying to solve negative evidence in
     <firengine-bell.st1.mst5.0620 [1945 1969] Hz>
```

```
    Performing Discrepancy Diagnosis.
    OBJECT: (<firengine-bell.st1.mst5.0620 [1945 1969] Hz>)
    REGION: T:[4.27 4.51] F:[1929 1985] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


**** Attempting to confirm WIND#004's
     mstream at [525 766] Hz in time [3.99 5.00].
     Contours found: <contour.0470>
                     <contour.0469>

     2 contours confirm
     <wind.st1.mst1.0603 [525 766] Hz>
     in time ([4.00 4.24} [4.66 4.99})

**** Trying to solve negative evidence in
     <wind.st1.mst1.0604 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0604 [525 766] Hz>)
    REGION: T:[4.22 4.67] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL


    Source WIND#004 is now disbelieved.

**** Attempting to confirm WIND#003's
     mstream at [525 766] Hz in time [3.72 3.93].

     ---> mstream at [525 766] Hz unconfirmed

    Source WIND#003 is now disbelieved.


**** Attempting to confirm POLICECAR-SIREN-STREAM1#005's
     mstream at [2250 2281] Hz in time [3.74 4.25].
     Contours found: <contour.0477> <contour.0476>

     2 contours confirm
     <policecar-siren.st1.mst4.0625 [2250 2281] Hz>
     in time ([4.00 4.16})
```

```
**** Attempting to confirm POLICECAR-SIREN-STREAM1#005's
     mstream at [2813 2851] Hz in time [3.74 4.19].
     Contours found: <contour.0488> <contour.0487>

     2 contours confirm
     <policecar-siren.st1.mst5.0636 [2813 2851] Hz>
     in time ([3.75 4.19})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#005's
     mstream at [3375 3414] Hz in time [3.74 4.19].
     Contours found: <contour.0494> <contour.0493>
                     <contour.0492> <contour.0491>

     4 contours confirm
     <policecar-siren.st1.mst6.0641 [3375 3414] Hz>
     in time ([3.75 4.18})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#005's
     mstream at [550 580] Hz in time [3.74 4.19].
     Contours found: <contour.0500> <contour.0499>

     <contour.0500>: limitation due to length.

     <contour.0499> confirms
     <policecar-siren.st1.mst1.0645 [550 580] Hz>
     in time ([3.75 4.14})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#005's
     mstream at [1674 1710] Hz in time [3.74 4.19].
     Contours found: <contour.0505> <contour.0504>
                     <contour.0503> <contour.0502>

     <contour.0505>: limitation due to length.

     <contour.0504> <contour.0503>
     <contour.0502>
     confirm
     <policecar-siren.st1.mst3.0652 [1674 1710] Hz>
     in time ([3.75 4.11})

**** Attempting to confirm POLICECAR-SIREN-STREAM1#005's
     mstream at [1110 1150] Hz in time [3.74 4.19].
     Contours found: <contour.0511> <contour.0510>
                     <contour.0509>
```

```
     3 contours confirm
     <policecar-siren.st1.mst2.0656 [1110 1150] Hz>
     in time ([3.85 3.93} [4.00 4.21})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#005's
     mstream at [907 947] Hz in time [4.21 4.71].
     Contours found: <contour.0516> <contour.0515>

     2 contours confirm
     <policecar-siren.st2.mst2.0695 [907 947] Hz>
     in time ([4.21 4.70})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#005's
     mstream at [1821 1843] Hz in time [4.21 4.71].
     Contours found: <contour.0521> <contour.0520>
                     <contour.0519>

     3 contours confirm
     <policecar-siren.st2.mst4.0697 [1821 1843] Hz>
     in time ([4.21 4.64})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#005's
     mstream at [2274 2304] Hz in time [4.21 4.71].
     Contours found: <contour.0528> <contour.0527>
                     <contour.0526> <contour.0525>

