Integrated Signal Processing and Signal Understanding[1]

Victor Lesser, Hamid Nawab[†],
Malini Bhandaru, Norman Carver,
Zarko Cvetanovic, Izaskun Gallastegi,
Frank Klassner

COINS Technical Report 91-34
November 1991

[†]Electrical and Computer Engineering Dept.
Boston University
44 Cummington Street
Boston, Massachusetts 02125

Abstract

This report outlines the IPUS paradigm, named for *Integrated Processing and Understanding of Signals*, which permits sophisticated interaction between theory-based problem solving in signal processing and heuristic problem-solving in signal interpretation. The need for such a paradigm arises in signal understanding domains that require the processing of complicated interacting signals under variable signal-to-noise ratios. One such application is sound understanding, in the context of which we report on a testbed experiment illustrating the functionality of key IPUS architecture components.

[1]This work was supported by the Office of Naval Research under University Research Initiative grant number N00014-86-K-0764, by the National Science Foundation under NSF-CER contract DCR-8500332, and by the Rome Air Development Center of the Air Force Systems Command under contract F30602-91-C-0038.

1 Introduction

In traditional signal understanding systems [8, 17], the front-end signal processing is usually fixed for all input signals, and these signals are not re-processed on the basis of higher-level problem-solving dynamics. Thus, the interaction between the interpretation problem-solving and the signal processing is limited to a sequential scheme in which the former accepts the latter's output data. Some recent systems [7, 9, 10, 5] have used architectures in which the signal processing is not immutable and can be affected by the results of interpretation activity. However, the interaction between signal processing and higher-level interpretation has been limited in these systems.

In contrast, we have developed a paradigm and implemented it in an architecture to *structure* the interaction between processing and interpretation in a more general way, with an emphasis on using the signal processing theories that often underlie signal processing tasks. We call this paradigm IPUS for *Integrated Processing and Understanding of Signals*, reflecting the fact that in this paradigm the search for appropriate signal processing algorithms to use is intimately tied to the search for the correct interpretation of the signal processing output data.

The need for a paradigm such as IPUS arises in applications where the situation-dependent nature of signal processing requirements leads to a combinatorial explosion in the number of different signal processing algorithms that a signal understanding system must have at its disposal. One such application is sound understanding [14], which involves real-time processing of acoustic signals in order to determine the types of sound sources (such as telephones, crying infants, household appliances, etc.) that may have generated those signals. Specific instances of sound understanding include robotic hearing and speech recognition in environments with non-speech background sound sources. The complexity of the signal processing requirements in the sound understanding problem is due to two factors:

1. The need to process a large variety of signal types due to the situation-dependent nature of the input. For example, a sound understanding system has to deal with input data obtained from harmony sources, impulsive sources, frequency-modulated signals, and combinations thereof. This variety is further increased by the presence of variable noise levels.

2. The need to change processing goals in a context-dependent way. For example, a signal understanding system might have as its main goal to respond to either the sounds of an infant or a ringing telephone and to ignore other sound sources. If an infant sound is detected, the system's main goal may then switch to determining whether the infant is crying or choking while ignoring any telephone rings.

For these reasons, it is impractical to design a single mathematically derived signal processing algorithm (SPA) to be applied to all possible input signals to produce the desired information for each input. There are too many types of input signals, each type best suited to a different kind of mathematical formulation. Additionally, the signal understanding system's current high-level goals and assumptions about the input signal nature may dictate any one of several mathematical formulations for the desired information itself. We therefore advocate the approach of having a database of signal processing algorithms at the disposal of the signal understanding system from which it should search for the most appropriate algorithms to use in a context-dependent fashion. As the monitored environment's signal changes, higher-level interpretation processes should be able to reconfigure numeric-level processing.

In the remainder of this paper, we first highlight the IPUS architecture and its components. Second, we describe related interpretation systems. Third, we present an example to illustrate the operation of our sound understanding testbed which is based on IPUS. Fourth, we discuss the primary components of the IPUS architecture. To conclude, we present some remarks on the research issues that have arisen in the context of the IPUS paradigm.

2 The IPUS Architecture

The starting point of the IPUS architecture's design is its database of signal processing algorithms (SPA). The SPA database contains a generic SPA for each class of algorithms available to IPUS. This organization relies on the fact that signal processing theory often supplies generic SPA with adjustable parameters. An SPA instance is created by the specification of particular values for the parameters, and has capabilities and limitations stemming from those parameter values.

