IPUS: An Architecture for the Integrated Processing and Understanding of Signals[1,2]

Victor Lesser   Hamid Nawab[†]   Frank Klassner

Computer Science Technical Report 93-29
April 1993

Computer Science Department	[†]ECS Department
University of Massachusetts	Boston University
Amherst, Massachusetts 01003	Boston, MA 02125
{*lesser klassner*} *@cs.umass.edu*	*hamid@buengc.bu.edu*

Abstract

The *Integrated Processing and Understanding of Signals* (IPUS) architecture is presented as a framework that exploits formal signal processing models to structure the bi-directional interaction between front-end signal processing and signal understanding processes. This architecture is appropriate for complex environments, which are characterized by variable signal to noise ratios, unpredictable source behaviors, and co-occuring objects whose signal signatures can distort each other. A key aspect of this architecture is that front-end signal processing is dynamically modifiable in response to scenario changes and to the need to *re-analyze* ambiguous or distorted data. This architecture tightly integrates the search for the appropriate front-end signal processing configuration with the search for plausible interpretations. In our opinion, this dual search, informed by formal signal processing theory, is a necessary component of perceptual systems for intelligent agents that must interact with complex environments. In order to explain this architecture in detail, an example of its use in an implemented system for acoustic signal interpretation is presented.

1 Introduction

Since the early 1970's, a major focus in the design of perceptual architectures has been to organize and exploit sufficient knowledge in order to recover from uncertainty introduced by numerical front-end signal processing algorithms (SPAs). One can categorize research efforts in architectural design along five dimensions based on where they emphasize the placement and organization of this knowledge:

1. within high-level interpretation knowledge sources (HLKSs) (e.g., as improved or approximate models of environmental phenomena [14, 15, 24, 35]),

2. within numeric-level KSs (SPAs) (e.g., as control parameter optimization processes or feedback loops [6, 17, 33]),

3. in the control of HLKSs' application (e.g., in planning architectures for controlling KS activation and sophisticated evidential representations [4, 7, 8, 15, 30]),

4. in the control of SPAs' application (e.g., as differential diagnosis rules for SPA application to disambiguate objects in the environment [11, 12, 22]), and

5. in the control of the interaction between HLKSs and SPAs [1, 2, 11, 12, 22, 18].

Each of the first four areas has been the scene of fruitful research over the past two decades, yielding significant architectural paradigms. However, we believe that some of the assumptions made in these efforts have resulted in a paradigm not well suited to the perception of complex environments. Such environments are characterized by variable signal-to-noise ratios, unpredictable source behavior, and the simultaneous occurrence of objects whose signal signatures can mask or otherwise distort each other.

Consider the architectural paradigm in Figure 1, which has usually been assumed by research efforts in the first four areas. It assumes that fixed signal processing in the front-end can provide adequate (not necessarily optimal) evidence for reliable interpretations regardless of the range of possible scenarios in the environment. In our opinion, this assumption is plausible for architectures that monitor stable environments, but not for those that monitor complex environments. In these environments, the choice of front-end SPAs is crucial to the generation of adequate evidence for interpretation processes. Parameter values inappropriate
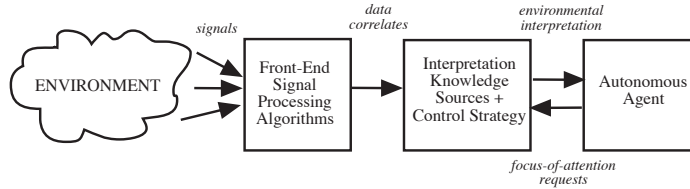
1

Figure 1: *Classic Knowledge-Based Signal Processing Architecture. This paradigm imposes a unidirectional control flow that limits interpretation processes' analysis to only the single set of observations afforded by the fixed signal processing. 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.*

to the current scenario can render a perceptual system unable to interpret entire classes of environmental events correctly. Front-end SPA sets for complex environments must be dynamically modifiable to respond to scenario changes and to reprocess ambiguous or distorted data. "Dynamically modifiable" refers both to the ability to change SPA control parameter values and to the ability to select entirely new sets of front-end SPAs.

Figure 2 illustrates the utility of dynamically modifiable SPAs to interpret a complex acoustic environment. Figure 2a shows the frequency tracks of four sound sources as they would appear if they were processed with Short-Time Fourier Transform [27] (STFT) SPAs appropriate for each portion of the scenario. Figure 2b shows how the tracks appear when the entire scenario is processed by one STFT SPA appropriate only for the steady-state portion of the last sound in the scenario. Due to inappropriate processing, the first two seconds' analyses contain several distortions that would lead to ambiguous interpretations and completely undetected sources (see Figure 2's caption).

These observations have led us to focus our work in the fifth knowledge area, controlling HLKS/SPA interaction. Since the late 1980's, there have been several efforts to design architectures allowing interpretation processes to reconfigure signal processing. However, these architectures' processing/interpretation interactions have tended to be informal and domain-specific (see Section 5).

In this paper we present the *Integrated Processing and Understanding of Signals* (IPUS) architecture as a formal and domain-independent framework for structuring HLKS/SPA interaction in complex environments [21, 23, 26]. The framework 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. The architecture enforces structured, bi-directional interaction
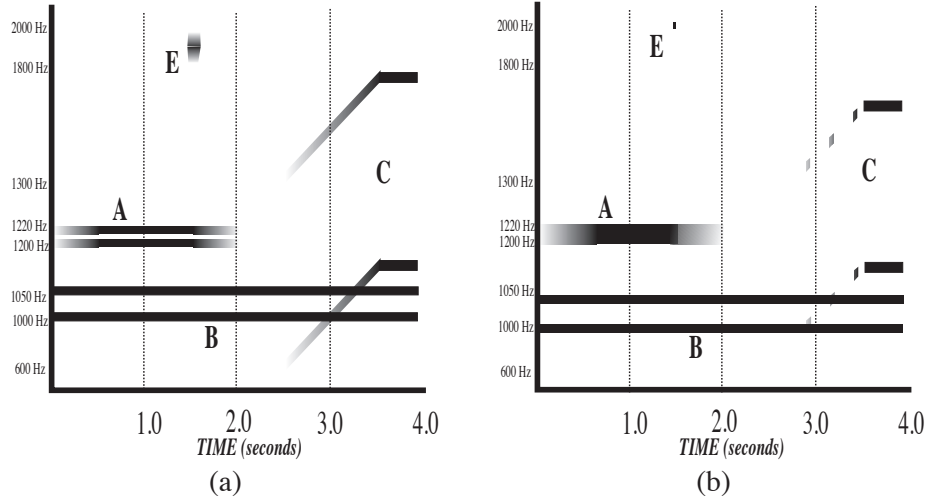
2

Figure 2: *Figure 2b shows distortions introduced by a STFT SPA with inappropriate parameter settings applied to the acoustic scenario described in Figure 2a. Darker shading indicates higher energy. The STFT parameter settings used throughout 2b were FFT-SIZE: 1024, STFT-WINDOW: 1024, and DECIMATION: 1024. The signal was sampled at 10KHz. DECIMATION is the position of the next analysis window from the start of the previous one. In this experiment the value was set to 1024 to permit the fastest possible processing of the data. In 2b's first second, the tracks from source A are merged due to the SPA's frequency resolution, which, although appropriate for source C's tracks, is not appropriate for source A's. Source E's frequency track is not even detected in 2b's next second because the SPA analysis windows don't provide adequate time resolution to isolate the source's spectral features.*

between the interpretational components and the signal processing components of an agent's perceptual system. This structured interaction is used to

- detect discrepancies between signal expectations and actual signal observations,

- diagnose these discrepancies and ascribe reasons for observational uncertainty,

- determine reprocessing strategies for uncertain data and expected scenario changes, and

- determine differential diagnosis strategies to disambiguate data with several alternative interpretations.

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.

We are currently applying IPUS to acoustic and radar domain problems. This paper discusses the generic architecture and its instantiation for acoustic signal interpretation. Acoustic signal interpretation in itself is an interesting problem that arises in applications such as assistive devices for the hearing impaired and robotic audition.[3]

In this paper we (1) discuss perception in complex environments (2) present motivations for the IPUS framework, (3) describe the generic IPUS architecture, (4) discuss related work, (5) describe an IPUS-based acoustic interpretation testbed, (6) illustrate the testbed's behavior using Figure 2's scenario, and (7) discuss architectural benefits.

2   Perception in Complex Environments

In this section we discuss the nature of perception in complex environments and the means by which these environments are perceived. In particular, we establish terminology for describing environments and for discussing context-dependent suitability of SPAs. We represent environments using the following definitions.

Definition 1 (Environment) *An environment is a triple $(\mathcal{O}, \mathcal{F}, \mathcal{R})$ where $\mathcal{O}$ is the set of observable objects, $\mathcal{F}$ is the set of all features that can be used to describe objects, and $\mathcal{R}$ is a set of context rules describing how features interact with each other when more than one object is being perceived in the environment.*

Definition 2 (Objects) *Each object belongs to a unique object class. Object classes are defined by sets of feature descriptions. Each set specifies a subset of features from $\mathcal{F}$ and ranges of permissible values for these features. An object is an instance of an object class if its feature values lie within a descriptor set of the class.*

Definition 3 (Contexts) *A context is the set of all specific objects, with their orientation, observed in an environment. A permissible context is defined as a set of objects which are permitted to co-occur. Unless otherwise proscribed by the specific application domain, a permissible context may contain several instances of the same object class.*

In audition, the orientation of an object includes domain-dependent characterizations such as distance, loudness, and velocity. In another domain such as vision, orientation would include characterizations such as pose, distance, and velocity.

---

[3]The problem of identifying and tracking sounds not limited to speech.

4

Definition 4 (Rules) *A context rule is a pair (C,F). C is a permissible context and $F \subseteq f_{obj} \times f_{env}$. Here f_{obj} is the union of instantiated features from all the objects in C, and f_{env} is a powerset of $\mathcal{F}$ with instantiated values. The set F indicates the observability of the objects' instantiated features when they are considered in the context C. Elements in F of the form $\{f_1, \{f_1\}\}$ indicate the instantiated feature f_1 is observable in the context; elements of the form $\{f_1, \{g_1, \ldots, g_n\}\}$ indicate the instantiated feature is masked or otherwise distorted to appear as different instantiated feature(s) $\{g_1, \ldots, g_n\}$ from f_{env}. Note that f_x indicates a feature and its particular value.*

The rules indicate how the features of co-occurring objects interact with each other without regard to how their signals are processed. For example, such rules from vision would address the occlusion of objects by other objects, while such rules from audition would address the summed-energy of overlapping frequency components from multiple sounds. Definition 4 describes only the *kind* (not the *form*) of knowledge that perceptual systems should have about contexts. The definition's knowledge representation is combinatorially explosive and certainly could not be used in any real system.