     4 contours confirm
    <policecar-siren.st2.mst5.0706 [2274 2304] Hz>
     in time ([4.22 4.67})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#005's
     mstream at [3180 3210] Hz in time [4.21 4.71].
     Contours found: <contour.0535> <contour.0534>
                     <contour.0533>

     3 contours confirm
     <policecar-siren.st2.mst6.0712 [3180 3210] Hz>
     in time ([4.22 4.59})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#005's
     mstream at [3634 3680] Hz in time [4.21 4.71].
     Contours found: <contour.0540> <contour.0539>

     2 contours confirm
     <policecar-siren.st2.mst7.0719 [3634 3680] Hz>
```

```
     in time ([4.32 4.53})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#005's
     mstream at [446 468] Hz in time [4.21 4.71].
     Contours found: <contour.0545> <contour.0544>
                     <contour.0543> <contour.0542>
                     <contour.0541>

     5 contours confirm
     <policecar-siren.st2.mst1.0724 [446 468] Hz>
     in time ([4.21 4.64})

**** Attempting to confirm POLICECAR-SIREN-STREAM2#005's
     mstream at [1360 1406] Hz in time [4.21 4.71].
     Contour found: <contour.0556> <contour.0555>
                    <contour.0554>

     3 contours confirm
     <policecar-siren.st2.mst3.0726 [1360 1406] Hz>
     in time ([4.21 4.51} [4.58 4.67})

**** Trying to solve negative evidence in
     <policecar-siren.st1.mst2.0659 [1110 1150] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st1.mst2.0659 [1110 1150] Hz>)
    REGION: T:[3.72 3.86] F:[1086 1174] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-PEAK-THRESHOLDING MS-ENERGY-THRESHOLDING)



**** Searching for reprocessing plans for
     (MS-PEAK-THRESHOLDING MS-ENERGY-THRESHOLDING)
      with the goals
     ((HAVE-HYPOTHESES-SUPPORT
         (<policecar-siren.st1.mst2.0659 [1110 1150] Hz>)))


**** Reprocessing started in frequency [1086 1174]
     during time period [3.74 3.85] for Discr. Diag.
     <FRONT-END.0006> with
     *STFT-ABSOLUTE-NOISE-THRESHOLD* = 0.00162363S0
     *NUM-PEAKS-STFT-SPECTRUM* = 1
```

```
     *STFT-CONTOUR-FREQ-RADIUS* = (40 . 40)
     *PEAK-NEIGHBOURHOOD* = 3
     in effect.
     Contours found!
     Synthesizing the following mstreams: {
     (<policecar-siren.st1.mst2.0659 [1110 1150] Hz>)}
     Reprocessing completed. Global context restored.


**** Trying to solve negative evidence in
     <policecar-siren.st2.mst7.0721 [3634 3680] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st2.mst7.0721 [3634 3680] Hz>)
    REGION: T:[4.51 4.72] F:[3610 3704] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-ENERGY-THRESHOLDING)



**** Searching for reprocessing plans for
     <EXPLANATION (MS-ENERGY-THRESHOLDING)> with the goals
     ((HAVE-HYPOTHESES-SUPPORT
        (<policecar-siren.st2.mst7.0721 [3634 3680] Hz>)))


**** Reprocessing started in frequency [3610 3704]
     during time period [4.53 4.71] for Discr. Diag.
     <FRONT-END.0007> with
     *STFT-ABSOLUTE-NOISE-THRESHOLD* = 0.0261603S0
     *NUM-PEAKS-STFT-SPECTRUM* = 1
     *STFT-CONTOUR-FREQ-RADIUS* = (46 . 46)
     *STFT-CONTOUR-ENERGY-RADIUS* = (0.5 . 0.5)
     *PEAK-NEIGHBOURHOOD* = 4
     in effect.
     No microstreams found.
     Reprocessing completed. Global context restored.