As an example of this instantiation concept, consider the class of short-time Fourier transform (STFT) algorithms [13], which can be used for time-dependent frequency analysis of signals. An instance in this class can be specified by particular values for its parameters, such as an analysis-window length (the number of signal data points analyzed at a time), a frequency-sampling rate, a temporal decimation factor (the overlap between two consecutive analysis windows), etc. Each STFT instance differs from the others because of the assumptions they make about the spectrum characteristics of the input signal and the manner in which these characteristics vary as a function of time. One instance may have the ability to provide excellent frequency resolution for signals whose primary components' frequencies remain constant over time, but poor time resolution for signals whose components quickly shift within the frequency spectrum over time. The Heisenberg Uncertainty Principle implies that one cannot obtain an STFT SPA instance (or, for that matter, design a new algorithm class) which simultaneously provides high frequency resolution and high time resolution.

The database organization links the search for SPA instances appropriate to a particular situation (i.e., the current set of assumptions about the signal being processed) with the search for appropriate interpretations. As a result, theoretical relationships between control parameters and SPA performance characteristics can be exploited by *both* search processes.

Suppose a particular input signal is to be processed by a generalized SPA. To select appropriate parameter values, the system must consider its current goals as well as knowledge about characteristics of the particular input signal. This leads to the dilemma that choosing the appropriate control 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, the search for appropriate signal interpretations is intimately connected with the search for appropriate SPA instances. At the heart of the IPUS architecture lies an iterative technique [14] for converging to the appropriate SPAs and parameter values. The following description is meant to serve as a summary of this iterative technique's stages. Section 5 provides a more detailed view of each stage.

The technique begins by using the best available guess for the control parameter values to process the input signal (in the worst case, this is a set of arbitrary values). The SPA output is then analyzed by a *discrepancy detection* mechanism to test for the presence of distorted SPA output data. If discrepancies are detected, a *diagnosis* is then performed to obtain an "inverse" mapping from the detected discrepancies and SPA parameter settings under which they were observed to qualitative hypotheses that explain the distortions. This diagnosis process uses the *formal theory* that underlies the signal processing carried out by the signal understanding system. The theory relates SPA parameter settings to the occurrence of specific distortions. The availability of such a formal theory is a major criterion for determining the IPUS architecture's applicability to any particular problem domain. A *signal re-processing* phase then proposes and executes a search plan for finding a new set of values for the generic SPA(s)' control parameters that eliminates or reduces the hypothesized distortions. In the course of this plan's execution, the signal data may be reprocess several times under different SPAs with different parameter values. Each time the data is reprocessed, a new parameter-value state in the SPA control parameter search space is examined and tested for how well it eliminates or reduces distortions.

The discrepancy detection process is crucial to the IPUS framework's iterative approach. We have specified that the process not only detect discrepancies but also categorize them to permit a system to choose actions based upon their severity or importance to the current processing scenario. The idea behind our current categorization is that when SPA output data is distorted, a signal understanding system must be able to detect discrepancies between this data and one or more of the following:

1. the *expected* form of the SPA output data. Such discrepancies are termed *conflicts*. There are two types of conflicts: *complete* and *partial*. The former indicates that there is total disagreement between the SPA output data and an expectation, while the latter indicates that there is a partial match. An example of a complete conflict occurs when the interpretations of past data

show two sinusoids at 200 Hz and at 250 Hz with no decline in their amplitudes and the current SPA output data contains neither of the sinusoids. A case where a partial conflict would be raised is when current data contained two out of three frequencies that supported the identification of a telephone ring, and after a search for the other frequency the system couldn't find it.

2. the *output data* from other signal processing algorithms applied to the same underlying signal data. Such discrepancies are termed *faults*. For example, suppose that the signal data is being processed with a zero-crossing analyzer and an STFT. If the zero-crossing analyzer were to indicate the presence of a sinusoidal signal but the STFT does not, a fault would be declared.

3. the *entire allowable* class of input signals for the application domain. Such discrepancies are termed *violations*. A violation occurs when the SPA output data has characteristics that are *a-priori* known to be absent in the entire class of possible signals in the application domain. For example, if the application domain is known to consist only of signals with frequencies below 500 Hz, SPA output data showing a signal at 700 Hz would give rise to a violation.

