IPUS: An Architecture for the Integrated Processing and Understanding of Signals

Victor R. Lesser* S. Hamid Nawab† Frank I. Klassner*

*Computer Science Department
University of Massachusetts
Amherst, Massachusetts 01003

†Electrical, Computer, and
Systems Engineering Dept.
Boston University
Boston, MA 02215

March 20, 1994

Abstract

The *Integrated Processing and Understanding of Signals* (IPUS) architecture is presented as a framework that exploits formal signal processing models to structure the bidirectional 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 the simultaneous occurrence of 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. The 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 that must interact with complex environments. To explain this architecture in detail, we discuss examples of its use in an implemented system for acoustic signal interpretation.

1 Introduction

Since the middle 1970's, a major focus in perceptual architecture design has been the identification and organization of knowledge to permit recovery from uncertainty introduced by front-end numeric signal processing algorithms (SPAs). One can categorize research efforts in this area along five dimensions according to where they emphasize the placement of this knowledge:

1. within high-level interpretation knowledge sources (HLKSs) (e.g., as improved or approximate models of environmental phenomena [18, 20, 32, 44]),

2. within numeric-level KSs (SPAs) (e.g., as control parameter optimization processes or feedback loops [8, 22, 42]),

3. in the control of HLKSs' application (e.g., in planning architectures for controlling KS activation and sophisticated evidential representations [6, 10, 11, 20, 39]),

4. in the control of SPAs' application (e.g., as differential diagnosis rules for SPA application to disambiguate objects in the environment [14, 15, 29] or as compiled "SPA trees" learned for particular objects [19]), and

5. in the control of the interaction between HLKSs and SPAs [1, 2, 5, 14, 15, 29, 23].

Over the past two decades, research efforts along each of the first four dimensions has been quite fruitful, 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 lying along the first four dimensions. 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 to the current scenario can render a perceptual system unable to interpret entire classes of environmental events correctly. Front-end SPA sets

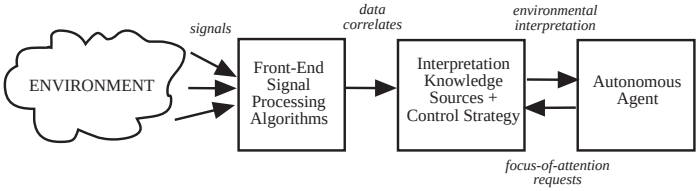

Figure 1: *Classic Knowledge-Based Signal Processing Architecture. This paradigm imposes a unidirectional control flow that limits interpretation processes' analysis to only the single set of observations afforded by the fixed signal processing. Interpretation processes do not usually provide structured feedback to the front-end about either the adequacy of the signal processing outputs to be interpreted or any anticipated signal behavior.*

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 [36] (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 along the fifth knowledge-placement dimension: 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 or 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 [27, 28, 30, 31, 34, 35]. It enforces structured, bidirectional interaction between a perceptual system's interpretational components and signal processing components. This interaction combines the search for front-end SPA configurations appropriate to the environment with the search for plausible interpretations of front-end processing results. The architecture is instantiated by a domain's formal signal processing theory. It has four primary components as conceptual "hooks" for organizing and applying signal processing

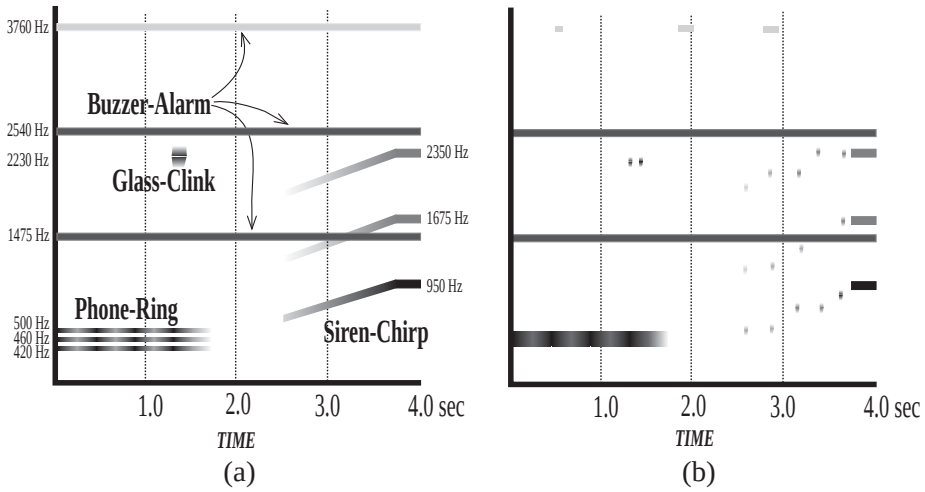

Figure 2: *Figure 2b shows distortions introduced by a STFT SPA and a peak-picker SPA with inappropriate parameter settings applied to the acoustic scenario described in Figure 2a. Darker shading indicates higher energy. The STFT parameter settings used throughout 2b were FFT-SIZE: 512, WINDOW-LENGTH: 512, and DECIMATION: 512, while the peak-picker's parameter setting was PEAK-THRESHOLD: 0.09. The signal was sampled at 8KHz. DECIMATION is the separation between consecutive analysis window positions; the value was set to 512 to permit the fastest possible processing of the data. PEAK-THRESHOLD is the normalized energy required for a discrete Fourier transform point to be considered as a peak. In 2b's first second,* `Phone-Ring`*'s tracks are merged because the STFT's frequency resolution is not adequate for such close features.* `Glass-Clink`*'s frequency track is not even detected in 2b's next second because the STFT's analysis window doesn't provide adequate time resolution to isolate the source's spectral features. The energy threshold causes the peak-picker to miss* `Buzzer-Alarm`*'s low-energy track.*

theory: discrepancy detection, discrepancy diagnosis, differential diagnosis, and signal reprocessing. These components have the following functionality:

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

This paper discusses the generic IPUS 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.[1] In the following sections 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, (7) discuss the architecture's implications for SPA design, and (8) indicate directions for our future research.

2 Perception in Complex Environments

In this section we discuss relationships between the nature of perception in complex environments and the means by which systems actually perceive environments. 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.

Definition 4 (Context 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*

[1] The problem of identifying and tracking sounds.

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 our concept of a perceptual system's environment, let us now consider SPAs, the means by which a system 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 [36] 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.[2]

In complex environments, there are often many SPAs which can potentially compute a correlate's value. The effectiveness of an SPA to produce correlates

[2] A variant analysis of the Heisenberg Uncertainty Principle implies that one cannot obtain a STFT SPA instance (or, for that matter, design a new generic SPA) that simultaneously provides infinite frequency resolution and infinite time resolution.