**** Trying to solve negative evidence in
     <policecar-siren.st2.mst6.0745 [3180 3210] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st2.mst6.0745 [3180 3210] Hz>)
```

```
    REGION: T:[4.58 4.72] F:[3156 3234] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-PEAK-THRESHOLDING)



**** Searching for reprocessing plans for
     <EXPLANATION (MS-PEAK-THRESHOLDING)> with the goals
     ((HAVE-HYPOTHESES-SUPPORT
        (<policecar-siren.st2.mst6.0745 [3180 3210] Hz>)))


**** Reprocessing started in frequency [3156 3234]
     during time period [4.58 4.72] for Discr. Diag.
     <FRONT-END.0008> with
     *NUM-PEAKS-STFT-SPECTRUM* = 1
     *STFT-CONTOUR-FREQ-RADIUS* = (30 . 30)
     *PEAK-NEIGHBOURHOOD* = 3
     in effect.
     Contours found! due to
     Synthesizing the following mstreams: {
     (<policecar-siren.st2.mst6.0745 [3180 3210] Hz>)}
     Reprocessing completed. Global context restored.


**** Trying to solve negative evidence in
     <policecar-siren.st2.mst7.0736 [3634 3680] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<policecar-siren.st2.mst7.0736 [3634 3680] Hz>)
    REGION: T:[4.19 4.34] F:[3610 3704] E:[0 10000.00]

    Discrepancy Explanation Proposed:
        (MS-ENERGY-THRESHOLDING)



**** Searching for reprocessing plans for
     <EXPLANATION (MS-ENERGY-THRESHOLDING)> with the goals
     ((HAVE-HYPOTHESES-SUPPORT
        (<policecar-siren.st2.mst7.0736 [3634 3680] Hz>)))


**** Reprocessing started in frequency [3610 3704]
```

```
     during time period [4.21 4.32] for Discr. Diag.
     <FRONT-END.0009> with
     *STFT-ABSOLUTE-NOISE-THRESHOLD* = 0.0261603S0
     *NUM-PEAKS-STFT-SPECTRUM* = 1
     *STFT-CONTOUR-FREQ-RADIUS* = (46 . 46)
     *STFT-CONTOUR-ENERGY-RADIUS* = (0.5 . 0.5)
     *PEAK-NEIGHBOURHOOD* = 4
     in effect.
     Contours found!
     Synthesizing the following mstreams: {
     (<policecar-siren.st2.mst7.0736 [3634 3680] Hz>)}
     Reprocessing completed. Global context restored.


**** Out of the possible explanations:
       WIND
     for unexplained spectral bands in time [4.00 5.00],
     the following were initially selected:
       WIND
     to initially create
       <WIND#005>


**** Attempting to confirm WIND#005's
     mstream at [525 766] Hz in time [4.67 4.93].
     Contour found: <contour.0613>

     1 contour confirms
     <wind.st1.mst1.0860 [525 766] Hz>
     in time ([4.67 4.93})

**** Attempting to confirm WIND#005's
     mstream at [525 766] Hz in time [4.37 4.67].
     ---> mstream at [525 766] Hz unconfirmed

**** Trying to solve negative evidence in
     <wind.st1.mst1.0861 [525 766] Hz>

    Performing Discrepancy Diagnosis.
    OBJECT: (<wind.st1.mst1.0861 [525 766] Hz>)
    REGION: T:[4.36 4.69] F:[509 782] E:[0 10000.00]

    Discrepancy Explanation Proposed: NIL
```

```
**** Attempting to confirm WIND#005's
     mstream at [525 766] Hz in time [0.47 4.37].
     Contours found: <contour.0621> <contour.0620>
                     <contour.0618> <contour.0617>

     Contours have incorrect energy
     ---> mstream at [525 766] Hz unconfirmed


**** Processing of [5.000 6.000} time period started.
     Explained energy ratio from previous block: 0.50S0

     DATA STREAM EXHAUSTED.

<WIND#005> dropped as answer
due to low (< 0.2) final rating.