Depending upon the class(es) of discrepancies detected and the context in which interpretation is being carried out, the system is expected to use different strategies to resolve (i.e., explain and possibly eliminate) the distortion. For example, in a situation where real-time processing deadlines are tight, the system may not even attempt to resolve partial-conflict discrepancies in order to conserve time. In another less time-critical situation 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 IPUS may decide that the successful *generation* of an explanation alone is sufficient to resolve the discrepancy. Finally, in a non-time-critical situation, 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.

Figure 1 illustrates the IPUS architecture's high-level organization. The signal data and the interpretation hypotheses derived from that data are stored on a blackboard with hierarchical information levels. The blackboard hypotheses fall into two basic categories: those posted to explain the signal data and those posted to specify expectations about the signal data nature.

The IPUS architecture is designed to serve as the basis of signal understanding systems that are driven by the goal of producing interpretations whose associated uncertainties have been reduced to "acceptable" levels, rather than by the goal of producing perfect interpretations of the data they examine. Therefore, control in IPUS-based systems requires some formalism for representing factors that can affect the levels of confidence in their interpretations. The control mechanism must be able to focus on particular uncertainties in a situation-dependent manner and

Figure 1: *IPUS Architecture.*

must support problem-solving strategies for reducing these uncertainties in a timely, yet intelligent, manner.

To meet this specification, IPUS uses the RESUN[2, 3] framework to control knowledge source[2] execution. RESUN views interpretation as a process of gathering evidence to resolve sources of uncertainty in interpretation hypotheses. The framework uses an explicit symbolic representation for the sources of uncertainty (SOU) in the various hypotheses. These SOUs are structures used by system control mechanisms to select problem-solving strategies for interpretation. Problem-solving is driven by the information maintained in a structure called the *problem-solving model*, which provides a summary of the current interpretation of data as well as the SOUs associated with each hypothesis. The interpretation process is controlled by an incremental reactive control planner that uses *control plans* and *focusing heuristics*. Control plans are schemas that define the interpretation methods and

[2]This is a term from blackboard technology referring to a code module which encapsulates expert knowledge for a particular domain.

information gathering actions (e.g., signal processing algorithms) available to the system for processing and interpreting data, and for resolving interpretation uncertainties. Focusing heuristics control the selection of which SOUs to resolve and/or which processing strategy to pursue next when there are several possibilities.

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.

3 Related Work

As stated earlier, some recent systems [7, 9, 10, 5] have begun to explore the interaction between interpretation activity and numeric-level signal processing. For example, Hayes-Roth [9] incorporates an input-data management component that controls the sampling rate of signals in response to overall system time and workload constraints. This is somewhat ad-hoc interaction, since it is based on system reasoning-time requirements alone, and it works primarily because the signals monitored are relatively simple in nature: heart-rate, temperature fluctuations, etc. The model of interaction does not appear adequate for signals containing complex structures that must be modeled over time.

Dawant [5] uses a more general approach in which signal interpretation knowledge is separated from signal processing knowledge, yet can guide the re-application of the signal processing knowledge. However, the system control appears highly goal-directed and employs a limited representation of model uncertainty (only three levels of certainty to characterize data matches with signal event models). Descriptions of the framework make it appear that it operates 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 source signal's *interaction* with the environment or unsuitable processing algorithms.

In GOLDIE [10], Kohl describes an image segmentation system that permits 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 can engage in a "hypothesize-and-test" search strategy 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 does not incorporate a centralized 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, published accounts of the system indicate it would not attempt to diagnose the discrepancy and exploit this information to reformulate the algorithm search but would select the next highest rated algorithm and proceed.

In view of the adaptive nature of the IPUS architecture, it is important to distinguish between the IPUS approach and the classic adaptive control theory approach [19]. 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 symbolic and 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 may thus 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.

4 Testbed Example

4.1 Testbed and Test Scenario Background

We have implemented a testbed in order to test the interpretation architecture's functionality in the context of a sound understanding application. The testbed runs on a TI Explorer II+ and is implemented in approximately 1400Kb of source code. All signal processing algorithms are implemented in software.

The testbed consists of a blackboard with six evidence abstraction levels and knowledge sources for tasks such as discrepancy detection, diagnosis, signal reprocessing planning and execution, and inferring hypotheses between different abstraction levels. Figures 2 to 7 provide a short description of the information represented in the six abstraction levels.