Having defined the concept of an agent's environment, let us now consider SPAs, the means by which an agent processes the signals from its environment. There are two levels of abstraction for describing SPAs: generic SPAs and SPA instances. SPA instances are specified by specific values for a generic SPA's control parameters. Where there is no ambiguity in the discussion between generic SPAs and SPA instances, we will use the term "SPA" to refer to an SPA instance. When applied to signals, SPAs produce *correlates*. These are used as evidence to support hypotheses that particular features (not necessarily associated with any object) are present in the environment. We refer to the correlate set produced by an SPA as that SPA's *computed correlate set.*

An SPA's parameter values induce capabilities or limitations with respect to the scenario being monitored. Consider the generic Short-Time Fourier Transform (STFT) algorithm [27] in the acoustic domain. An STFT instance has particular values for its parameters, such as analysis window length, frequency-sampling rate, and decimation interval (separation between consecutive analysis window positions). Depending on assumptions about a scenario's spectral features and their time-variant nature, these parameter values increase or decrease the instance's usefulness in monitoring the scenario. An instance with a large window length will provide fine frequency resolution for scenarios containing sounds ("acoustic objects") with time-invariant components, but at the cost of poor time resolution

for sounds with time-varying components.[4]

In complex environments, there are often many SPAs which can potentially compute a correlate's value. The effectiveness of an SPA to produce correlates that can support hypothesized object features is dependent in general upon the context in which the correlates are to be computed, the specific values of the object features, and the SPA's parameter values. We will consider an SPA's parameter values appropriate to a context if the SPA's correlates can provide not just support, but *unambiguous* support for all the features of all the objects in the context.

Figure 3 uses sound disambiguation to illustrate the relationship between context-dependent correlate computation and interpretation ambiguity more concretely. When analyzed in isolation, the hairdryer's two frequency tracks are unambiguously supported by the correlates from STFT-1. However, when the hairdryer's tracks are analyzed in conjunction with the telephone in the second context, ambiguity arises. The new tracks in Figure 3b indicate the potential presence of a new sound that matches the telephone model except for its lowest frequency track. The hairdryer's lower-frequency track cannot be unambiguously supported by the same SPA's correlates, since at least some of the track's potential support could alternatively support the phone's low-frequency components. Fourier theory can attribute the ambiguity to the SPA's poor frequency resolution capabilities and indicate that the second context should be reanalyzed by a more appropriate SPA. When the second context's signal is analyzed by STFT-2, the SPA's finer resolution confirms this explanation for the ambiguity and provides correlates that unambiguously support both the hairdryer's *and* the telephone's tracks.

At this point we see that to select SPA instances appropriate to a particular scenario, a perceptual system must consider the features corresponding to the input signal. This leads to the apparent circularity that choosing appropriate SPA parameter values requires knowledge about the signal, but this knowledge can only be obtained by first processing the signal with an SPA with appropriate parameter settings. Thus, in complex environments the search for appropriate interpretations must be intimately connected with the search for appropriate SPA instances.

The features that perceptual systems can monitor in complex environments fall into two classes. The first class contains features which can be used to indicate the existence of one or more objects, though not necessarily the objects' identities. These features often have supporting correlates that can be computed independent of the context being analyzed. In the auditory domain, for example, any collection

---

[4]A variant analysis of the Heisenberg Uncertainty Principle implies that one cannot obtain a STFT SPA instance (or, for that matter, design a new algorithm class) which simultaneously provides infinite frequency resolution and infinite time resolution.
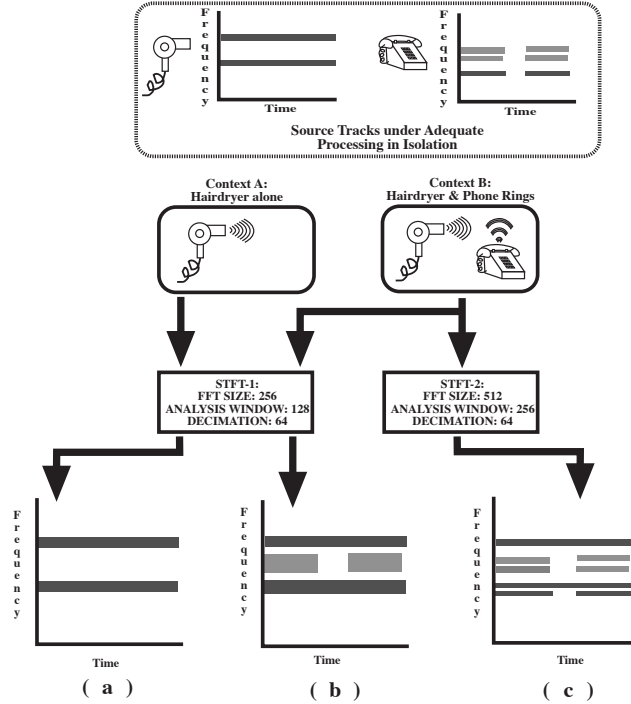
Figure 3: *Context-Dependent Correlate Computation. When STFT-1 analyzes context A's signal, its frequency correlates in (a) are adequate for unambiguously identifying the hairdryer's two frequency tracks. When the same SPA analyzes context B's signal, however, its frequency correlates in (b) are not adequate for unambiguously supporting the hairdryer's two tracks AND the phone's three tracks. Context B's signal requires processing by STFT-2 with a finer frequency resolution in order to produce correlates in (c) that unambiguously support the two sources' tracks.*

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. Such gross features' correlates can generally be computed in a context-independent manner; hence we call them *context-independent features*.

The second feature class contains those features which can be used to identify an object or track the behavioral changes of an object. The computation of correlates to support these features is often very sensitive to the context being analyzed; hence we call them *context-dependent features*. In the auditory domain, for example, a frequency track would be a context-dependent feature of a sound ("acoustic object"). If the current scenario has no sounds besides the sound containing a

7

particular track T_0, then an STFT with parameters providing only very coarse frequency resolution would still produce correlates that could support the track's existence. Now assume that the current scenario changes so that there are other sounds in the environment with frequency tracks $T_1, \ldots, T_n$. In this new scenario only STFTs providing frequency resolution of at least the minimum difference between T_0's frequency and the other tracks' frequencies would produce correlates that could unambiguously support T_0's existence.

It is important to note that the distinction between context-independent and context-dependent features lies in the features' usage. 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.

This section's discussion about complex environments and the basic means for analyzing their signals serves as background for Section 3. The focus in that section is on how a domain's signal processing theory can be used to guide the design of an architecture for controlling the process of SPA application.

3   Architectural Motivation

Past research efforts within the traditional paradigm for perceptual system design (Figure 1) have produced architectures that require the identification of a set of features and SPAs applicable to all scenarios the environment may produce. This requirement is feasible only for significantly constrained environments. Under the traditional paradigm, complex environments can require combinatorially explosive SPA sets with multiple parameter settings to capture the variety of signals adequately [13] and to handle the variety of processing goals the current scenario may dictate. As an example of variable processing goals, consider a system with the primary goal of responding to either the sounds of an infant or a ringing telephone while ignoring other sounds. This may be done by monitoring a medium-frequency band. If an infant sound is detected, the system's goal may then switch to determining whether the infant is crying or choking while ignoring telephone rings. Such a goal might then be accomplished by switching to lower-frequency spectral regions with specialized SPAs.

To circumvent the combinatorial explosion, one could reason that a small SPA set might be sufficient if comparisons could be made between the SPAs' computed correlates and dynamically-generated formal expectations. We use the term *anticipated correlate set* to refer to the set of expectations about an SPA's computed

8

correlate set. Any computed correlates whose coordinates and values do not match those of any anticipated correlates are considered unanticipated. Unmet SPA output expectations can indicate that either the expectations are based on incorrect interpretations or that 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 re-interpret the current scenario based on the SPA's correlates, while in the second case a perceptual system could reconfigure the SPA's parameters or replace it with a more appropriate SPA. The important assumption in this solution is that there is a basis for generating the expectations, detecting the unmet expectations, and deciding between the two possible classes of explanations for the unmet expectations. We argue that a domain's *formal signal processing theory* can play this role.

An SPA's correlates can be compared with expectations based on object models or on *a priori* environment constraints such as maximum bounds on sounds' rate of temporal change in frequency. Referring back to our assumption about rules for the interaction of co-occuring objects' features, these "context rules" could also provide a basis for checking SPA appropriateness. Most importantly, a domain's signal processing theory can specify how one SPA's correlates for a context-independent feature can serve as the basis of expectations for another SPA's output correlates. This specification can serve to check an SPA's appropriateness to the environment. It can also serve to decide where to selectively apply another SPA in the signal data stream to obtain correlates for context-dependent features.

Figure 4 illustrates these concepts with an example from the acoustic processing of footsteps in a noisy environment. The example uses two complementary generic SPAs: a time-domain energy tracker and an STFT. The time-domain energy tracker detects a short, uniform energy burst that should correspond to short tracks in the frequency domain, according to acoustic signal processing theory. When analyzed by STFT-1 with its wide analysis window, the footstep's impulsive energy is smoothed with surrounding noise and fails to appear as a short frequency track in the STFT's correlates. In other words, the STFT's correlates are subject to a smoothing distortion. The temporal locations and durations of the energy tracker's energy bursts serve two purposes. First, they indicate that STFT-1 was potentially inappropriate to the current environment. Second, they serve as the basis for generating STFT-2 with a narrower analysis window and smaller time decimation interval to apply to the region in the signal where a new source is suspected. This STFT's correlates not only confirm the belief that the first STFT was inappropriate to the environment but also more strongly support the existence of the impulsive footsteps than the energy tracker's correlates did by
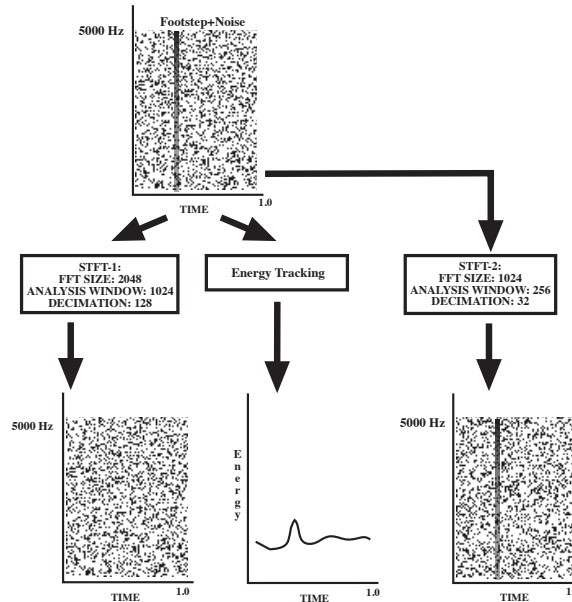
themselves.

Figure 4: *Context-Dependent Correlate Computation. The energy tracking SPA provides correlates for context-independent energy burst features. These features which guide the focused application of an STFT with parameters to find frequency-track correlates for the footstep impulse in a noisy environment.*

The preceding example provides instances of three generic roles that a domain's formal signal processing theory can play in guiding interpretation and processing in a complex environment:

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

- 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.