  Results from top-level plan SOLVE-PROBLEM:
    PS-ANSWERS:  (<FIRENGINE-BELL#002>
<POLICECAR-SIREN-STREAM1#005> <POLICECAR-SIREN-STREAM2#005>
<POLICECAR-SIREN-STREAM1#004> <POLICECAR-SIREN-STREAM2#004>
<POLICECAR-SIREN-STREAM1#003> <POLICECAR-SIREN-STREAM2#003>
<POLICECAR-SIREN-STREAM1#002> <POLICECAR-SIREN-STREAM2#002>
<POLICECAR-SIREN-STREAM2#001> <POLICECAR-SIREN-STREAM1#001>)

user time    =    557.609
system time  =      3.410
Elapsed time =   0:09:36
Allocation   = 973282704 bytes
0 Page faults
; SCENARIO:2 (S02)

;*********************************************************
; START: 30 Jun 1996 12:44:00  FINISH: 30 Jun 1996 12:53:40
;*********************TEST15-max-RUN-2.TEXT*****************
;SOURCES: 12 in scenario, 41 in library
;HR:0.92  MR:0.08  FA:0  T+:0.95  T-:0.04 (Over:0.04 FA:0.0)
;TOTAL-HYPS: 16    NONANSWERS:  5    ANSWERS: 11
;spec:F+=_13291520 R+=________0 (__13291520)
;spec:Fx=_26583040 Rx=________0 (__26583040)
;REPROCESSING CONTEXTS: spec=1   peak=6   bound=1  cont=3
;DISTINCT DISCREPANCY DIAGNOSES: 6
; NIL: 34
; (end-of-data-boundary): 1 (S:1 F:0)
```