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 show 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 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

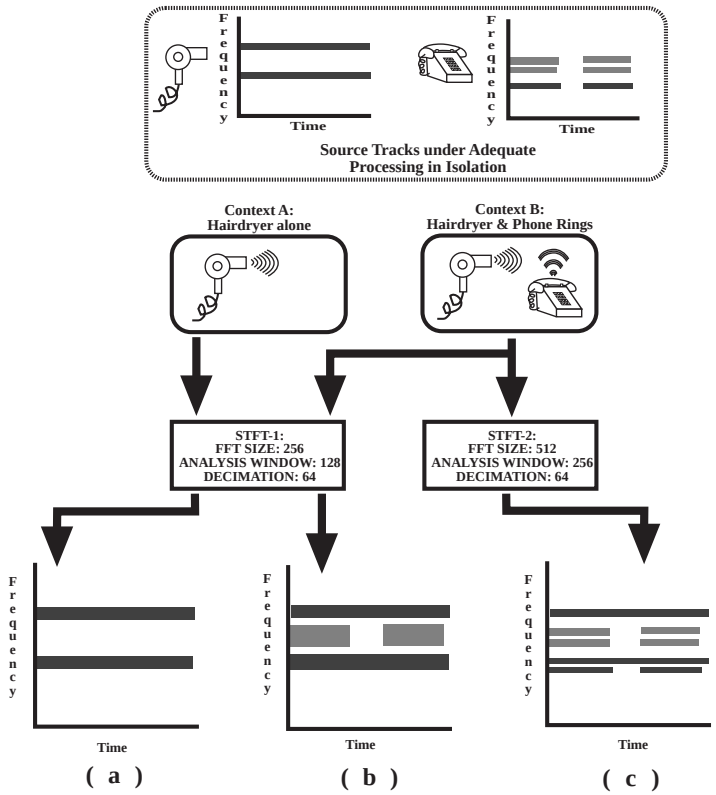

Figure 3: *Context-Dependent Correlate Computation. When STFT-1 analyzes context A's signal, its frequency correlates in (a) are adequate for unambiguously identifying the hairdryer's two frequency tracks. When the same SPA analyzes context B's signal, however, its frequency correlates in (b) are not 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.*

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 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 [17] 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 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.

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.

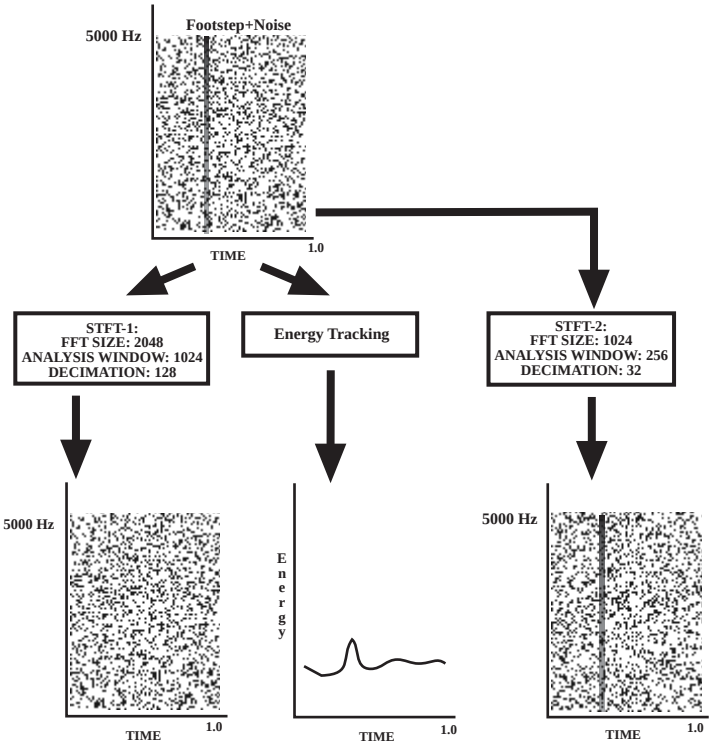

Figure 4: *Context-Dependent Correlate Computation. The energy tracking SPA provides correlates for context-independent energy burst features. These features which guide the focused application of an STFT with parameters to find frequency-track correlates for the footstep impulse in a noisy environment.*

- 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 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 provides summaries of the algorithms used to instantiate the IPUS components in the acoustic interpretation testbed [31].

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 expectations. 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 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

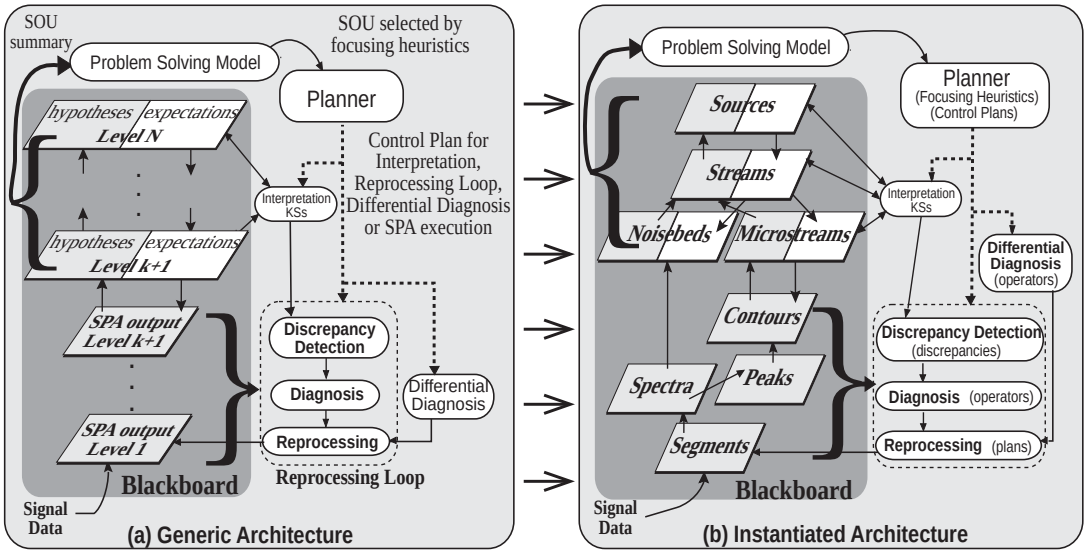

Figure 5: *Figure 5a shows the generic IPUS architecture and Figure 5b shows the architecture instantiated for the sound understanding testbed. Solid arrow lines indicate dataflow relations. Dotted arrow lines indicate classes of plans that the planner can pursue when trying to reduce or eliminate particular uncertainties (discrepancies) in the problem solving model that were selected by the focusing heuristics. Parenthesized terms indicate knowledge added to the planner or system knowledge sources to instantiate the architecture for an application. Note that reprocessing plans can cause SPA execution at any SPA output level, not just the lowest.*

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. Additionally, reprocessing strategies and discrepancy detection application-constraints tests can include the postponement of reprocessing or discrepancy declarations until specified conditions are met in the next data block(s).

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 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 inter-

preting 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. Because we make expectations take on the maximum possible values for their object features, this conflict could also indicate that the expectation's duration was too long.

Examination of a wide range of domains reveals two generic classes of correlates: *point correlates* and *region correlates*. A point correlate is a value associated with one point in the SPA output coordinate space. A region correlate is a value associated with a subset of the SPA output space. Consider the following examples. A spectral peak energy value in the "time, frequency, energy" space of acoustic signal processing and an image pixel intensity value in the "x, y, intensity" space of image processing are examples of point correlates. A noise-distribution tag for a region in a radar sweep and a mean-intensity value for a region in the output of an image filtering SPA are examples of region correlates. A track of spectral peaks over 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 W sample points is applied to a signal sampled at R samples per second. If the signal came from a scenario containing frequency tracks closer than R/W Hz, 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 [37]. 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 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 provide 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 [11, 12] 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* 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 hypothe-

ses' uncertainty. It emphasizes the separation of hypothesis belief evaluation from control decision evaluation by making control responsive not only to the levels of numeric belief in hypotheses but also to the presence of specific SOUs in the problem-solving model. 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, which are 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-placement dimensions 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 [37] and in work on meta-level control [24, 25] that used a process of fault-detection, diagnosis, and replanning to decide the most appropriate parameters for controlling a problem-solving system.

Although we oriented this research most strongly along the fifth knowledge-placement dimension, we feel it has implications for work along the other four dimensions as well. The architecture supports the use of an application domain's formal signal processing theory in selecting approximate or specialized SPAs for context-dependent application to specific portions of a signal [33]. For this reason the research also extends work that emphasizes the fourth dimension (control of SPA application).

Several recent systems have been developed that provide for structured interaction between interpretation activity and numeric-level signal processing. In this section we discuss selected frameworks or systems 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 [23] 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 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 [14] 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. [15] 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).