- 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 lead to our decision to incorporate a "discrepancy

10

detection, diagnosis, and reprocessing loop" as the backbone of the IPUS architecture. We believe that the explicit representation of the knowledge in signal processing theory is crucial to systems that monitor complex environments. Our design of IPUS is motivated by the thesis that complex environments require dynamic, context-dependent feature selection concurrent with dynamic, context-dependent selection of appropriate SPAs for extracting correlates to support the features. The goal of the framework is to use theoretical relationships between SPA parameters and SPA outputs to structure the dual searches for SPAs appropriate to a scenario and for interpretations appropriate to the SPAs' correlates.

4   Generic IPUS Architecture

This section has three parts. The first part presents a summary of the architecture. The second part discusses the generic specifications of each component of the architecture's reprocessing loop: discrepancy detection, discrepancy diagnosis, reprocessing, and differential diagnosis. The third part describes the architecture's control framework. Section 6.1 and [23] provide detailed descriptions of the algorithms used to instantiate the IPUS components in the acoustic interpretation testbed.

4.1   Architecture Summary

The generic IPUS architecture, with its primary data and control flow, appears in Figure 5a. Figure 5b shows its instantiation in the acoustic interpretation testbed to be discussed in Section 6.2. 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.

Our design of the IPUS framework assumes that signal data is submitted for analysis a block at a time. IPUS 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. 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 expecta-
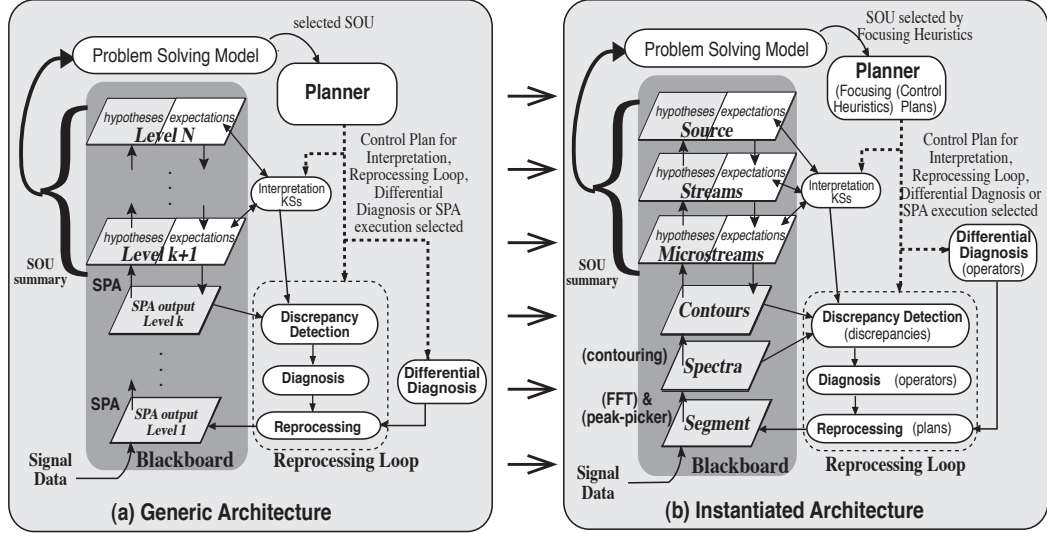
Figure 5: *Figure 5a shows the generic IPUS architecture, figure 5b shows the architecture instantiated for the sound understanding testbed. Solid arrow lines indicate dataflow relations. Dotted arrow lines indicate classes of plans that the planner can pursue when trying to reduce or eliminate particular uncertainties (discrepancies) in the problem solving model that were selected by the focusing heuristics. Knowledge added to the planner or system knowledge sources to instantiate the architecture for an application is shown in parentheses. Note that reprocessing plans can cause SPA execution at any SPA output level, not just the lowest.*

tions. Architectural control 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 then 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

12

current block's processing results 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.

4.2 IPUS Reprocessing Loop Components

This section discusses the generic specifications of each component of the architecture's reprocessing loop, as depicted in Figure 5a.

4.2.1 Discrepancy Detection

The discrepancy detection process is crucial to the IPUS architecture's iterative approach. Our specification of the process requires it to recognize three groups of discrepancies, based on the source of the anticipated correlates 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. As an example, refer to Figure 4 where 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 since Fourier theory requires the burst's presence in both analyses, given the assumption that the STFT analysis was appropriate to the context.

violation A discrepancy detected between an SPA's computed correlates and domain constraints. This class is included based on the proposition that correlates, if computed by SPAs appropriate 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 original definition. In the first case reprocessing based on the environment's definition

13

should succeed in eliminating the discrepancy. In the second case reprocessing based on the environment's (invalid) definition will fail. Failures of the second type are recorded as distortions to be expected due to environmental changes and prevent needless execution of the reprocessing loop when they are detected again.

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. "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 sinusoids at 230 Hz and at 250 Hz with no decline in their amplitudes and current STFT correlates support one or none of the sinusoids. It could indicate that possibly the STFT's energy threshold is inappropriate because the sound's volume decreased, or that a new sound is masking the expected sound.

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 or a mean-intensity value for a region in the output of an image filtering SPA is a region correlate. A track of spectral peaks over time from a series of FFT analyses is an example of a region correlate comprised of non-contiguous subsets of the SPAs' output space.

For both point and region correlates, we require that the IPUS discrepancy detection component be able to check 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 we encounter this discrepancy when an image region's hue label produced by an intensity analysis SPA is brighter than 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.

4.2.2 Discrepancy Diagnosis

A domain's formal signal processing theory can predict the form computed correlates will take not only when an SPA is applied with parameter values appropriate to the context, but also when an SPA is applied with inappropriate parameter values. We relate a signal processing theory's content to SPAs and their interaction with the environment in terms of *SPA processing models.* An SPA processing model describes how the output of the SPA changes when one of its control parameters is varied while all the others are held fixed.

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 corresponding to the STFT's WINDOW-LENGTH parameter and how this model can be used to define distortions. Refering to Figure 3, as this parameter's value increases, merged and missing correlate discrepancies disappear. Conversely, as the parameter's value decreases, merged and missing correlate discrepancies occur more frequently. Formally, assume that an STFT with an analysis window of length W is applied to a signal sampled at rate R. If the signal came from a scenario containing frequency tracks closer than R/W, Fourier theory predicts that the tracks will be merged in the STFT's computed correlates.

When discrepancies are detected, *diagnosis* can be performed to obtain an "inverse" mapping from the discrepancies and to qualitative hypotheses that explain them in terms of distortions. This diagnosis process relies on an environment's context rules and the domain's SPA processing models to define distortion processes that take place when an SPA's assumptions about its input signals are violated [28]. Note that there is a difference between discrepancies and signal distortion processes. Distortion processes are used to explain discrepancies. It is also possible for several distortion processes to explain the same kinds of discrepancies. A "low frequency resolution" process explains the 'missing' and 'unassociated' discrepancies in Figure 3's example, and a "low time resolution" process explains the 'missing' discrepancy in Figure 4's example.

As another simple diagnostic example, consider the conflict discrepancy where frequency components previously observed at 225 Hz and 250 Hz "disappear" from the current STFT output but a "new" component is observed midway between the original components' positions. The STFT processing models provide us with the concept of a "low frequency resolution" distortion process which can account for the missing and unanticipated correlates in the STFT output. In discrepancy diagnosis, this specific distortion's definition would serve as the basis for checking if it is plausible that the two components may have drifted too close to each

16

other for the current STFT instance to be able to resolve them. If this is indeed plausible, the distortion process explains the presence of just a single component in the current STFT output.

4.2.3 Reprocessing and Differential Diagnosis

The *signal reprocessing* component uses explanations from the diagnosis component 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 SPA processing models to select new SPAs and/or parameter values when instantiating the proposed reprocessing plan. Continuing the frequency resolution example from the previous subsection, the STFT processing model's quantitative relationship between parameter values and correlate output would indicate the need for a STFT instance with a longer analysis window for obtaining better frequency resolution.

In the course of processing signal data, IPUS-based systems will encounter signals that could support several alternative interpretations. In addition to natural similarities among several objects' features, ambiguous sets of alternative interpretations can also arise from co-occuring objects' interactions and from applying SPAs inappropriate to a context. The differential diagnosis component implements what we have previously referred to in Section 3 as the dynamic, context-dependent selection of features to disambiguate objects. It uses 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.

The dual search in IPUS becomes obvious with the following two observations. Each time the 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. At the same time, the distortion elimination or disambiguation measurement is predicated on the assumption that the system's current state in the interpretation space matches the actual context being observed. We will see later in Section 7.2 that failure to remove a hypothesized distortion after a bounded search in the SPA instance space will often lead to a new search in the interpretation space. This happens based on the following

reasoning. The diagnosis and reprocessing results represent an attempt to justify the assumption that the current interpretation is correct. If either diagnosis or reprocessing fails, there is a strong likelihood that the current interpretation is not correct and a new search is required in the interpretation space. Furthermore, the results of failed reprocessing can constrain the new interpretation search by eliminating from consideration objects with features requiring correlates that should have been found during the reprocessing.

4.3   Control in IPUS

Depending upon the class(es) of discrepancies detected and the context in which interpretation is being carried out, an IPUS-based system can use different strategies to resolve (i.e. explain and possibly eliminate) the discrepancies. For example, in a situation where real-time processing deadlines are tight, the system may not even attempt to resolve conflict discrepancies involving minor mismatches in order to conserve time. In a situation where time is costly but not prohibitive, however, the system may decide to engage the diagnostic process on the discrepancy, but then to forego actual reprocessing of the signal because the proffered explanation would require reprocessing a set of data too large to be accommodated by the time constraints. That is, for this case the system may decide that the successful *generation* of an explanation alone is sufficient to resolve the discrepancy. Finally, in a non-time-critical situation or when analyzing data from an important source, the system may decide to engage the diagnostic process and reprocess the data on the basis of the explanation in order to verify the explanation's plausibility as part of resolving the discrepancy.

We designed IPUS to serve as the basis of systems for producing perceptual interpretations with acceptable uncertainty levels. Therefore, we had to endow the architecture's control framework with a formalism for representing factors that affect interpretations' confidence levels. The control framework also had to support context-sensitive focusing on particular uncertainties in order to control engagement and interruption of the architecture's reprocessing loop.

For these reasons, IPUS uses the RESUN [8, 9] framework to control knowledge source (KS) execution. This framework supports the view of interpretation as a process of gathering evidence to resolve hypotheses' sources of uncertainty (SOUs). It incorporates a language for representing SOUs as structures which trigger the selection of appropriate interpretation strategies. Problem-solving is driven by information in the *problem solving model*, which is a summary of the current interpretations and the SOUs associated with each one's supporting hypotheses. An incremental, reactive planner maintains control using *control plans*

18

and *focusing heuristics*. Control plans are schemas that define the strategies and SPAs available to the system for processing and interpreting data, and for resolving interpretation uncertainties. Focusing heuristics are context-sensitive tests to select SOUs to resolve and processing strategies to pursue.