```
; (ms-energy-thresholding): 4 (S:1 F:3)
; (ms-frequency-resolution): 2 (S:0 F:2)
; (ms-peak-thresholding): 1 (S:1 F:0)
; (ms-peak-thresholding ms-energy-thresholding):2 (S:2 F:0)
;DISTINCT DIFFERENTIAL DIAGNOSES: 0 (NIL = 0)
;*************************************************************
;*    INSTANCE NAME            RAT-  NEG.   TIME     supp.*
;*                             ING   EV.    SPAN     cntxt*
;*************************************************************
;*POLICECAR-SIREN-STREAM2#004  .841 .040 [3.27 3.77]   4  *
;*POLICECAR-SIREN-STREAM1#001  .839 .066 [0.00 0.48]   6  *
;*POLICECAR-SIREN-STREAM2#003  .835 .047 [2.34 2.82]   4  *
;*POLICECAR-SIREN-STREAM2#002  .822 .033 [1.40 1.88]   4  *
;*POLICECAR-SIREN-STREAM2#005  .810 .087 [4.21 4.70]   6  *
;*POLICECAR-SIREN-STREAM2#001  .776 .126 [0.45 0.94]   4  *
;*POLICECAR-SIREN-STREAM1#003  .766 .107 [1.86 2.34]   6  *
;*POLICECAR-SIREN-STREAM1#002  .750 .134 [0.93 1.38]   6  *
;*POLICECAR-SIREN-STREAM1#005  .737 .163 [3.75 4.21]   7  *
;*FIRENGINE-BELL#001           .640 .023 ([2.26 4.99]) 20 *
;*POLICECAR-SIREN-STREAM1#004  .636 .267 [2.80 3.27]   7  *
;*WIND#001                     .156 .596 ([0.90 2.99]) 4  *
;*WIND#004                     .098 .702 ([2.82 4.99]) 2  *
;*WIND#002                     .118 .724 ([1.90 3.99]) 2  *
;*WIND#005                     .007 .000 ([4.67 4.93]) 2  *
;*WIND#003                     .000 .500 ([3.72 3.99]) 1  *
;*************************************************************
```

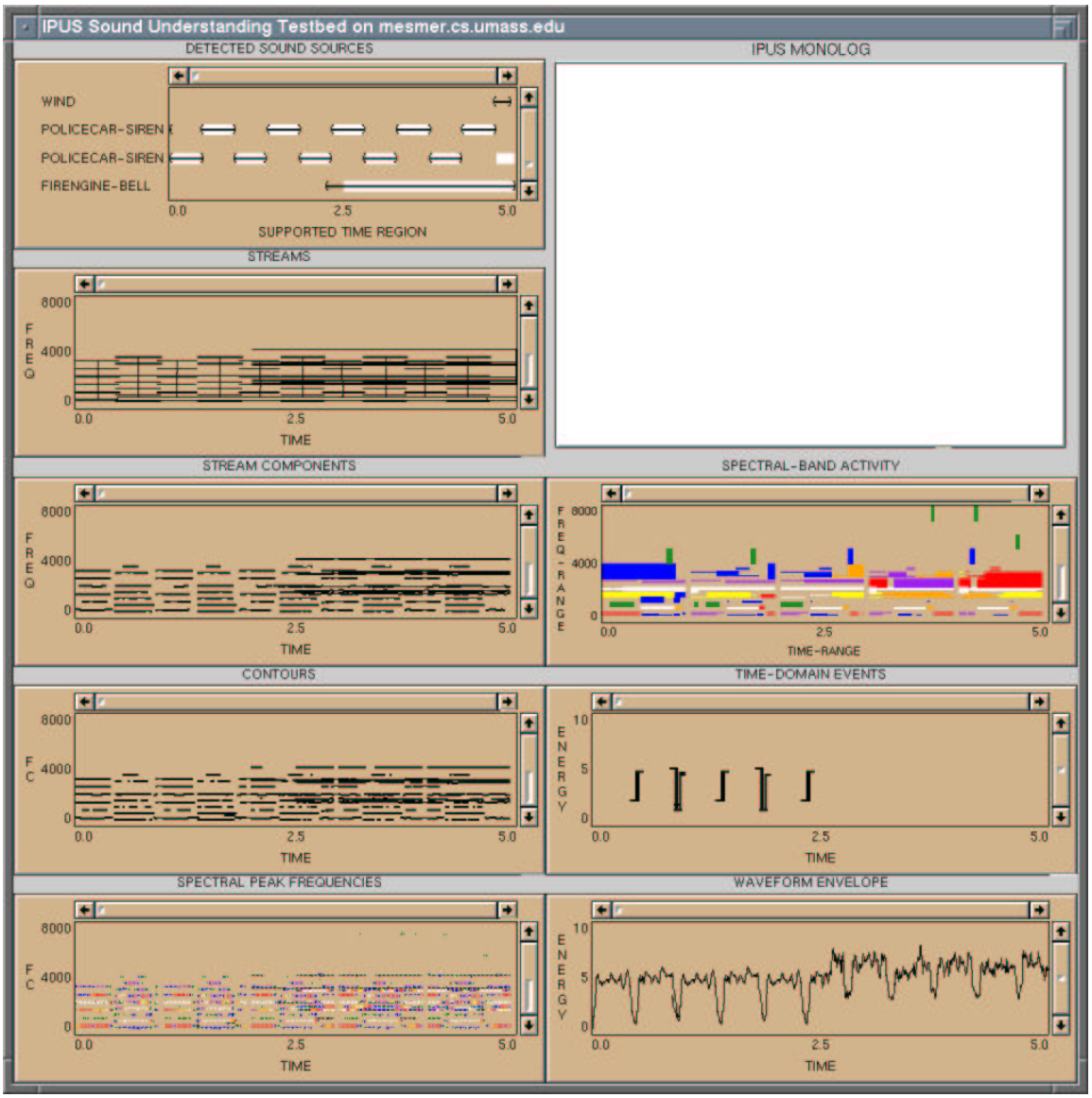


Figure B.7. Post Block-5 Status.

REFERENCES

[Auditec 1989] *"Sound Effects" Tape*, Auditec of St. Louis, Inc., 330 Selma Avenue, St. Louis, MO, 63119, copyright 1989.

[Bell and Pau, 1990a] Bell, B. and Pau, L. F., "Context knowledge and search control issues in object-oriented prolog-based image understanding," *The Proceedings of the 1990 European Conference on Artificial Intelligence*, 1990.

[Bell and Pau, 1990b] Bell, B. and Pau, L. F., "Contour tracking and corner detection in a logic programming environment," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, pp. 913–917, August 1990.

[Bobick and Bolles, 1992] Bobick, A. F. and Bolles, R. C., "The representation space paradigm of concurrent evolving object descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 146–156, February 1992.

[Brady and Wielinga, 1978] Brady, J. M. and Wielinga, B. J., "Reading the writing on the wall," in *Computer Vision Systems*, Hanson and Riseman, eds. New York: Academic Press. 1978.

[Bregman 1990] Bregman, A., *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press. 1990.

[Brown and Cooke, 1994] Brown, G., and Cooke, M., "Perceptual grouping of musical sounds: a computational model," *Journal of New Music Research*, 23(1), pp. 107–132, 1994.

[Brown and Cooke, 1992] Brown, G., and Cooke, M., "A computational model of auditory scene analysis," *Proceedings of the International Conference on Spoken Language Processing*, pp. 523-526, 1992.

[Carver and Lesser, 1993a] Carver, N. and Lesser, V., "The evolution of blackboard control," *Expert Systems with Applications*, Special Issue on The Blackboard Paradigm and Its Applications, vol. 7, no. 1, pp. 1–30, 1994.