GOLDIE [29] is an image segmentation system that uses high-level interpretation goals to guide the choice of numeric-level segmentation algorithms, their sensitivity settings, and region of application within an image. The system's architecture represents the set of architectures that place strong emphasis on selective SPA application without explicit guidance from formal signal-processing theory. The system uses a "hypothesize-and-test" strategy to search for algorithms that will satisfy high-level goals, given the current image data. While it incorporates an explicit representation of algorithm capabilities to aid in this search, and an explicit representation of reasons for why it assumes an algorithm is appropriate or inappropriate to a particular region, the system notably does not incorporate any diagnosis component for analyzing unexpected "low quality" segmentations. If an algorithm were applied to a region and the resulting segmentation were of unexpectedly low quality, the framework would not parallel IPUS and attempt to diagnose the discrepancy and exploit this information to reformulate the algorithm search but would select the next highest rated algorithm from the original search.

In the same category as GOLDIE is TraX [5], a system for interpreting image frame sequences. Although its design was driven by the goal of supporting multiple, concurrent object descriptions, the system incorporates some concepts similar to those in our formulation of the IPUS architecture. The system supports detection of deviations from expected measurements and determination of the possibility that these deviations might have resulted from processing techniques inappropriate to the current context. In a manner similar to conflict discrepancy detection in IPUS, TraX compares higher-level expectations from previous frames against its segmentation SPAs' outputs for the current frame. In contrast to the IPUS architecture specification, however, TraX does not use models derived from an underlying theory for its SPAs to inform the discrepancy detection and diagnosis processes. It relies instead on empirically derived statistical performance models for the segmentation algorithms. While TraX allows for the use of different SPAs for different contexts, it does not support the adaptation of SPAs' control parameters for different contexts.

Bell and Pau [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* [21] 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 [43], and tends to use a control-theoretic approach for making reconfiguration decisions. It is indeed sometimes possible to reduce the reconfiguration of small sets of front-end SPAs to problems in linear control theory. In general, however, the problem of deciding when an SPA (e.g., a specialized shape-from-X algorithm or an acoustic filter) with particular parameter settings is appropriate to a given environment may involve nonlinear control or be unsolvable with current control theory techniques.

It is important to clarify the relationship between the IPUS approach and the classic control theoretic approach [42]. 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 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 CONSTRAINT SOU, 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 W sample points is applied to a signal sampled at R samples per second. If the signal came from a scenario containing frequency tracks closer than R/W Hz, 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 [37] 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 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.

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 [41]. 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

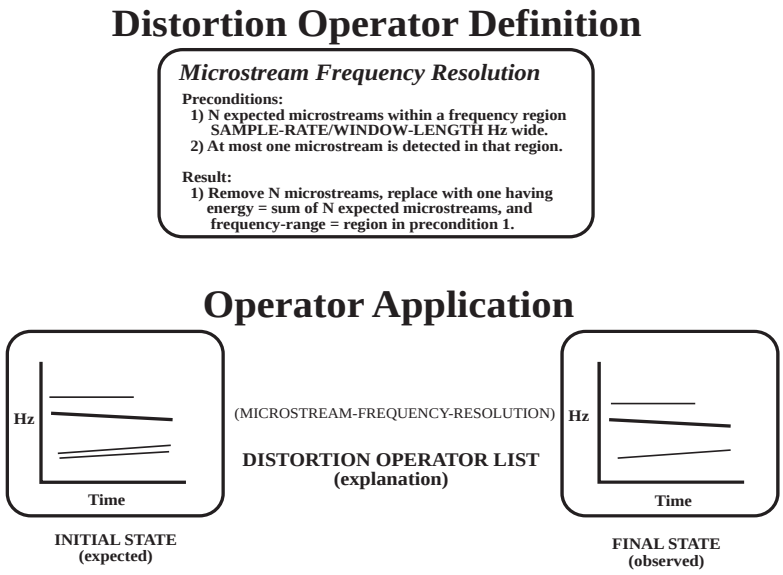

Figure 6: *Microstream Frequency Resolution Operator from the Acoustic Interpretation Testbed. When applied to a state, the operator replaces each set of expected microstreams whose members are closer than SAMPLE-RATE/WINDOW-LENGTH with a single microstream, reflecting the resolving limits associated 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.*

them in the original sequence. If no patch is possible, and no alternative explanations 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.

An issue not addressed in earlier work [37] 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

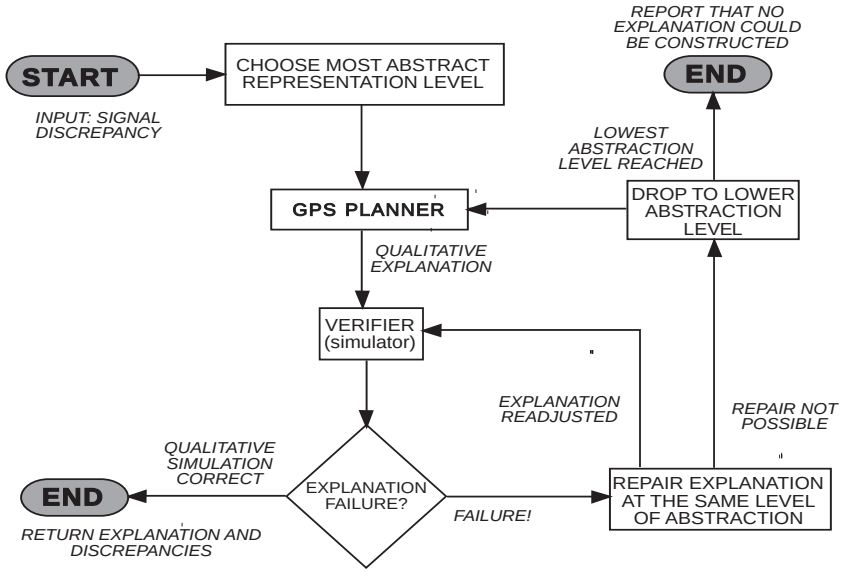

Figure 7: *The plan-and-verify strategy of the IPUS discrepancy diagnosis knowledge source.*

parameter values. When a complete distortion-operator sequence is generated, all operators' support-specifications are conjunctively combined to form a single expectation specification. This specification is then attached to the expectation units of the hypotheses involved in the original discrepancy. For those feature hypotheses, this annotation reduces the quality-level required for future evidence. 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.