The RESUN framework endows IPUS with two basic problem-solving modes: *evidence aggregation* and *differential diagnosis*. Evidence aggregation problem solving seeks data for increasing or decreasing the certainty of one particular interpretation, whereas differential diagnosis problem solving seeks data for resolving ambiguities that produced competing interpretations. Through these problem solving approaches, IPUS-based systems can decide when to reprocess data previously examined under one SPA with another SPA to obtain evidence for resolving uncertainties.

The RESUN framework was developed to address current 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 the levels of numeric belief in hypotheses but also to the presence of specific SOUs in the problem-solving model. The control plan formalism supports opportunistic control through a refocusing mechanism that lets the planner switch among several plan elaboration points (current leaf nodes in the plan tree) in a context-dependent manner. It also permits reprocessing strategies to be expressed as alternative control plans to be selected on the basis of SOUs describing discrepancies and their explanations.

5   Related Work

The IPUS architecture explores how formal signal processing knowledge such as Fourier theory can be organized and applied in the fifth of the knowledge area discussed in Section 1. This research 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 [28] and in work on meta-level control [19, 20] that used a process of fault-detection, diagnosis, and replanning to decide the most appropriate parameters for controlling a problem-solving system.

Although we place this research most strongly in the fifth knowledge area, we feel it has implications for the other four areas as well. The architecture supports the use of an application domain's formal signal processing theory in selecting SPAs for context-dependent application to a signal. For this reason the research also extends work in the fourth area (control of SPA application) concerned with

the appropriate application of approximate or highly-specialized SPAs to specific portions of an environment's signal [25].

Several recent systems have been developed that provide for structured interaction between interpretation activity and numeric-level signal processing. All have made use of components performing at least one of the tasks of the IPUS architecture components, but none have included (and therefore none have attempted to interface) components for *all* the architecture's tasks. In this section we discuss selected frameworks as representatives of general approaches to the problem of controlling the interaction of signal processing and environmental interpretation in perceptual systems. The general approaches are described in terms of the IPUS components they functionally include.

The perceptual framework of Hayes-Roth's GUARDIAN system [18] 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 incorporates an input-data management component that controls the sampling rate of signals in response to agent 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 tasks leads to inadequate generality for the wide range of signals-environment interactions which can include signals containing complex structures that must be modeled over time in the presence of variable noise levels. Note that we are not implying that frameworks in this class do not perform any diagnostic reasoning. We are only observing that this reasoning capability is not applied to the identification of potentially adverse interactions between the environmental signal and the front-end processing.

Dawant's framework [11] 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 (regardless of whether their control parameters are variable) that are all always executed leads to systems where more and more SPAs are added to front-ends 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 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. [12] 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. No behavior supporting the expectation is observed, so diagnostic reasoning should be attempted to explain why).

The GOLDIE [22] architecture represents the set of architectures that place strong emphasis on selective SPA application without explicit guidance from formal signal-processing theory. GOLDIE 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 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.

Bell and Pau [1, 2] 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* [16] 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.

Research in active vision and robotics has recognized the importance of tracking-oriented front-end SPA reconfiguration [34], 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.

Given the adaptive nature of the IPUS architecture, it is important to distinguish between the IPUS approach and the classic adaptive control theory approach [33]. 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.

6 The IPUS Acoustic Interpretation Testbed

This section presents an acoustic interpretation testbed that we designed to experimentally examine the behavior of an IPUS-based system. The testbed runs on a TI Explorer II+ and is implemented in approximately 1400Kb of source code. All SPAs are implemented in software. Figure 5b shows the IPUS architecture's realization in this testbed. The testbed description is divided into two parts. In the first part we describe how each of the generic IPUS components was instantiated in the testbed. The second part describes the testbed's acoustic domain knowledge as background for understanding the trace in Section 7.2.

6.1 Instantiated IPUS Components

As we describe the testbed KSs, note that our KS algorithm descriptions are only intended as *instances* of algorithms that can implement the components. For example, the testbed's actual discrepancy diagnosis algorithm will be seen to be means-ends analysis using difference operators to encode the distortions implied by Fourier theory SPA processing models. Other algorithms using rules or case-based reasoning or qualitative models to apply the SPA processing models could have been used, as long as they provided the same diagnostic functionality.

6.1.1 Discrepancy Detection

The task of detecting discrepancies is distributed among all the knowledge sources responsible for interpreting correlates or lower-level interpretations as higher-level concepts. When executed, each such KS checks to see if any support is available for a higher-level concept. If none can be found, or if only partially supportive data is available, the KS will record this as a SOU (see Section 4.3) in the problem

solving model, to be resolved at the discretion of the focusing heuristics. At the end of each data block's numeric signal processing, a fault discrepancy detection KS is executed to check if SPA outputs are consistent with each other. Again, when discrepancies are found, SOUs are posted in the problem solving model. The basic SOU types defined in the RESUN framework are:

- **partial evidence** – Denotes the fact that there is incomplete evidence for the hypothesis.

- **possible alternative support** – Denotes the possibility that there may be alternative evidence that could play the same role as a current piece of support evidence.

- **possible alternative explanation** – Denotes the possibility that there may be alternative explanations for the hypothesis.

- **alternative extension** – Denotes the existence of competing, alternative versions of the same hypothesis.

- **negative evidence** – Denotes the failure to be able to produce some particular support evidence or to find any valid explanations.

In the integration of the IPUS and RESUN frameworks, an important issue is the relationship between the SOUs associated with various hypotheses and the discrepancy descriptions generated by the discrepancy detection process. Our architecture uses the following relationships:

1. **Conflict-type Discrepancies and SOU's.** Conflict-type discrepancies occur when signal processing output data does not match expectations. When an expectation is first posted, it has no supporting evidence because none has been searched for yet. To reflect this fact, the expectation is annotated with a PARTIAL SUPPORT SOU, which is a *partial evidence* type of SOU. To resolve this uncertainty, IPUS searches for evidence matching the expectations. If any portion of the expectation is unmatched after supporting evidence has been sought, a conflict discrepancy is raised for that expectation. When a conflict discrepancy is detected, a SUPPORT EXCLUSION SOU, a *negative evidence* type of SOU, is attached to the expectation.

2. **Fault-type Discrepancies and SOU's.** Fault-type discrepancies arise when two different signal processing algorithms produce conflicting hypotheses about the same underlying signal data. In such cases, a composite hypothesis is created that is a copy of the more reliable of the two data hypotheses

24

and is considered to be an extension of that hypothesis. A link labeled with a *negative evidence* SOU (in particular, a SUPPORT LIMITATION SOU, which indicates that support for a hypothesis is limited until results of further processing are obtained) connects the less reliable hypothesis to the composite hypothesis.

3. **Violation-type Discrepancies and SOU's.** A violation-type discrepancy occurs when signal processing output data violates the *a-priori* known characteristics of the entire class of possible input signals in the application domain. When such an output data hypothesis is posted on the interpretation blackboard, a *negative evidence* type of SOU is attached to it. This SOU contains a description of the violated condition.

In addition to the discrepancy detection components of the interpretation KSs (that perform conflict discrepancy detection), the testbed contains KSs for fault discrepancy detection and violation discrepancy detection.

The actual comparisons implemented in the testbed discrepancy detection components were derived from an inspection of the SPAs available to the testbed designers and the context-dependent and context-independent features these SPAs' correlates could support.

6.1.2 Discrepancy Diagnosis

The discrepancy diagnosis KS is designed to take advantage of the fact that the SPA processing models from an environment's signal processing theory can predict how SPA output will be distorted if the SPA is misapplied. Refering back to a previous example, assume that an STFT with an analysis window of length W is applied to a signal sampled at rate R. If the signal came from a scenario containing frequency tracks closer than R/W, Fourier theory predicts that the tracks will be merged in the STFT's computed correlates.

Our testbed instantiation of the diagnosis component models this knowledge in a database of formal distortion operators. When applied to an abstract description of anticipated or computed correlates, an operator returns the description modified to contain the operator's distortion. The KS uses these operators in a means-ends analysis framework incorporating multiple abstraction levels and a verification phase [28] to "explain" fault, violation, and conflict 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

25

the goal state description. Figure 6 illustrates the formal operator definition of the previously described frequency resolution distortion that the STFT SPA data correlates can be subject to, as well as its use in a short explanation.

Distortion Operator Definition

Microstream Frequency Resolution

Preconditions:
  1) N expected microstreams within a frequency region
     SAMPLE-RATE/WINDOW-LENGTH Hz wide.
  2) At most one microstream is detected in that region.

Result:
  1) Remove N microstreams, replace with one having
     energy = sum of N expected microstreams, and
     frequency-range = region in precondition 1.

Operator Application

Hz / Time

(MICROSTREAM-FREQUENCY-RESOLUTION)

DISTORTION OPERATOR LIST
(explanation)

Hz / Time

INITIAL STATE
(expected)

FINAL STATE
(observed)
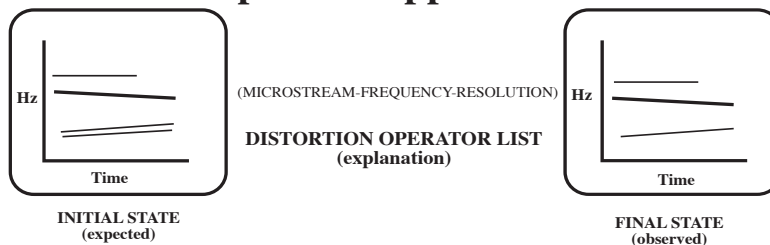
Figure 6: *Microstream Frequency Resolution Operator from the Acoustic Interpretation Testbed. When applied to a state, the operator replaces each set of expected microstreams whose members are closer than SAMPLE-RATE/WINDOW-LENGTH with a single microstream, reflecting the resolving limits associated with the current value of WINDOW-LENGTH. In the short example illustrated, this operator effectively reduces the differences between the expected state and the observed state.*

The KS's search for an explanatory distortion operator sequence is iteratively carried out using progressively more complex abstractions of the initial and goal states, until a level is reached where a sequence can be generated using no more signal information than is available at that level. Thus, the KS mimics expert diagnostic reasoning in that it offers simplest (shortest) explanations first [32]. Once a sequence is found, the KS enters its verify phase, "drops" to the lowest abstraction level, and checks that each operator's pre- and post-conditions are met when all available state information is considered. If verification succeeds, the operator sequence and a diagnosis region indicating the signal hypotheses involved in the discrepancy are returned. If it fails, the KS attempts to "patch" the sequence by finding operator subsequences that eliminate the unmet conditions and inserting them in the original sequence. If no patch is possible, and no alternative explana-

tions can be generated, the involved signal hypotheses are annotated with an SOU with a very negative rating. Figure 7 outlines the plan-and-verify strategy of the diagnostic process.
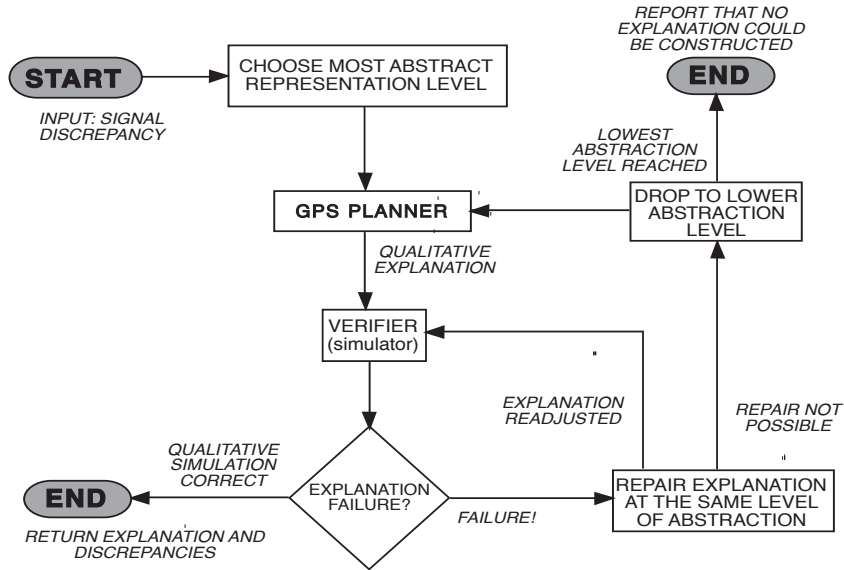