[Carver and Lesser, 1993b] Carver, N. and Lesser, V., "A planner for the control of problem-solving systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Special Issue on Planning, Scheduling and Control, vol. 23, no. 6, pp. 1519–1536, November/December 1993.

[Carver and Lesser, 1991] Carver, N. and Lesser, V., "A new framework for sensor interpretation: planning to resolve sources of uncertainty," *The Proceedings of the 1991 National Conference on Artificial Intelligence (AAAI-91)*, pp. 724–731, Anaheim, California, July 1991.

[Cooke and Brown, 1994] Cooke, M., and Brown, G., "Computational auditory scene analysis: exploiting principles of perceived continuity," *Speech Communication*, Jan/Feb 1994.

[Cooke *et al.*, 1993] Cooke, M., Brown, G. J., Crawford, M., and Green, P. D., "Computational auditory scene analysis: listening to several things at once," Endeavor 17(4), pp. 186–190, 1993.

[Dawant 1991] Dawant, B. and Jansen, B., "Coupling numerical and symbolic methods for signal interpretation," *IEEE Transactions on Systems, Man and Cybernetics*, pp. 115–124, Jan/Feb 1991.

[Decker *et al.*, 1990] Decker, K., Lesser, V., and Whitehair, R., "Extending a blackboard architecture for approximate processing," *Journal of Real-Time Systems*, 2, pp 47–79, Kluwer Academic Publishers, 1990.

[De Mori *et al.*, 1987] De Mori, R., Lam, L., and Gilloux, M.,"Learning and plan refinement in a knowledge-based system for automatic speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 289–305, February 1987.

[Dorken 1994] Dorken, E., "Approximate processing and knowledge-based reprocessing of non-stationary signals," Ph.D. thesis, Department of Electrical and Computer Engineering, Boston University, 1994.

[Dorken *et al.*, 1992] Dorken, E., Nawab, S. H., and Lesser, V., "Extended model variety analysis for integrated processing and understanding of signals," *The Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. V, pp. 73-76, San Francisco, California, March 1992.

[Draper, 1993] Draper, B., "Learning object recognition strategies," Ph.D. thesis, Department of Computer Science, University of Massachusetts, Amherst, MA, 1993.

[Ellis, 1996] Ellis, D. P. W., "Prediction-driven computational auditory scene analysis," Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1996.

[Erman *et al.*, 1980] Erman, L., Hayes-Roth, F., Lesser, V., Reddy, D., "The hearsay II speech understanding system: integrating knowledge to resolve uncertainty," *Computing Surveys 12*, vol. 2, pp. 213–253, June 1980.

[Gabor 1946] Gabor, D., "Theory of communication," *Journal of the Institute of Electrical Engineering*, vol. 93, pp. 429–441, 1946.

[Giannesini *et al.*, 1986] Giannesini, F., Kanoui, H., Pasero, R., and van Caneghem, M., *Prolog*, Reading, MA: Addison Wesley, 1986.

[Hayes-Roth *et al.*, 1989] Hayes-Roth, B., Washington, R., Hewett, R., Hewett, M., and Seiver, A., "Intelligent monitoring and control," *The Proceedings of the 1989 International Joint Conference on Artificial Intelligence*, pp. 243–249, Detroit, Michigan, August 1989.