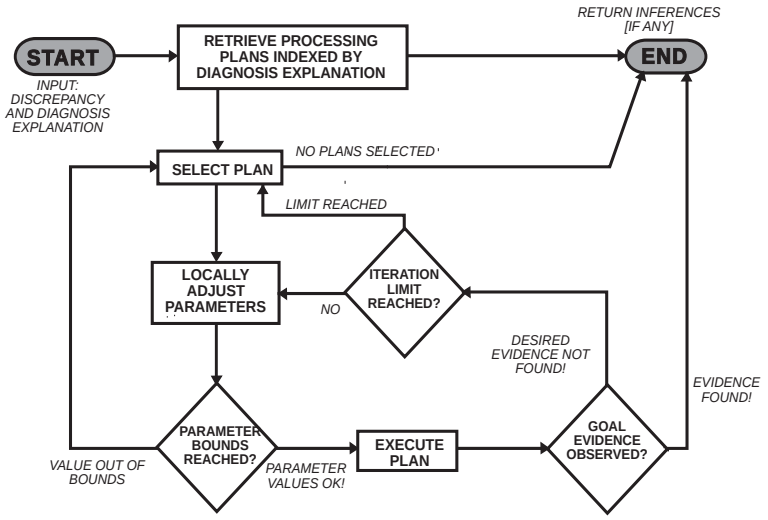

Figure 8: *The IPUS reprocessing knowledge source's framework*

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, failure is indicated automatically. The second criterion relies on fixed lower and upper bounds for signal processing parameters. If a plan reiteration requires a

```
(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.*

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
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 a typical execution for the testbed differential diagnosis KS.

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 not a rich set of explanations available. Only simple single-operator distortions 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. For related reasons, the differential diagnosis KS does not return support specifications that reduce the quality-level required for future evidence. The KS's shallow explanations generated for finding contrasts within a set of several sources might not justify the acceptance of lower quality evidence for a single source from that set.

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.

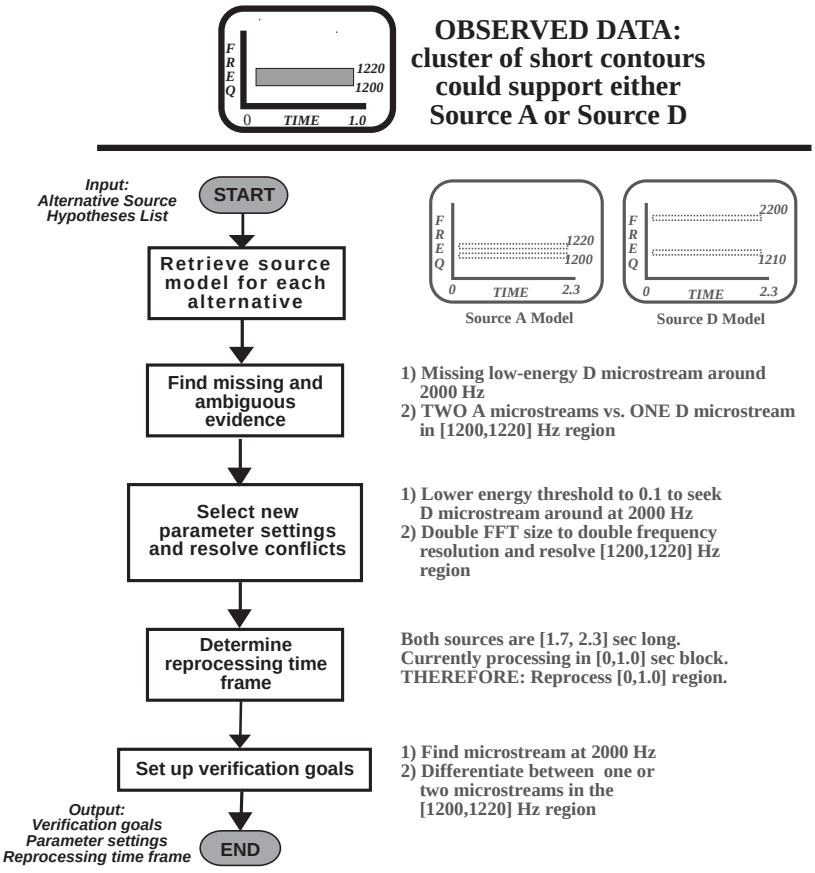

Figure 10: *A flowchart for the IPUS differential diagnosis KS and its execution in a typical acoustic scenario. In this example a database query returns more than one sound model whose frequency components overlap the observed data in the* $[1200, 1220]$ *Hz region. For each model, the IPUS system posts an interpretation hypothesis supported by the observed data. In the problem-solving model, an* $ALTERNATIVE\text{-}EXPLANATION$ *SOU is recorded for each hypothesis. These SOUs are left unresolved until selected by the system's focusing heuristics.*

6.2 Testbed Domain Knowledge

The testbed consists of a blackboard with eight 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*.[3]

Figure 11 describe 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. Time-domain statistics such as zero-crossing density, average energy, etc, are also maintained for segments. The second level consists of spectral hypotheses derived for each waveform segment through Fourier-Transform-based algorithms such as the STFT and Wigner-Distribution [13] algorithms. The third level consists of peak hypotheses derived for each spectrum and is used to support narrow-band features of sounds. The fourth level consists of contour hypotheses, each of which corresponds to a group of peaks whose time indices, frequencies, and amplitudes represent a contour in the time-frequency-energy space with uniform frequency and energy behavior. The fifth 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 sixth level we represent noisebeds as wideband frequency regions supported by regions within spectra. Noisebeds represent the wideband component of a sound source's acoustic signature. Usually microstreams form "ridges" on top of noisebed "plateaux," but not every noisebed has an associated microstream. Groups of microstreams and noisebeds synchronized according to time and/or other psychoacoustic criteria such as harmonic frequency sets support stream hypotheses in the seventh level. Bregman [7] provides a highly detailed account of various psychoacoustic streaming processes. At the eighth level, sequences of stream hypotheses are interpreted as sound-source hypotheses.

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

[3]Configuration *C.2* is currently under development as a platform for exploring approximate processing and scaling issues.

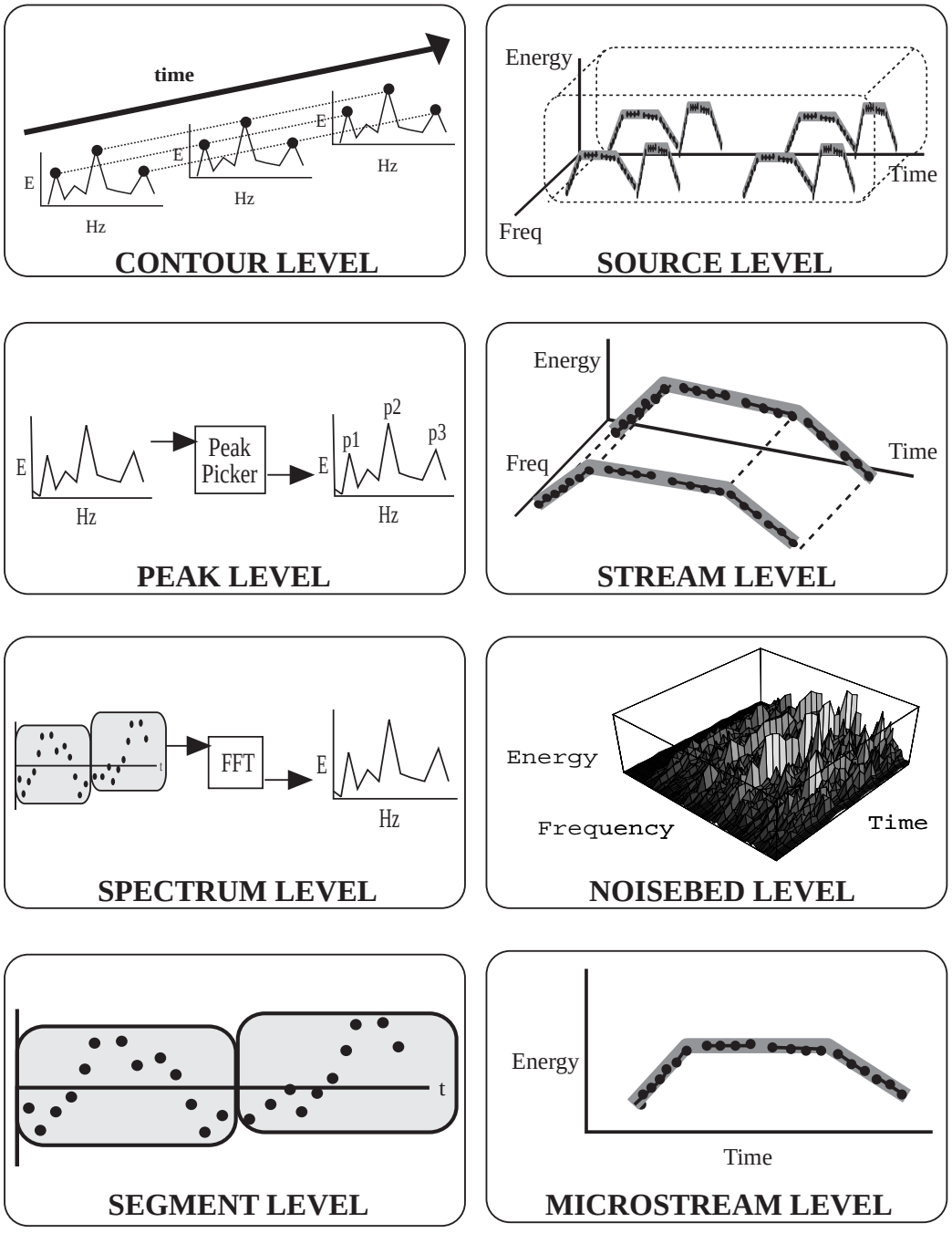

Figure 11: *Testbed Evidence Abstractions.*

7 Acoustic Interpretation Testbed Operation

In this section we provide a detailed analysis of the acoustic interpretation testbed's behavior as it interprets the waveform data from an acoustic scenario constructed from real-world, narrowband signals. By showing the IPUS components' functionality and their use of formal relationships between signal characteristics and SPA parameters, the example illustrates the important role that a formal theory of signal processing can play in signal interpretation.

7.1 Scenario Overview

Figure 12a shows the time-domain waveform (sampled at 8KHz) provided to the testbed, while Figure 12b shows how the sources in the scenario would appear using context-appropriate processing. `Phone-Ring` and `Siren-Chirp` are 1.2 times as energetic as `Buzzer-Alarm`, and `Glass-Clink` is an impulsive source 3.0 times as energetic as `Buzzer-Alarm`. Figure 12c shows how the events are distorted when the testbed's initial front-end configuration is applied throughout the scenario.

The testbed was initially configured to interpret waveform data in 1.0-second blocks, and to identify quickly any occurrences of `Siren-Chirp`. In particular, the system's SPA parameters were set to detect `Siren-Chirp`'s steady-energy behavior:

FFT-SIZE: 512
: *The number of uniformly-spaced frequency samples computed for each Short-Time Fourier Transform (STFT) analysis window position.*

WINDOW-LENGTH: 512
: *The number of data points to which each FFT in the STFT algorithm is applied ($\leq$ FFT-SIZE).*

DECIMATION: 512
: *The number of points between consecutive STFT analysis window positions. The value was set to 512 to permit the fastest possible processing of the data.*

PEAK-THRESHOLD: 0.09
: *Spectrum points with energy below this value are rejected by the peak-picking algorithm.*

For processing this example, the testbed's source database was loaded with models for the five narrowband sources shown in Figure 13. In the figure the sources' frequency components are labelled by single-frequency values only for clarity; the formal source definitions have frequency *ranges* specified for each component.

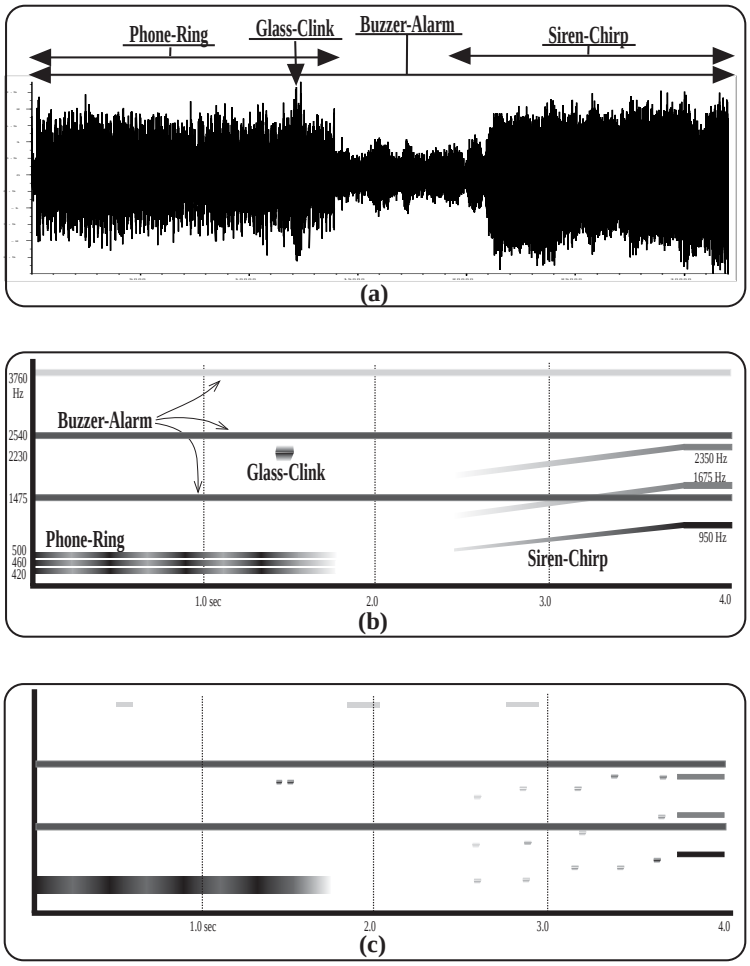

Figure 12: *Acoustic Scenario Events. Figure 12a shows the scenario's time-domain waveform. Figures 12b and 12c show the scenario's frequency-domain events. Darker shading indicates higher frequency-domain energy.*

There are several critical actions that the IPUS acoustic testbed must perform if it is to reasonably analyze Figure 12a's signal. In block 1, the testbed encounters two alternative interpretations of the data in the $[420, 500]$ frequency region. That is, there is the possibility that it could be caused by `Phone-Ring` or `Car-Horn`, or 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 `Car-Horn`'s low-energy microstream from being detected if in fact it were present. The second reason is that the frequency-sampling provided by the STFT algorithm's FFT-SIZE parameter does not provide enough frequency sample points to resolve the $[420, 500]$ region into `Phone-Ring`'s three microstreams. The

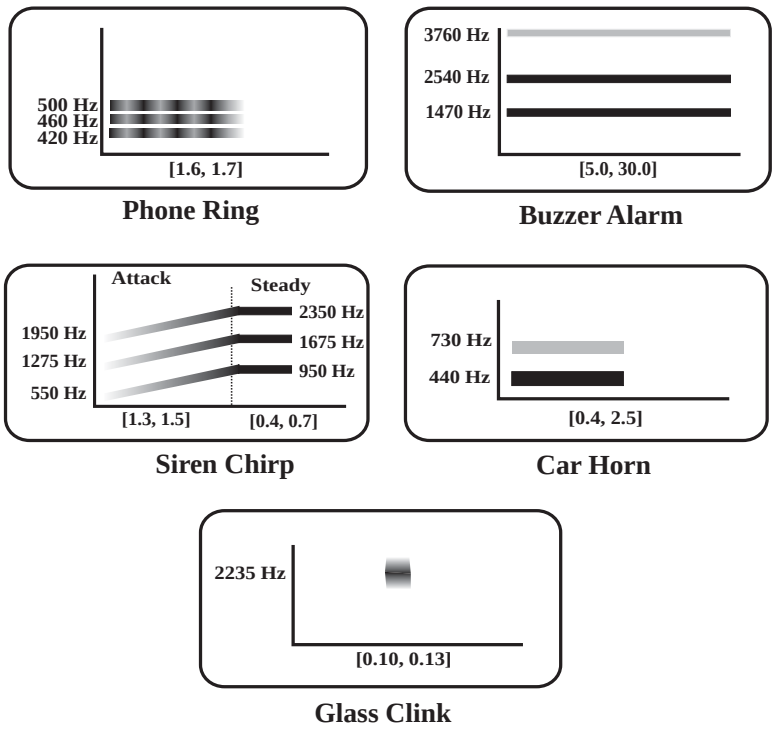