Figure 7: *The plan-and-verify strategy of the IPUS discrepancy diagnosis knowledge source.*

An issue not addressed in earlier work [28] that arose in the development of IPUS is the problem of inapplicable explanations. Sometimes the first explanation offered by the KS will not enable the reprocessing mechanism to eliminate a discrepancy. In these cases, the architecture's control framework (expressed as control plans) permits reactivation of the diagnostic KS with the previous explanation supplied as one that must not be returned again. To avoid repetition of the search performed for the previous explanation, the KS stores with its explanations the search-tree context it was in when the explanation was produced. The KS's search for a new explanation begins from that point.

The discrepancy diagnosis KS's output is also used 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 complete distortion-operator sequence is generated, each operator's support-specification is conjunctively combined to form a single specification that is used to annotate the expectation units for the hypotheses

27

involved in the original discrepancy. This annotation serves to locally reduce for those feature hypotheses the high quality-level usually required by the system for all evidence for any feature expectation. The specification indicates to the system that when it is seeking data correlates from an SPA X for object features which were previously distorted by X, it can use data correlates which match the specification's distortions *without* raising a discrepancy.

6.1.3 Signal Reprocessing

Once the distortions have been hypothesized by the discrepancy diagnostic reasoning process, the next task is to search for the appropriate SPAs and control parameter settings under which signal reprocessing may remove those distortions. Figure 8 illustrates the organization of the reprocessing knowledge source used in the testbed. This reprocessing portion of the architecture consists of the following major components: *situation assessment, reprocessing-plan selection, and reprocessing-plan execution.* The input to the reprocessing knowledge source includes a description of the input and output signal states (see diagnostic reasoning section above), the distortion operator sequence hypothesized by the diagnosis stage, and a description of the discrepancies present between the input and output signal states. The situation assessment phase uses case-based reasoning to generate multiple reprocessing plans, each of which has the potential of eliminating the hypothesized distortions present in the current situation. Plans for eliminating various categories of distortions are stored in a knowledge base. Figure 9 shows the definition for one reprocessing plan schema from our acoustic interpretation testbed. This reprocessing plan's role is to extract a short high-energy contour which was missed by the front-end STFT instance but whose presence was indicated by the front-end's time-domain energy tracker.

From the retrieved set of applicable plans, one is selected during the plan-selection stage. Selections are governed by "cost" criteria such as plan execution time. The execution of a reprocessing plan consists of incrementally adjusting the SPA control parameters, applying the SPA 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 often at least partially qualitative, and therefore it is generally impossible 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. Currently there are two independent criteria for determining plan failure in IPUS. The first criterion simply considers the number of plan iterations. If the number surpasses a fixed threshold,
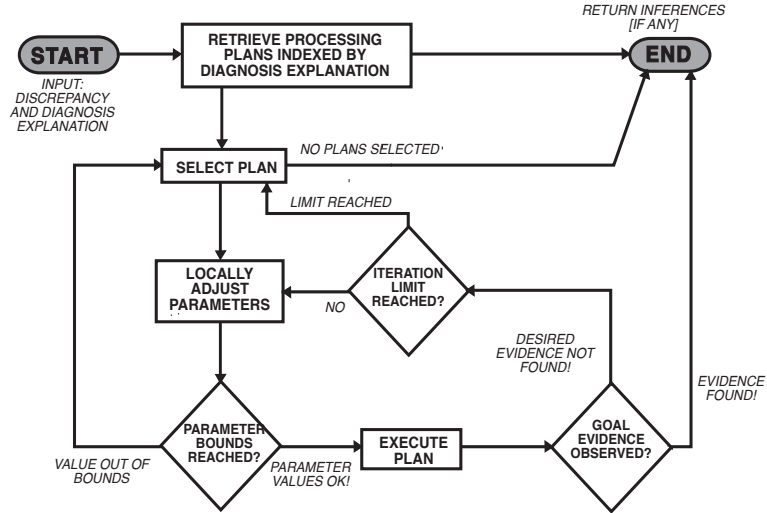
Figure 8: *The IPUS reprocessing knowledge source's framework*

failure is indicated automatically. The second criterion relies on fixed lower and upper bounds for signal processing parameters. If a plan reiteration requires a parameter value outside of its prespecified range, the plan is considered to have failed.

When failure is indicated, the discrepancy diagnosis process can be re-invoked to produce an alternative explanation for the distortions present in the original signal data. If no alternative explanation is available (i.e., the diagnostic knowledge source fails to find another distortion operator sequence), an IPUS-based system annotates the hypothesized features involved in the discrepancy with SOUs indicating low confidence due to unresolvable discrepancies. These SOUs' effects on the features' confidence levels are then propagated to object interpretations based on those features, causing their existence to be disbelieved more strongly.

6.1.4 Differential Diagnosis

In the course of processing signal data, IPUS-based systems will encounter signals that could support several alternative interpretations. We include the differential diagnosis KS to produce reprocessing plans that will enable the system to prune the interpretation search space when ambiguous data correlates are encountered. Its input is the ambiguous data's set of alternative interpretations, and its output is a triple containing:

1. the time region in the signal data to be reprocessed

29

```
(CONTOUR-1
  (state :name faulty
         :hyp-type contour
         :hyp =x)
  (state :name faultless
         :hyp-type contour
         :hyp =y)
  (operator-sequence (STFT-TIME-RESOLUTION))
  (discrepancy
         :type fault
         :name MISSING-STFT-CONTOUR-PRESENT-TD-CONTOUR
         :level contour
         :duration =x1
         :energy =x2
         :frequency =x3
         :expected-region =z)
  --> (reprocessing-plans
         ((reprocessing-plan
             :input-variables (:faulty-hyp =x
                               :faultless-hyp =y
                               :expected-region =z)
             :parameters (*STFT-OVERLAP*
                          *WINDOW-LENGTH*
                          *STFT-PEAK-ENERGY-THRESHOLD*)
             :parameter-changes
               ((lambda (p) (/ p 8))
                (lambda (p) (/ p 4))
                (lambda (p) 0.9))
             :primitive-plans (delete-all-reprocessing-units
                               reprocess-spectra-for-contours
                               reprocess-contours)
             :goal-condition (contours-present?)))))
```

Figure 9: *The definition for a reprocessing plan from the acoustic interpretation testbed to handle the distortion-operator sequence (CONTOUR-TIME-RESOLUTION). The plan specifies that on each iteration of the primitive plan list, the STFT-OVERLAP and WINDOW-LENGTH parameter values are divided by 8 and 4, respectively, while the STFT-PEAK-ENERGY-THRESHOLD parameter value is maintained at 0.9. At the end of each iteration, the goal-condition CONTOURS-PRESENT? is tested for. This goal requires that the sought high-energy contour appear.*

2. the support evidence (verification goals) that must be found for each interpretation

3. the set of reprocessing plans and parameter values proposed for revealing the desired support evidence.

Our implementation of this KS uses the following strategy. The KS first compares the interpretation hypotheses to determine their overlapping regions. Any observed evidence in these regions is labeled "ambiguous". The KS then determines the hypotheses' discriminating regions (e.g., Hyp1, 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. Figure 10 depicts the differential diagnosis KS's execution for the ambiguous data observed in block 1 of the trace (see page 38).

We should note that the explanatory reasoning performed in the differential diagnosis KS for missing evidence is primitive compared to that available in the discrepancy diagnosis KS; there is no explicit modeling of distortions via formal operators, nor is there a rich set of explanations available. Only simple distortions
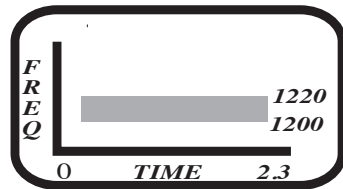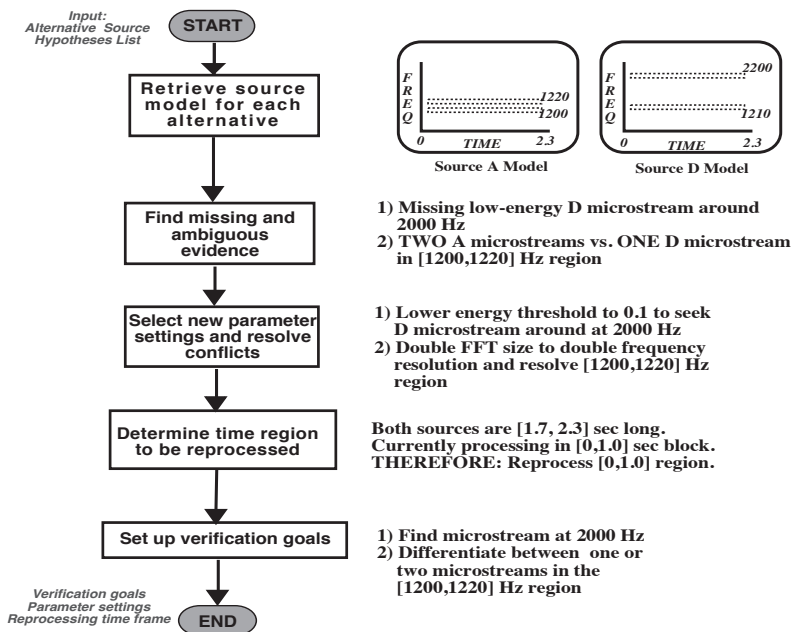
Figure 10: *A flowchart for the IPUS differential diagnosis KS and its execution in the acoustic testbed scenario in the next section. This situation occurs when a query to the source database returns more than one source model whose frequency components (or energy levels, or whatever other indexing feature is used) overlap the observed data. An example of this type of event appears in the trace when data in block 1 in the* [1200, 1230] *range could support both source A and source D's existence. For each retrieved model, an interpretation hypothesis supported by the observed data is created, and for each hypothesis, a source of uncertainty (ALTERNATIVE-EXPLANATION-SOU, in this case) is recorded in the problem-solving model. This SOU is left unresolved until the focusing heuristics deem its resolution appropriate to the current problem-solving context.*

like loss of low-energy components due to energy thresholding are considered. This design is justified because the differential diagnosis KS's role is to trigger reprocessing that quickly prunes large areas of underconstrained interpretation spaces, *without preference* for any particular interpretation. On the basis of this specification, it is not appropriate to devote time consuming, sophisticated reasoning to the generation of missing-evidence explanations.