[Helmholtz 1885] Helmholtz, H. von, *On the sensations of tone as a physiological basis for the theory of music*, (Second English edition; translated by A. J. Ellis, 1885). Reprinted by Dover Publications, 1954.

[Hudlická and Lesser, 1987] Hudlická, E. and Lesser, V., "Modeling and diagnosing problem-solving system behavior," *IEEE Transactions on Systems, Man and Cybernetics*, Special Issue on Diagnostic Reasoning, vol. 17, no. 3, pp. 407–419, May/June 1987.

[Hudlická and Lesser, 1984] Hudlická, E. and Lesser, V., "Meta-level control through fault detection and diagnosis," *The Proceedings of the 1984 National Conference on Artificial Intelligence (AAAI-84)*, pp. 153–161, Austin, Texas, July 1984.

[Jones and Parks, 1990] Jones, D. L. and Parks, T. W., "A high resolution data-adaptive time-frequency representation," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 12, pp. 2127–2135, December 1990.

[Kashino and Tanaka, 1993] Kashino, K., and Tanaka, H., "A sound source separation system with the ability of automatic tone modeling," *Proceedings of the 1993 International Computer Music Conference*, pp. 248–255, 1993.

[Kohl *et al.*, 1987] Kohl, C., Hanson, A., and Reisman, E., "A goal-directed intermediate level executive for image interpretation," *The Proceedings of the 1987 International Joint Conference on Artificial Intelligence*, pp. 811–814, Milan, Italy, August 1987.

[Lee *et al.*, 1990] Lee, K., Hon, H., and Reddy, R., "An overview of the SPHINX speech recognition system," in *Readings in Speech Recognition*, Morgan Kaufmann Publishers, Inc., San Mateo, CA; Waibel, A., and Lee, K., eds., pp. 600–610, 1990.

[Lesser *et al.*, 1995] Lesser, V., Nawab, S. H., and Klassner, F., "IPUS: an architecture for the integrated processing and understanding of signals," *Artificial Intelligence*, vol. 77, no. 1, pp. 129–171, August 1995.

[Lesser *et al.*, 1993] Lesser, V., Nawab, S. H., Gallastegi, I., and Klassner, F., "IPUS: an architecture for integrated signal processing and signal interpretation in complex environments," *The Proceedings of the 1993 National Conference on Artificial Intelligence (AAAI-93)*, pp. 249–255, Washington, DC, July 1993.

[Lesser *et al.*, 1977] Lesser, V., Hayes-Roth, F., Birnbaum, M., and Cronk, R., "Selection of word islands in the hearsay-II speech understanding system," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 791-794, 1977.

[Lowerre and Reddy, 1980] Lowerre, B., and Reddy, R., "The HARPY speech understanding system," in *Trends in Speech Recognition*, Prentice-Hall, 1980; Speech Science Publications, Apple Valley, MN, 1986.

[Marr, 1982] Marr, D. C., *Vision*. San Francisco: W. H. Freeman & Co. 1982.

[McAdams, 1984] McAdams, S. "Spectral fusion, spectral parsing and the formation of auditory images," Ph.D. thesis, CCRMA, Stanford University, 1984.

[Mellinger, 1991] Mellinger, D. K., "Event formation and separation in musical sound," Ph.D. thesis, CCRMA, Stanford University, 1991.

[Milios and Nawab, 1989] Milios, E. E. and Nawab, S. H., "Signal abstractions in signal processing software," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 6, pp. 913–928, June 1989.