Figure 13: *IPUS Source Database. The vertical axis represents frequency and the horizontal axis represents time in seconds. The energy changes for each microstream are represented qualitatively by the shading gradations. Note that* `Phone-Ring` *is a ring from a phone different from the one in Figure 3.*

uncertainty in this situation is resolved through reprocessing under the direction of differential diagnostic reasoning, which increases resolution and decreases the energy threshold.

During block 1's analysis, the testbed also determines that `Buzzer-Alarm`'s track at 3760 Hz is missing. One reason for this is that the track's energy might be too low for the peak-picker's PEAK-THRESHOLD parameter setting. The discrepancy is resolved through reprocessing the previously-produced spectra with a lower PEAK-THRESHOLD value.

In block 2, the testbed detects a discrepancy between the outputs of its time-domain energy estimator SPA and its STFT SPA. The energy estimator SPA detects a substantial energy increase followed about 0.1 seconds later by a precipitous decrease. The STFT SPA, however, produces no significant set of peaks to account for the signal energy flux. This is because the algorithm's decimation parameter is too high. The testbed also detects a discrepancy between expectations established from block 1 for the $[420, 500]$ frequency region and the STFT

SPA's output. The STFT SPA produces short contours that cannot support the expected microstreams for `Phone-Ring` because of inadequate frequency sampling in the region. Both discrepancies are resolved by reprocessing. The first discrepancy is resolved through reprocessing with a smaller DECIMATION value and smaller STFT intervals, while the second is resolved through reprocessing with the finer frequency sampling provided by a 1024 FFT-SIZE.

In block 3, `Siren-Chirp`'s attack interacts with the poor time-resolution of the STFT SPA to produce a set of widely-separated short contours that the testbed cannot immediately interpret as the attack portion of microstreams. In block 4, however, the testbed uses the discovery of `Siren-Chirp`'s steady region as the basis for re-interpreting block 3's short contours as evidence for the sound's attack region.

7.2 Testbed Trace

The following is a high-level trace of the significant events that occurred as the system processed the signal in Figure 12a.

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. In deciding what evidence to examine first, the heuristics choose to look for any evidence in the steady-phase frequency regions of high-priority sources (`Siren-Chirp` in this case). No contours are found in these regions. At this point in the experiment, there are no other explicit source expectations.

- **Drive Unexplained Data to Higher Levels:** Contours in the $[1460, 1480]$ and $[2530, 2550]$ Hz regions are used to support microstream hypotheses. These in turn are used to support a `Buzzer-Alarm` source hypothesis. However, support for `Buzzer-Alarm`'s third microstream is not found in the peak-picker's correlates, causing a conflict discrepancy SOU to be posted with the source.

- **Discrepancy-Detection:** The testbed uses the heuristic that short contours[4] should not be used as microstream evidence. Because the block has a large number of short contours relative to the total number of contours detected, the testbed performs discrepancy detection to determine if there are tight short-contour clusters that could indicate distorted sources. The system finds such a cluster in the $[420, 500]$ Hz range, and then queries the source database to find a source hypothesis to explain the cluster. `Phone-Ring` and `Car-Horn` are retrieved because at least one of each source's frequency components overlaps the cluster. Therefore the testbed posts both sources 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 four SOUs have been posted: one each for the violation discrepancies associated with `Phone-Ring` and `Car-Horn` being supported by a cluster, one for the uncertainty associated with the existence of competing interpretations for the same cluster, and one for `Buzzer-Alarm`'s missing microstream. 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.

    1. **Differential Diagnosis:** 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. In this case the KS selects the low-energy, 900 Hz microstream of `Car-Horn` and the number of microstreams in the $[420, 500]$ Hz region for each source (`Phone-Ring` has 3, `Car-Horn` has 1) as discriminating features. It specifies that a lower energy-threshold be used to attempt to "bring out" `Car-Horn`'s low-energy microstream at 730 Hz. To attempt to find `Phone-Ring`'s three microstreams, it specifies an FFT-SIZE value of 1024 to increase the frequency sampling in the $[420, 500]$ Hz region. Note that the testbed at this time is not committed to either interpretation, nor to the possibility that **both** sources are present. Any decisions will wait for the results of reprocessing.
    2. **Differential Reprocessing:** The reprocessing KS is executed and the sought-after `Car-Horn` microstreams are not found. However, *three* well-

[4]Contours having between 1 and 3 peaks. 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.

defined contours are found in the [420, 500] Hz range that can support `Phone-Ring`'s microstreams. Therefore `Phone-Ring`'s belief is increased, while `Car-Horn`'s belief is decreased. `Car-Horn`'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 `Phone-Ring`'s violation-discrepancy uncertainty.

- **Handle Selected Uncertainties: (continued)** Focusing heuristics now select the conflict discrepancy SOU of `Buzzer-Alarm`'s missing microstream for resolution. This is handled through calling the discrepancy diagnosis KS and executing a reprocessing plan based on its explanation.

  1. **Discrepancy Diagnosis:** The diagnosis KS produces the explanation (MS-ENERGY-THRESHOLDING) for the discrepancy. That is, peak-picker SPA's PEAK-THRESHOLD parameter has a value too high to detect enough peaks to generate long contours for the microstream.
  2. **Discrepancy Reprocessing:** The reprocessing KS uses the explanation to decide to reprocess spectra from the entire block with a peak-picker SPA having a reduced PEAK-THRESHOLD value of 0.04. This produces seven peaks in the [3750, 3770] Hz region, which create a significant-length contour. This contour's existence resolves the conflict discrepancy. `Buzzer-Alarm`'s 3760 Hz microstream is annotated with a support specification that indicates that very short (one peak) contours or none at all are acceptable evidence as long as the PEAK-THRESHOLD value is higher than 0.04.