In cases where an IPUS system prefers a particular interpretation over alternatives, and needs an explanation for why the interpretation is missing certain support, it will make use of the discrepancy diagnosis KS, with the initial state reflecting the preferred interpretation.

6.2   Testbed Domain Knowledge

The testbed consists of a blackboard with seven evidence abstraction levels, KSs for the primary IPUS components and for inferring hypotheses between different abstraction levels, an acoustic source library, and control plans. The testbed version described in this paper is called configuration $C.1$[5]

Figures 11 to 17 provide a short background description of the information represented in the evidence abstractions. At the lowest level are waveform segments derived from the input waveform. Each segment is a collection of points to which some SPA will be applied. The second level consists of spectral hypotheses derived for each waveform segment through frequency- and time-domain SPAs. The third level consists of "contour" hypotheses, each of which corresponds to a group of peaks (each from a different segment) whose time indices, frequencies, and amplitudes represent a contour in the time-frequency-energy space with uniform frequency and energy behavior. The fourth level contains microstream hypotheses supported by one contour or a sequence of contours. Each microstream has an energy pattern consisting of an attack region (signal onset), a steady region, and a decay (signal fadeout) region. In the fifth level we represent noisebeds as wideband frequency regions supported by clusters of relatively-low energy contours. Noisebeds represent the wideband component of a sound source's acoustic signature. Clusters represent groups of contours for which no microstream or noisebed hypothesis can be generated given the inference KS's default interpretation crite-

---

[5]In configuration $C.2$, which is currently under development, the control plans have been changed to accomodate a larger library of 33 real-world sounds with more complicated structure. The strategies in the new control plans still rely on the basic IPUS framework but now incorporate more goal-directed processing of microstreams and do not propagate the contour interpretations in a bottom-up manner to the microstream level. The processing strategies incorporate approximate-knowledge peak clustering algorithms to constrain source-model selection.

ria (e.g., noisebed contours must have an energy correlation of greater than X, microstream contours should not be separated from each other by more than Y milliseconds, etc.). Groups of microstreams and noisebeds synchronized according to time and/or some other psychoacoustic criteria such as harmonic frequency sets support "stream" hypotheses in the sixth level. For a detailed description of acoustic streaming processes, see [5]. At the seventh level, sequences of stream hypotheses are interpreted as sound-source hypotheses.
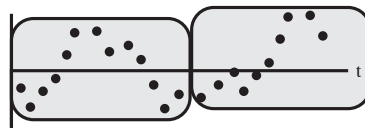
Figure 11: SEGMENT LEVEL: *A segment is a collection of raw data points for which such time-domain statistics such as zero-crossing density, average energy, etc, are maintained. Numeric-level SPAs operate on one segment at a time.*
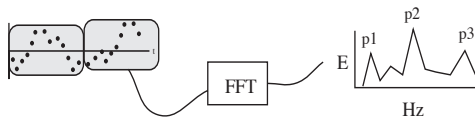

Figure 12: SPECTRUM LEVEL: *The second level consists of spectrum hypotheses derived for each waveform segment through Fourier-Transform-based algorithms such as the STFT and Wigner-Distribution [10] algorithms and peak-picking algorithms.*
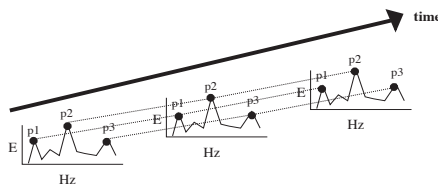

Figure 13: CONTOUR LEVEL: *The third level consists of contour hypotheses, each of which corresponds to a sequence of time-ordered peaks (each peak from a different segment) that are within specified time, frequency, and amplitude radii of their sequence neighbors.*

Sources are represented in the source database by an acoustic grammar specifying microstream and noisebed frequency ranges and permissible ranges of energy

Figure 15: NOISEBED LEVEL: *The fifth evidence abstraction level contains noisebed hypotheses supported by one or more contour clusters. Noisebeds represent the wideband component of a source's acoustic signature. Usually microstreams form "ridges" on top of noisebed "plateaux," but not every noisebed has an associated microstream.*
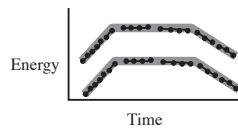

Figure 16: STREAM LEVEL: *Groups of microstreams and/or noisebeds synchronized according to time and/or some psychoacoustic criteria (e.g., harmonic sets, frequency separation) support stream hypotheses in the sixth level.*
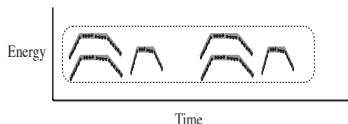
Figure 17: SOURCE LEVEL: *At the seventh level, sequences of stream hypotheses are used to support sound-source hypotheses.*

relationships among microstreams and noisebeds within source streams. The grammar also specifies the permissible range of durations for each source's microstreams and streams, and the stream sequences and periodic patterns that characterize the source.

7   IPUS Testbed Behavior

In this section we provide a detailed analysis of the acoustic interpretation testbed's behavior as it interprets an acoustic scenario. This example, though its scenario is constructed from synthetic narrowband sources, is indicative of the role and importance that a formal theory of signal processing can play in signal interpretation. It shows the architectural components' functionality and how they can exploit a formal theory of the relationship between the scenario characteristics and the SPA configurations.

7.1   Scenario Description

For this paper's testbed demonstration, the IPUS source database contains 5 narrowband synthetic sources (A, B, C, D, and E) whose frequency-energy behaviors are illustrated in Figure 18. The shaded regions indicate signal strength (e.g., volume or energy) as a function of time. The sources' frequency components are labeled by single-frequency values for ease of display. In the formal source definitions, frequency *ranges* are specified for each component.

Figure 19a shows the time-frequency display of the sources (A, B, C, and E) in a the experiment's scenario. In relative energy terms, the two sources A and C are 1.2 times as energetic as source B. Source E is an impulsive source with acoustic energy 5 times that of source B.

For this particular experiment, the testbed was configured to interpret waveform data in 1.0-second blocks, and was directed to identify quickly any occurrences of source C. The system's front-end signal processing parameters were therefore
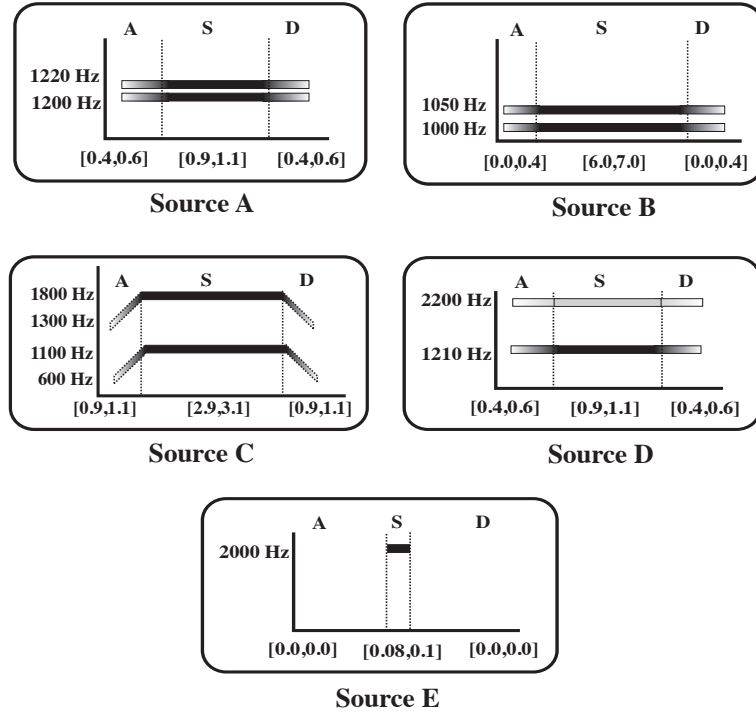
36

Figure 18: *IPUS Source Database. The vertical axes represent frequency and the horizontal axes represent time. The energy-level changes for each microstream are represented qualitatively by the shading gradations.*

set to detect source C's steady-energy behavior. The pertinent SPA parameters and their initial values in this experiment are:

FFT-SIZE: 1024

> *The number of uniformly-spaced samples from continuous Fourier spectrum used to model spectra.*

WINDOW-LENGTH: 1024

> *The number of data points to which each FFT in the Short-Time Fourier Transform (STFT) algorithm is applied ($\leq$ FFT-SIZE).*

STFT-OVERLAP: 0

> *Number of data points common to consecutive STFT intervals. In this experiment the value was set to zero to permit the fastest possible processing of the data.*

37

STFT-PEAK-ENERGY-THRESHOLD: 0.2

Used in peak-picking algorithm. Only peaks with energy above this percentage of the spectrum's maximum-energy peak are detected.

ABSOLUTE-NOISE-THRESHOLD: 0.001

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

SAMPLING-FREQUENCY: 10KHz

Rate at which the data stream is sampled.

Figure 19b shows how the scenario's source signatures actually would appear if the initial SPA parameters were used throughout the scenario without any recourse to reprocessing.

There are several critical actions that IPUS must perform if it is to reasonably analyze the scenario in Figure 19a. In block 1, IPUS encounters two alternative interpretations of the data in the [1200, 1220] frequency region. That is, there is the possibility that it it could be caused by source A or source D, or even 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 D's low-energy microstream from being detected if in fact it were being produced by the environment. 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 [1200, 1220] region into source A's two microstreams (see Figure 19b). The uncertainty engendered by this event is resolved through reprocessing under the direction of differential diagnostic reasoning, which increases resolution and decreases the energy threshold.

In block 2, IPUS detects a discrepancy between its time-domain energy-estimator algorithm output and its STFT algorithm output. The energy-estimator detects a substantial energy increase followed about 0.1 seconds later by a precipitous decrease. The STFT algorithm, however, produces no significant set of peaks to account for the signal energy flux. This is because the algorithm's time decimation (STFT-OVERLAP) is too small. IPUS also detects a discrepancy between expectations established from block 1 for the [1200, 1220] frequency region and the STFT algorithm's output. The STFT algorithm produces short contours that cannot support the expected microstreams for A because of inadequate frequency sampling in the region. Both discrepancies are resolved by reprocessing. The first discrepancy is resolved through reprocessing with a larger STFT-OVERLAP value and smaller STFT intervals, while the second is resolved through reprocessing with the finer frequency sampling provided by a 2048 FFT-SIZE.