[Nagao and Matsuyama, 1980] Nagao, M. and Matsuyama, T. *A Structural Analysis of Complex Aerial Photographs.* N.Y.: Plenum Press, 1980.

[Nakatani *et al.*, 1995] Nakatani, T., Okuno, H. G., and Kawabata, T., "Residue-driven architecture for computational auditory scene analysis," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 1, pp. 165–172, Montreal, Canada, August 20–25, 1995.

[Nawab *et al.*, 1995] Nawab, S. H., Espy-Wilson, C., Mani, R., and Bitar, N. N., "Knowledge-based analysis of speech mixed with sporadic environmental sounds," to appear in *Computational Auditory Scene Analysis*, Erlbaum Publishers, 1996.

[Nawab and Dorken, 1995] Nawab, S. H. and Dorken, E., "Efficient STFT computation using a quantization and differencing method," *The Proceedings of the 1993 IEEE Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 587–590, Minneapolis, Minnesota, April 1993.

[Nawab and Quatieri, 1988] Nawab, S. H. and Quatieri, T., "Short-time fourier transform," *Advanced Topics in Signal Processing*, Prentice Hall, New Jersey, 1988.

[Nawab *et al.*, 1987] Nawab, S. H., Lesser, V., and Milios, E., "Diagnosis using the underlying theory of a signal processing system," *IEEE Transactions on Systems, Man and Cybernetics*, Special Issue on Diagnostic Reasoning, vol. 17, no. 3, pp. 369-379, May/June 1987.

[Newell and Simon, 1969] Newell, A. and Simon, H., "GPS: a program that simulates human thought," *Computers and Thought*, Feigenbaum and Feldman, eds., McGraw-Hill, pp. 279–293, 1969.

[Nii, 1986] Nii, P. H., "The blackboard model of problem solving," *AI Magazine*, vol. 7, no. 2, pp. 38–53, 1986.

[Oppenheim and Schafer, 1989] Oppenheim, A. V. and Schafer, R. W., *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[Parsons 1976] Parsons, T. W., "Separation of speech from interfering speech by means of harmonic selection," *Journal of the Acoustical Society of America*, vol. 60, pp. 911–918, 1976.

[Peng and Reggia, 1986] Peng, Y. and Reggia, J. "Plausibility of diagnostic hypotheses: the nature of simplicity," *The Proceedings of the 1986 National Conference on Artificial Intelligence (AAAI-86)*, pp. 140–145, Philadelphia, Pennsylvania, July 1986.

[Rioul and Vetterli, 1991] Rioul, O., and Vetterli, M., "Wavelets and signal processing," *IEEE Signal Processing Magazine*, pp. 14–38, October 1991.

[Seborg 1986] Seborg, D., "Adaptive control strategies for process control: a survey," *AIChE Journal*, vol. 32, no. 6, pp. 881–913, June 1986.

[Strat 1991] Strat, T. M., "Natural object recognition," Ph.D. thesis, Department of Computer Science, Stanford University, Aug. 1991.

[Swain and Stricker, 1991] Swain, M. and Stricker, M., eds. *Promising Directions in Active Vision*, NSF Active Vision Workshop, Technical Report CS 91-27, Computer Science Department, University of Chicago, 1991.

[Warren, 1984] Warren, R. M., "Perceptual restoration of obliterated sounds," *Psychological Bulletin* vol. 96, pp. 371–383, 1984.

[Weintraub 1986] Weintraub, M., "A computational model for separating two simultaneous talkers," *Proceedings of the 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, vol. 3, pp. 1–4, April 1986.