- **Define Expectations:** Because `Phone-Ring`'s description indicates that its steady region is approximately 1.7 seconds long, and at most 1.0 second has been found, an explicit expectation for `Phone-Ring`'s microstreams is posted for block 2's time period. Explicit expectations for the continuation of `Buzzer-Alarm`'s microstreams are also posted for block 2.

7.2.2 BLOCK 2

- **Bottom-Up Processing:** Bottom-up processing creates spectra and contours for block 2. `Glass-Clink` emits a high-energy, short-duration (0.12 sec) signal burst. 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. 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.

- **Seek Evidence for Current Expectations:** Since the duration of the fault discrepancy indicates that it is not related to `Siren-Chirp`,[5] the focusing heuristics act on `Siren-Chirp`'s priority and decide to examine data found in the source's expected frequency regions. No contours are found in these regions.

- **Handling Selected Uncertainties:** The testbed's focusing heuristics select fault-type SOUs for resolution before the control plans apply any interpretation KSs that might handle frequency regions affected by fault discrepancies. Thus, before the components of any non-priority expected sources are searched for, the fault discrepancy is selected for handling by the focusing heuristics. For this SOU, the control plans specify a strategy that executes discrepancy diagnosis followed by reprocessing.

    1. **Discrepancy Diagnosis:** The diagnosis KS explanation for the fault discrepancy is (CONTOUR-TIME-RESOLUTION). That is, the STFT decimation is too high to detect enough peaks to generate contours of significant length to account for the signal energy increase.
    2. **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 an STFT SPA having a 256-point WINDOW-LENGTH, a 512-point FFT-SIZE, and a 192-point DECIMATION. This produces four peaks in the $[2230, 2240]$ Hz region, which create a significant-length contour. This contour's existence resolves the fault discrepancy.

- **Seek Evidence for Current Expectations:** At this point, the focusing heuristics decide to gather evidence for explicit source expectations. Contours found in the expected regions of `Buzzer-Alarm` support that source'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. As happened in block 1, the front-end processing parameters produce a cluster of short contours in the

[5] `Siren-Chirp`'s duration is much longer than the fault's.

[420, 500] Hz range. The testbed's short-contour heuristic leads to a lack of support for the persistence of `Phone-Ring`'s microstreams into block 2.

- **Discrepancy Detection:** The testbed checks for conflict and violation discrepancies. The lack of support for `Phone-Ring`'s microstreams raises a conflict discrepancy.[6] No violation discrepancies are found.

- **Handle Selected Uncertainties: (continued)** The focusing heuristics select the conflict SOU in `Phone-Ring`'s three microstreams for resolution. Control plans specify a strategy of discrepancy diagnosis followed by reprocessing.

    1. **Discrepancy Diagnosis:** The discrepancy detection KS returns the explanation (COARSE-FREQUENCY-SAMPLING); the STFT analysis was done with inadequate frequency sampling, causing the three microstreams to appear as the contour cluster actually observed. The KS also returns a support specification that in the next block under the same initial parameter settings, `Phone-Ring`'s microstreams will appear like the contour cluster again. In this scenario the support specification will not be useful, however, since based on `Phone-Ring`'s maximum possible duration the microstreams should not extend into block 3.[7]
    2. **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 ($2 * 512 = 1024$ in this case). The doubling of 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 0.12-second contour is found to match `Glass-Clink`'s characteristics. A hypothesis for the source is therefore posted.

---

[6] Remember that differential diagnosis does not annotate hypotheses with support specifications (see section 6.1.4). Thus, `Phone-Ring`'s microstreams do not have specifications to prevent the testbed from registering the contour cluster as a distortion.

[7] A shortcoming of configuration *C.1* is that support specifications are not propagated across periodic streams. Thus, the support specification will not even be useful for any future rings. We are correcting this problem in configuration *C.2*.

- **Define Expectations:** Because `Buzzer-Alarm`'s model indicates that its steady behavior could continue for 3 to 28 more seconds, an explicit expectation for its continuation is posted for Block 3's time period. No expectation for `Phone-Ring` is posted because its model specifies a maximum duration of 1.7 seconds.

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:** `Siren-Chirp`'s frequency regions are examined for contours. Some short contours are present in this block from the source's attack phase, but because the testbed first recognizes sources by steady characteristics (due to their more predictable behavior), their presence does not cause the creation of a `Siren-Chirp` source hypothesis. Contours extending source `Buzzer-Alarm`'s microstreams are sought for and found.

- **Drive Unexplained Data to Higher Levels:** Because of their short lengths, the contours caused by `Siren-Chirp`'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 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:** An expectation for `Buzzer-Alarm`'s microstreams to continue into block 4 is posted.

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:** The testbed first searches `Siren-Chirp`'s frequency regions for contours. Contours supporting the source's steady region are detected, and a source hypothesis is posted. The testbed also finds contours to support `Buzzer-Alarm`'s microstreams.

- **Handle Selected Uncertainties:** Because its attack region is unsupported, `Siren-Chirp`'s confidence level is low. Due to `Siren-Chirp`'s priority, the

focusing heuristics decide to resolve this missing-support SOU. The control plans specify a strategy of accepting sets of short contours that reflect the slope of the chirp when grouping peaks into contours. No diagnosis is performed; the reprocessing is simply a context-dependent interpretation strategy for detecting chirps when their presence is suspected.

1. **Reprocessing:** To find "enough" (60% in this case) of `Siren-Chirp`'s attack region, the testbed must search back into block 3 and reinterpret the previously-detected but unrecognized short contours as valid attack-region contours. `Siren-Chirp`'s attack region and its chirp characteristics are identified in the previous block's signal data[8]. At this point `Siren-Chirp` is determined to be present with high confidence.