In block 4, IPUS must handle an overlapping-sources situation. This arises from the overlapping of source B's steady region with source C's attack. IPUS also uses the discovery of source C's steady region in block 4 as the basis of re-interpreting block 3's short contours as evidence for source C's attack region.
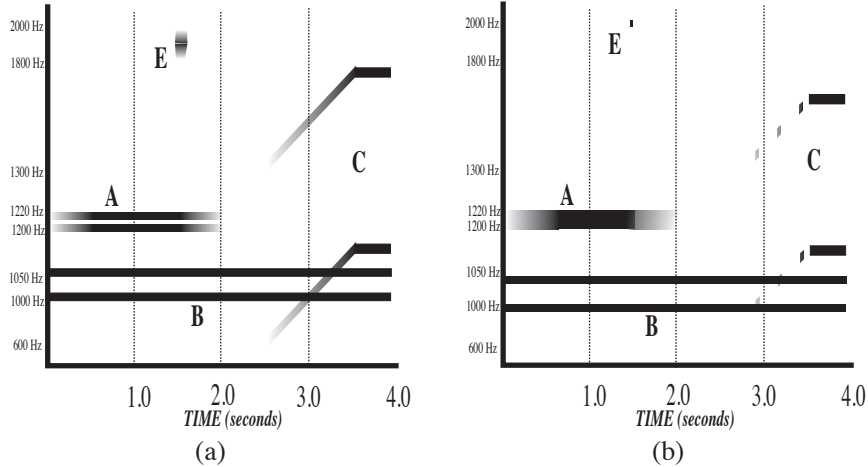

Figure 19: *Signal Events (figure 19a) and their distortions (figure 19b) in Testbed Example. Shading indicates attack (light-to-dark) and decay (dark-to-light) regions.*

7.2 IPUS Trace

The following is a high-level trace of the significant events that occurred as the system processed the scenario in Figure 19a.

7.2.1 BLOCK 1

Bottom-Up Processing: The testbed focusing heuristics specify that spectral information be gathered for the input waveform sampled during block 1. It is processed by a KS representing the STFT signal processing algorithm and a KS that uses a time-domain algorithm for estimating waveform energy as a function of time. Continuing in a data-driven manner, the spectra peaks produced are grouped by similar frequency and energy into *contours*.

Seek Evidence for Current Expectations: The focusing heuristics next direct the testbed to act upon current high-level expectations and search for support evidence. At this point in the scenario, however, there are no explicit

source expectations. In the source database, though, C's definition identifies it as a high priority source. This priority will impose an ordering on the data examined in the next step.

Drive Unexplained Data to Higher Levels: In deciding what evidence to examine first, testbed focusing heuristics choose to examine any evidence in the frequency regions given in source C's defined steady phase because of C's priority. No such data is found. Long contours found in the range [1000, 1050] Hz are used to hypothesize the existence of microstreams. These in turn are used to hypothesize the existence of the B source. Therefore a source-level hypothesis for B is posted to the interpretation blackboard.

Discrepancy-Detection Clustering: The testbed uses the heuristic that short (two- or three-peak) contours should not be used as evidence for microstreams since short contours could be the result of random noise, and the system should apply as little computing time as necessary to the processing of noise. Because a large number of short contours relative to the total number of contours is detected in the current block, IPUS performs discrepancy detection to determine if there are short-contour clusters that could indicate distorted sources. The system finds a high-population cluster in the [1200, 1250] Hz range, and then attempts to find a source hypothesis to explain the stream. A query to the source database yields two possible explanations, source A and source D. They are retrieved because for each source at least one of its frequency components overlaps the cluster frequency region. Therefore IPUS posts both sources' hypotheses as alternative explanations for the contour cluster. This use of short contours in place of long contours to support interpretations raises a violation discrepancy, since the a priori expectation that sources are indicated only by long contours is violated.

Handle Selected Uncertainties: At this point three SOUs have been posted: one each for the violation discrepancies associated with source A and source D being supported by a cluster, and one for the uncertainty associated with the existence of competing interpretations for the same cluster. The focusing heuristics elect to resolve the uncertainty associated with the alternative explanations. For doing this, the control plans specify a strategy of first performing differential diagnosis and using its results to guide data reprocessing. The differential diagnosis KS determines features of the two sources that should be searched for in the signal data because their presence or absence will permit differentiation between the alternatives.

40

Differential Diagnosis: The differential diagnosis KS selects the low-energy, 2000 Hz microstream of source D and the number of microstreams in the [1200, 1220] Hz region for each source (A has 2, D has 1) as discriminating features. It specifies that a lower energy-threshold be used to attempt to "bring out" source D's low-energy microstream at [2190, 2210] Hz. To attempt to find A's 2 microstreams, it specifies an FFT-SIZE value of 2048 to increase the frequency sampling in the [1200, 1220] Hz region. Note that IPUS at this time is not committed to either interpretation, nor to the possibility that **both** sources are present. It awaits the results of reprocessing.

Differential Reprocessing: The reprocessing KS is then executed and the sought-after D microstreams are not found. However, *two* well-defined contours are found in the [1200, 1220] Hz range that can support source A's microstreams. Therefore source A's belief is increased, while D's belief is decreased. D's belief level is very low at this point and is no longer considered as a significant alternative explanation for the original stream hypothesis. Note that this reprocessing opportunistically resolves not only the competing-interpretation uncertainty, but also the violation-discrepancy uncertainty of source A.

Define Expectations: Because source A's description indicates that its steady region is approximately 1 second long, and at most 0.5 seconds have been found, an explicit expectation for A's microstreams is posted for block 2's time period. Explicit expectations for the continuation of B's microstreams are also posted for block 2.

7.2.2   BLOCK 2

Bottom-Up Processing: Purely bottom-up processing creates spectra and contours for block 2. In the environment being monitored, source E emits a high-energy, short-duration (0.09 sec) signal burst. The testbed control plans were designed to perform fault discrepancy detection immediately after bottom-up signal processing is completed. This causes a fault discrepancy to be detected between the time-domain energy monitoring algorithm and the STFT algorithm. The time-domain algorithm detects a sharp increase followed by a sharp decrease in signal energy, whereas the STFT produced no peaks to generate a significant-length contour that started and stopped around the times indicated by the signal-energy shifts.

Handling Fault Discrepancy Uncertainties: The tesbed's focusing heuristics were designed to select SOUs related to faults for handling before the control

plans apply any interpretation KSs. Thus, before any source expectations' components are searched for, the fault discrepancy is selected for handling by the focusing heuristics[6]. The diagnosis KS explanation for this discrepancy is (CONTOUR-TIME-RESOLUTION). That is, the STFT overlap is too low to detect enough peaks to generate contours of significant length to account for the signal energy increase.

Discrepancy Reprocessing: The reprocessing KS uses the explanation to decide to reprocess data from the 0.09-second time region (NOT the entire block) with a 256-point STFT window length, a 1024-point FFT length, and a 128-point interval-overlap. This produces seven peaks in the 2200 Hz region, which create a significant-length contour. This contour's existence resolves the fault discrepancy.

Seek Evidence for Current Expectations: The focusing heuristics act on C's priority and decide to examine data found in C's expected frequency regions. No data is found. At this point, the focusing heuristics decide to gather evidence for explicit source expectations. Contours in source B's expected regions are found, thus support is found for B's persistence into block 2. Note that when support for a source's microstreams is found, it is immediately propagated through the higher evidence levels (microstream and stream) to the source level.

Discrepancy Detection: Discrepancy detection is now performed for the other discrepancy classes: conflicts and violations. As occurred in block 1, the front-end processing parameters produce a cluster of short contours in the [1200, 1250] Hz range. Again, IPUS relies on the heuristic that short contours not be used to support microstreams. Thus, the expected microstream extensions for source A are not found. This raises a conflict discrepancy. No violation discrepancies are found.

Handling Conflict Discrepancy Uncertainty: The discrepancy diagnosis KS takes as input a desired interpretation state containing the two microstreams and an observed interpretation state containing neither. The KS's means-ends analysis algorithm returns the explanation (COARSE-FREQUENCY-SAMPLING). That is, the KS proposes that the STFT analysis was done

---

[6]Since the duration of this discrepancy indicates that it is not related to C, the astute reader may wonder why C's high criticality did not override this choice and force IPUS to look for C data first. This is a shortcoming of the *C.1* heuristic set. It is rectified in the *C.2* configuration's heuristic set.

with inadequate frequency sampling, causing the two microstreams to appear as the contour cluster actually observed. The KS also returns a "scenario-specification" indicating that if this diagnosis is correct, in the next block under the same initial parameter settings, A's microstreams will appear like the contour cluster again, and to avoid spending time to diagnose the discrepancy, IPUS should accept the cluster's short contours as support without raising a discrepancy. In this scenario, however, the scenario-specification will not be useful because source A's steady region should not extend into block 3.

Discrepancy Reprocessing: The reprocessing KS acts upon the diagnosis explanation and retrieves a processing plan directing that the data be reprocessed up to the microstream level of abstraction with an FFT-SIZE value of twice the original (2 * 1024 = 2048 in this case). The doubling of the FFT-SIZE provides finer frequency sampling in the spectra produced by the STFT algorithm. After one iteration of this plan, the desired microstreams are found, and their expectations in the next block are annotated with the discrepancy diagnosis KS's scenario-specification.

Drive Unexplained Data to Higher Levels: The 0.09-second contour is found to match the short-duration, higher-frequency, higher-energy characteristics of source E. Hence, a source hypothesis for E is posted. Source E has no other components, and its attack and decay regions are undetectable, so no other processing is possible for verifying E's existence.

Resolve Selected Uncertainties: The entire attack and decay regions of A have not been found; therefore A only has partial evidential support for its identification. Thus, an SOU indicating this partial support is present for the source in the problem-solving model. Source A's belief rating is rather high because support has been found for practically all of its expected second microstream region (steady). Thus, IPUS chooses not to pursue the search for more evidence for a source whose existence is strongly believed.

Define Expectations: Again, because B's model indicates that the source's steady behavior could continue for 0 to 4 more seconds, an explicit expectation for its continuation is posted for Block 3's time period. NO expectation for source A is posted because its description indicates that its duration is at most 2 seconds long.

7.2.3 BLOCK 3

Bottom-Up Processing: Block 3's signal data is now processed. Bottom-up processing culminates in the creation of contours.

Seek Evidence for Current Expectations: Data is sought for in source C's frequency regions. It is true that some contours are present in this block from C's attack phase, but because the IPUS testbed first recognizes sources by steady characteristics (due to their more predictable behavior), they are not used immediately to support the creation of a C source hypothesis. Contours extending source B's microstreams are sought for and found.

Drive Unexplained Data to Higher Levels: Because of their short lengths and the short-contour heuristic mentioned earlier, the contours caused by C's attack phase are not selected to hypothesize the existence of any microstreams. They are simply labeled as possible-noise data. These contours are spread evenly across a wide frequency region. Therefore, the violation-detection clustering algorithm does not find any high-density cluster to justify raising a discrepancy.

Define Expectations: The testbed posts an expectation that B's microstreams will continue into block 4.

7.2.4 BLOCK 4

Bottom-Up Processing: Block 4's signal data is now processed. Bottom-up processing culminates in the creation of contours.

Seek Evidence for Current Expectations: Contours supporting a small part of C's steady region are detected in block 4. Because C's attack region is unsupported, however, the confidence level for C is low. Due to C's criticality, however, the focusing heuristics decide to resolve this low-confidence uncertainty. IPUS engages in goal-directed processing to find contours supporting C's attack region's existence.

Reprocessing: To find "enough" (60% in this case) of C's attack region, IPUS must search back into block 3 and reinterpret the previously-detected but unrecognized short contours as valid attack-region contours. C's attack region and its chirp characteristics are identified in the previous block's signal data[7].

---

[7]In the current implementation, signal data from the current block and the 2 most recent blocks are buffered. Future implementations will have this buffering governed by a parameter.

44

At this point source C is determined to be present with high confidence.

Seek Evidence for Current Expectations: (cont) Contours extending source B's microstreams are searched for and found.

8   Conclusions

In this paper we have considered the problem of signal understanding in complex environments involving interacting objects which mask and/or distort data correlates of their respective features. This implies that during its operation, the perceptual system must continually update, in a context-dependent fashion, what feature-set to focus upon and what SPAs to use in order to extract the feature's data correlates. It is important to observe that the selection of a particular SPA is determined not only by the subset of features whose data correlates are sought, but also the presence of data unrelated to those features. We have argued that adaptive selection of features and their corresponding SPAs requires sophisticated but principled control of the interactions between the actions of high-level knowledge sources and the actions of SPAs in a signal understanding system. Motivated by this insight, we have formulated the IPUS architecture for the integrated processing and understanding of signals.

IPUS provides a framework for structuring bi-directional interaction between the search for SPAs appropriate to the environment and the search for interpretation models to explain the SPAs' output data. The availability of a formal signal processing theory is an important criterion for determining the architecture's applicability to any particular domain. IPUS allows system developers to organize diverse signal processing knowledge along the lines of formal concepts such as SPA processing models, discrepancy tests, distortion operators, and reprocessing-SPA application strategies. A major contribution of the architecture is to formalize and unify front-end SPA reconfiguration performed for interpretation processes such as differential diagnosis with that performed for refining data correlates under discrepancy diagnosis as a single reprocessing concept driven by the presence of SOUs.

With respect to scaling, one might argue that the time required by multiple reprocessings under IPUS would be unacceptably high in noisy environments. This view ignores IPUS's advantage over other paradigms in that it *selectively* samples several front-end processings' outputs, avoiding the traditional approach of continuously sampling several front-end processings' results. In this respect the IPUS paradigm decreases the expected processing time for contexts requiring several

independent processing views for plausible interpretations.

Our sound understanding testbed experiments indicate that the basic functionality of the architecture's components and their interrelationships are realizable. Since the IPUS architecture is intended to be applicable to all signal understanding domains for which the SPAs have a rich underlying theory, it is important to build and experiment with IPUS-based systems in other domains as well. An indication of the architecture's generality can be seen in its use not only in the sound understanding testbed discussed in this paper but also in a radar interpretation system being developed at Boston University.

IPUS has important implications for SPA design because it encourages the development and application of fast, highly specialized, theoretically sound SPAs for reprocessing in appropriate contexts. Traditionally the focus in SPA design has been to develop SPAs that extract, as precisely as possible, *all* details of the desired information from the input signals. The premise behind this approach is that obtaining precise, detailed information through the signal processing will lead to less uncertain interpretations and a more efficient search for those interpretations. This strategy is appropriate provided it can be guaranteed that the signal understanding system will not encounter signals which violate the underlying assumptions made in the design of those SPAs. As we have asserted earlier, this premise is not appropriate for perceptual systems operating in complex environments. Since in such domains the SPA assumptions will often be violated, it appears unreasonable to devote computational resources to the extraction of detailed and precise information that is likely to be misleading. Instead, it would be more appropriate to use an IPUS strategy wherein the initial signal processing sacrifices detail and precision, which are then sought during the signal re-processing phase when a better assessment of the signal environment is available. The advantage of sacrificing precision and detail in the initial signal processing is two-fold; the initial signal processing can be more computationally efficient and the discrepancy detection following it is not encumbered by needless quantities of detail. In the course of our own research on the acoustic interpretation testbed, we have developed a novel algorithm [25] for computing an approximation to the STFT. This approximation retains the major features in the regular STFT output but its computation requires essentially no multiplications (a major part of regular STFT computation) and significantly fewer additions than the regular STFT.

In summary, we have shown how knowledge from formal signal processing theory regarding the effectiveness of specific SPA configurations for particular environments can be used to develop a highly adaptive signal understanding architecture. This architecture tightly integrates the search for the appropriate SPA configura-

tion with the search for plausible interpretations of the SPA output data. In our opinion, this dual search, informed by formal signal processing theory, is a necessary component of perceptual systems for intelligent agents that must interact with complex dynamic environments.

9 Acknowledgments

References

[1] B. Bell and L. F. Pau, "Context Knowledge and Search Control Issues in Object-Oriented Prolog-Based Image Understanding," *ECAI '90 Proceedings*.

[2] B. Bell and L. F. Pau, "Contour Tracking and Corner Detection in a Logic Programming Environment," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, August 1990.

[3] N. Bitar, E. Dorken, D. Paneras and H. Nawab, "Integration of STFT and Wigner Analysis in a Knowledge-Based Sound Understanding System," *IEEE ICASSP '92 Proceedings*, March 1992.

[4] P. Bonissone and P. Halverson, "Time-Constrained Reasoning under Uncertainty," *The Journal of Real-Time Systems*, vol. 2, Kluwer Academic Publishers, pp. 25-45, 1990.

[5] A. Bregman, "Auditory Scene Analysis: The Perceptual Organization of Sound," MIT Press, 1990.

[6] R. Brooks, "Intelligence without Representation," *Artificial Intelligence*, vol. 47, no. 1-3, Jan. 1991, pp. 139–159.

[7] N. Carver and V. Lesser, "The Evolution of Blackboard Control," to appear in *Expert Systems with Applications*, special issue on The Blackboard Paradigm and Its Applications, Nov. 1993 (also available as Technical Report 92-71, Computer Science Department, University of Massachusetts, 1992).

[8] N. Carver and V. Lesser, "A Planner for the Control of Problem-Solving Systems," to appear in *IEEE Transactions on Systems, Man, and Cybernetics* Special Issue on Planning, Scheduling and Control, 1993.

[9] N. Carver and V. Lesser, "A New Framework for Sensor Interpretation: Planning to Resolve Sources of Uncertainty," *AAAI '91 Proceedings*, pp. 724–731.

[10] T. Claasen and W. Meclenbrauker, "The Wigner Distribution: A Tool for Time-Frequency Signal Analysis," *Phillips J. Res.*, vol. 35, pp. 276–350, 1980.

[11] B. Dawant, B. Jansen, "Coupling Numerical and Symbolic Methods for Signal Interpretation," *IEEE Transactions on Systems, Man and Cybernetics*, Jan/Feb 1991.

[12] R. De Mori, L. Lam, and M. Gilloux, "Learning and Plan Refinement in a Knowledge-Based System for Automatic Speech Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Feb 1987, pp 289-305.

[13] E. Dorken, H. Nawab and V. Lesser "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.*

[14] W. Dove, *Knowledge-Based Pitch Detection*, PhD Thesis, MIT, 1986.

[15] L. Erman, R. Hayes-Roth, V. Lesser, D. Reddy, "The Hearsay II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys*, v. 12, June 1980.

[16] F. Giannesini, H. Kanoui, R. Pasero, and M. van Caneghem, *Prolog*, Reading, MA: Addison Wesley, 1986.

[17] W. E. L. Grimson and T. Lozano-Pérez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *the International Journal of Robotics Research*, vol. 3, no. 3, 1984.

[18] B. Hayes-Roth, R. Washington, R. Hewett, M. Hewett, and A. Seiver. "Intelligent Monitoring and Control," *IJCAI '89 Proceedings*, pp. 243–249.

[19] E. Hudlická, and V. Lesser, "Modeling and Diagnosing Problem-Solving System Behavior," *IEEE Transactions on Systems, Man and Cybernetics. Special Issue on Diagnostic Reasoning.* May/June 1987.

[20] E. Hudlická, and V. Lesser, "Meta-Level Control Through Fault Detection and Diagnosis," *AAAI '84 Proceedings*, 1984, pp. 153–161.

[21] F. Klassner, *Data Reprocessing and Assumption Representation in Signal Understanding Systems*, TR 92-52, Computer Science Dept., University of Massachusetts, 1992.

[22] C. Kohl, A. Hanson and E. Reisman, "A Goal-Directed Intermediate Level Executive for Image Interpretation," *IJCAI '87 Proceedings*, pp 811–814.

[23] V. Lesser, H. Nawab, M. Bhandaru, Z. Cvetanović, E. Dorken, I. Gallastegi, and F. Klassner, *Integrated Signal Processing and Signal Understanding*, TR 91-34, Computer Science Dept., University of Massachusetts, 1991.

[24] J. N. Maksym, et al, "Machine Analysis of Acoustical Signals," *Issues in Acoustic Signal/Image Processing and Recognition*, C. H. Chen, ed, NATO ASI Series, Vol F1, Springer-Verlag, 1983.

[25] H. Nawab and E. Dorken, "Efficient STFT Computation Using a Quantization and Differencing Method," *Proceedings of the 1993 IEEE Conference on Acoustics, Speech and Signal Processing*, Minneapolis, Minnesota, to be held April 1993.

[26] H. Nawab and V. Lesser, "Integrated Processing and Understanding of Signals," ch 6, *Knowledge-Based Signal Processing*, A. Oppenheim and H. Nawab, editors, Prentice Hall, New Jersey, 1991.

[27] H. Nawab and T. Quatieri, "Short-Time Fourier Transform," *Advanced Topics in Signal Processing*, Prentice Hall, New Jersey, 1988.

[28] H. Nawab, V. Lesser, E. Milios, "Diagnosis Using the Underlying Theory of a Signal Processing System," *IEEE Transactions on Systems, Man and Cybernetics. Special Issue on Diagnostic Reasoning.* May/June 1987.

[29] A. Newell and H. Simon, "GPS: A Program that Simulates Human Thought." *Computers and Thought*, Feigenbaum and Feldman, eds., pp 279–293. McGraw-Hill, 1963.

[30] H. Nii, E. Feigenbaum, J. Anton, A. Rockmore, "Signal-to-Symbol Transformation: HASP/SIAP Case Study," *AI Magazine*, vol 3, Spring 1982.

[31] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[32] Y. Peng and J. Reggia, "Plausibility of Diagnositic Hypotheses: The Nature of Simplicity," *AAAI '86 Proceedings*, pp 140–145.

[33] D. Seborg, et al., "Adaptive Control Strategies for Process Control: A Survey," *AIChE Journal*, vol. 32, no. 6, pp. 881-913, June 1986.

[34] M. Swain, and M. Stricker, eds. *Promising Directions in Active Vision,* NSF Active Vision Workshop, Technical Report CS 91-27, Computer Science Dept, University of Chicago, 1991.

[35] M. Williams, "Hierarchical Multi-Expert Signal Understanding," in *Blackboard Systems*, Robert Engelmore and Anthony Morgan, editors, Addison-Wesley, 1988.