At the end of the scenario the testbed had recognized all the sounds and had tracked at least 85% of each sound's duration. There were no false-alarm sound hypotheses. However, there was one false-alarm discrepancy, which, for purposes of clarity, was omitted from the trace. In block 3 the testbed's fault-detection claimed that another discrepancy between the STFT and energy-estimator outputs had occurred. The focusing heuristics did select the associated SOU for handling, but in the course of reprocessing in the same manner as in block 2, no new peaks were found. Thus, the discrepancy was disproven.

This detailed trace shows how the architecture components can implement a dual search to find (1) SPAs appropriate to a scenario with real-world sounds and (2) interpretations appropriate to the SPAs' correlates. The components' activation rates in the trace should not, however, be taken as a measure of their individual utilities in the problem of complex signal interpretation. To determine these utilities, our current work is focused on developing two statistical models. One relates acoustic scenario complexity to distortion rates, and the other relates distortion rates to architecture component activation rates. It is our hope that these models not only will determine each IPUS component's utility for various classes of scenarios but also will generate recognition-rate benchmarks for perceptual systems that do not use various IPUS components.

8 IPUS and SPA Design

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.

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

The motivation for this design paradigm has been that such SPAs could provide precise information that would efficiently constrain interpretation search and produce interpretations with low uncertainty. 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. This premise, however, does not appear appropriate for perceptual systems operating in complex environments [16]. Since in such domains the SPA assumptions will often be violated, it seems unreasonable to devote computational resources to the extraction of detailed and precise information that is likely to be misleading.

The IPUS architecture 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. IPUS provides a framework for using such SPAs in strategies where 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 [33] 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.

9 Future Research

In addition to our work on designing new SPAs and on developing statistical relationships among scenarios, distortion rates, and IPUS components' effectiveness, we are extending our testbed's control plans to explore the issue of scaling. Specifically, we are investigating the use of approximate processing and model-learning.

In configuration *C.2*, which is currently under development, the testbed control plans have been changed to accomodate a larger library of 35 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.

The frequency features of the sound models used in the testbed trace were hand-

crafted in a time-consuming process. When dealing with environments with large numbers of signal objects, it will be desirable to automate the model-acquisition process. The construction of these models will require the identification of features that avoid distortions caused by SPAs and/or model interactions as much as possible. Research is being done on incorporating the IPUS reprocessing loop into a framework for learning acoustic source models [3].

On initial consideration, it might seem that the time required by multiple reprocessings under IPUS would be unacceptably high in noisy environments. However, because traditional systems *continuously* sample several front-ends' data while IPUS-based systems *selectively* sample several front-end processings' data, the IPUS paradigm should decrease the expected processing time for contexts requiring several independent processing views. We are working on verifying this claim.

10 Summary

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 features' 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 bidirectional 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 SPA application strategies. A major contribution of the architecture is to formalize and unify front-end SPA reconfiguration performed for interpretation processes (e.g. differential diagnosis) with that performed for data correlate refinement. under

discrepancy diagnosis. This results in a single reprocessing concept driven by the presence of SOUs.

Our sound understanding testbed experiments indicate that the basic functionality of the architecture's components and their interrelationships are realizable. We believe the IPUS architecture is applicable to any signal understanding domains for which the SPAs have a rich underlying theory. This view is supported by the similarities shared between the testbed's acoustic domain theory and that of many other signal domains such as sonar [39], weather radar [9], music [26], and biomedical signals [14].

In conclusion, 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 configuration 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 that must interact with complex environments.

11 Acknowledgments

We would like to acknowledge Norman Carver and Izaskun Gallastegi for their roles in designing and implementing the IPUS architecture's control framework and evidential reasoning capabilities; Malini Bhandaru and Zarko Cvetanović for their contributions to the testbed's early implementation stages; Erkan Dorken for his significant contribution to designing the testbed's SPAs; and Ramamurthy Mani for his help in creating the acoustic database for testbed experiments.

This work was supported by the Rome Air Development Center of the US Air Force Systems Command under contract F30602-91-C-0038, and by the Office of Naval Research under contract N00014-92-J-1450. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] Bell, B. and Pau, L. F., "Context Knowledge and Search Control Issues in Object-Oriented Prolog-Based Image Understanding," *The Proceedings of the 1990 European Conference on Artificial Intelligence*, 1990.

[2] Bell, B. and Pau, L. F., "Contour Tracking and Corner Detection in a Logic Programming Environment," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, pp. 913–917, August 1990.

[3] Bhandaru, M. K., Draper, B. A., and Lesser, V. R., "Learning Image to Symbol Conversion," pp. 6-9, AAAI Technical Report FS-93-04, 1993.

[4] Bitar, N., Dorken, E., Paneras, D., and Nawab, S. H., "Integration of STFT and Wigner Analysis in a Knowledge-Based Sound Understanding System," *The Proceedings of the 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. IV, , pp. 585–588, San Francisco, March 1992.

[5] Bobick, A. F. and Bolles, R. C., "The Representation Space Paradigm of Concurrent Evolving Object Descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 146–156, February 1992.

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

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

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

[9] Campbell, S. D. and Olson, S. H., "WX1 – An Expert System for Weather Radar Interpretation," in *Coupling Symbolic and Numerical Computing in Expert Systems*, J. S. Kowalik, ed., Elsevier Science Publishers, B. V. (North-Holland), 1986.

[10] Carver, N. and Lesser, V., "The Evolution of Blackboard Control," *Expert Systems with Applications*, Special Issue on The Blackboard Paradigm and Its Applications, vol. 7, no. 1, pp. 1–30, 1994, (also available as Technical Report 92-71, Computer Science Department, University of Massachusetts, 1992).

[11] Carver, N. and Lesser, V., "A Planner for the Control of Problem-Solving Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Special Issue on Planning, Scheduling and Control, vol. 23, no. 6, pp. 1519–1536, November/December 1993.

[12] Carver, N. and Lesser, V., "A New Framework for Sensor Interpretation: Planning to Resolve Sources of Uncertainty," *The Proceedings of the 1991 National Conference on Artificial Intelligence (AAAI-91)*, pp. 724–731, Anaheim, California, July 1991.

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

[14] Dawant, B. and Jansen, B., "Coupling Numerical and Symbolic Methods for Signal Interpretation," *IEEE Transactions on Systems, Man and Cybernetics*, pp. 115–124, Jan/Feb 1991.

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

[16] Dorken, E., "Approximate Processing and Knowledge-Based Reprocessing of Non-Stationary Signals," PhD Thesis, Electrical, Computer, and Systems Engineering Dept, Boston University, 1993.

[17] Dorken, E., Nawab, S. H., and Lesser, V., "Extended Model Variety Analysis for Integrated Processing and Understanding of Signals," *The Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. V, pp. 73-76, San Francisco, California, March 1992.

[18] Dove, W., *Knowledge-Based Pitch Detection*, PhD Thesis, Electrical Engineering and Computer Science Dept., MIT, 1986.

[19] Draper, B. A., Hanson, A. R., and Riseman, E. M., "Learning Blackboard-Based Scheduling Algorithms for Computer Vision," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 2, pp. 309–328, April, 1993.

[20] Erman, L., Hayes-Roth, F., Lesser, V., Reddy, D., "The Hearsay II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys 12*, vol. 2, pp. 213–253, June 1980.

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

[22] Grimson, W. E. L. and Lozano-Pérez, T., "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 3–36, 1984.

[23] Hayes-Roth, B., Washington, R., Hewett, R., Hewett, M., and Seiver, A., "Intelligent Monitoring and Control," *The Proceedings of the 1989 International Joint Conference on Artificial Intelligence*, pp. 243–249, Detroit, Michigan, August 1989.

[24] Hudlická, E. and Lesser, V., "Modeling and Diagnosing Problem-Solving System Behavior," *IEEE Transactions on Systems, Man and Cybernetics*, Special Issue on Diagnostic Reasoning, vol. 17, no. 3, pp. 407–419, May/June 1987.

[25] Hudlická, E. and Lesser, V., "Meta-Level Control Through Fault Detection and Diagnosis," *The Proceedings of the 1984 National Conference on Artificial Intelligence (AAAI-84)*, pp. 153–161, Austin, Texas, July 1984.

[26] Katayose, H., Kato, H., Imai, M., and Inokuchi, S., "An Approach to an Artificial Music Expert," *Proceedings of the 1990 International Computer Music Conference*, pp. 139–147, 1990.

[27] Klassner, F., Lesser, V., and Nawab, S. H., "Fusing Multiple Reprocessings of Signal Data," *The Proceedings of the SPIE Sensor Fusion VI Conference*, vol. 2059, Boston, Massachusetts, September 1993.

[28] Klassner, F., *Data Reprocessing and Assumption Representation in Signal Understanding Systems*, Technical Report 92-52, Computer Science Department, University of Massachusetts, 1992.

[29] Kohl, C., Hanson, A., and Reisman, E., "A Goal-Directed Intermediate Level Executive for Image Interpretation," *The Proceedings of the 1987 International Joint Conference on Artificial Intelligence*, pp. 811–814, Milan, Italy, August 1987.

[30] Lesser, V., Nawab, S. H., Gallastegi, I., and Klassner, F., "IPUS: An Architecture for Integrated Signal Processing and Signal Interpretation in Complex Environments," *The Proceedings of the 1993 National Conference on Artificial Intelligence (AAAI-93)*, pp. 249–255, Washington, DC, July 1993.

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

[32] Maksym, J. N., Bonner, A. J., Dent, C. A., and Hemphill, G. L., "Machine Analysis of Acoustical Signals," *Issues in Acoustic Signal/Image Processing and Recognition*, C. H. Chen, ed, NATO ASI Series, vol. F1, Springer-Verlag, pp. 95–112, 1983.

[33] Nawab, S. H. and Dorken, E., "Efficient STFT Computation Using a Quantization and Differencing Method," *The Proceedings of the 1993 IEEE Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 587–590, Minneapolis, Minnesota, April 1993.

[34] Nawab, S. H. and Lesser, V., "Integrated Processing and Understanding of Signals," chapter 6, *Knowledge-Based Signal Processing*, pp. 251–285, A. Oppenheim and H. Nawab, eds., Prentice Hall, New Jersey, 1991.

[35] Nawab, S. H. and Lesser, V., "High-Level Adaptive Signal Processing", *Northeast AI Consortium Final Report*, Technical Report RADC-TR-90-404, vol. 17, Rome Laboratories, Griffiss Air Force Base, NY, 13441-5700, Dec 1990.

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

[37] Nawab, S. H., Lesser, V., and Milios, E., "Diagnosis Using the Underlying Theory of a Signal Processing System," *IEEE Transactions on Systems, Man and Cybernetics*, Special Issue on Diagnostic Reasoning, vol. 17, no. 3, pp. 369-379, May/June 1987.

[38] Newell, A. and Simon, H., "GPS: A Program that Simulates Human Thought." *Computers and Thought*, Feigenbaum and Feldman, eds., McGraw-Hill, pp. 279–293, 1969.

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

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

[41] Peng, Y. and Reggia, J. "Plausibility of Diagnositic Hypotheses: The Nature of Simplicity," *The Proceedings of the 1986 National Conference on Artificial Intelligence (AAAI-86)*, pp. 140–145, Philadelphia, Pennsylvania, July 1986.

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

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

[44] Williams, M., "Hierarchical Multi-Expert Signal Understanding," *Blackboard Systems*, R. Engelmore and A. Morgan, eds., Addison-Wesley, 1988.