It could be argued that the first three levels contain information generated by numeric-level signal-processing criteria such as energy thresholds, while the last three levels contain information generated by higher-level, more interpretive, psychoacoustic criteria such as source timbre or source-sequencing (i.e., footsteps often occur after a phone ring is heard). Elevate that distinction in the contributing sources of abstraction levels to an ironclad rule, however, would miss the point of the IPUS architecture, which is that strict separation of high-level interpretation and numeric-level signal processing is undesirable. As we will show in the following

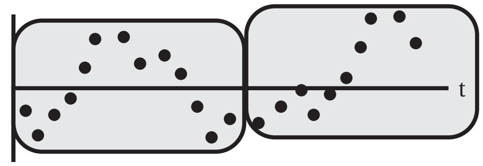

Figure 2: SEGMENT LEVEL: *A segment is a collection of raw data points for which such time-domain statistics like zero-crossing density, average energy, etc, are maintained. Numeric-level SPA's operate on one segment at a time.*

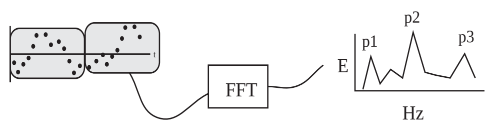

Figure 3: SPECTRUM LEVEL: *The second level consists of spectrum hypotheses derived for each waveform segment through Fourier transforms and peak-picking algorithms.*

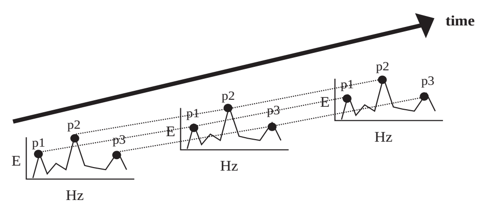

Figure 4: CONTOUR LEVEL: *The third level consists of contour hypotheses, each of which corresponds to a sequence of time-ordered peaks (each peak from a different segment) that are within specified time, frequency, and amplitude radii of their sequence neighbors.*

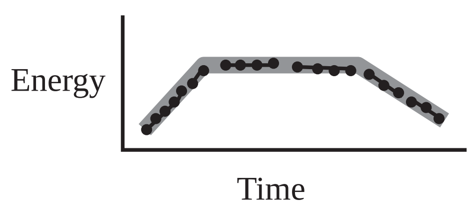

Figure 5: MICROSTREAM LEVEL: *The fourth evidence abstraction level contains microstream hypotheses supported by one contour or a sequence of contours in time. Each microstream has an energy pattern consisting of an attack region (signal onset – increasing energy), a steady region, and a decay region (signal fadeout – decreasing energy).*

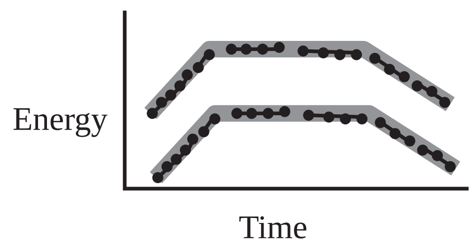

Figure 6: STREAM LEVEL: *Groups of microstreams synchronized according to time and/or some psychoacoustic criteria (e.g., harmonic sets, frequency separation) support stream hypotheses in the fifth level.*

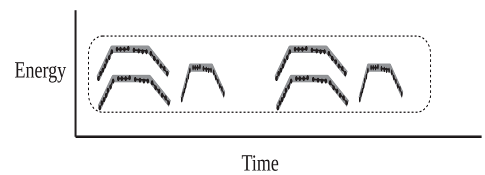

Figure 7: SOURCE LEVEL: *At the sixth level, sequences of stream hypotheses are used to support sound-source hypotheses.*

example, the architecture produces an intimate, yet structured, interaction between the two levels of processing.

For the initial system functionality tests, the IPUS source database contained 5 narrowband sources (A, B, C, D, and E) whose frequency-energy behaviors are illustrated in Figure 8. The shaded regions indicate signal strength (e.g., volume or energy) as a function of time. The sources' frequency components are labelled by single-frequency values for ease of display. In the formal source definitions, frequency *ranges* are specified for each component.

Figure 9 shows the time-frequency display of the streams (A, B, C, and E) in a signal produced by the testbed signal simulator for one of the system functionality tests. In relative energy terms, the two sources A and C are 1.2 times as energetic as source B. Source E is an impulsive source with acoustic energy 5 times that of source B.

For this particular experiment, the testbed was configured to interpret waveform data in 1.0-second blocks, and was directed to identify quickly any occurrences of source C. The system's front-end signal processing parameters are set to detect source C's steady-energy behavior. The pertinent SPA parameters and their initial values in this experiment are:

FFT-SIZE: 1024

The number of uniformly-spaced samples from DTFT used to model spectra.

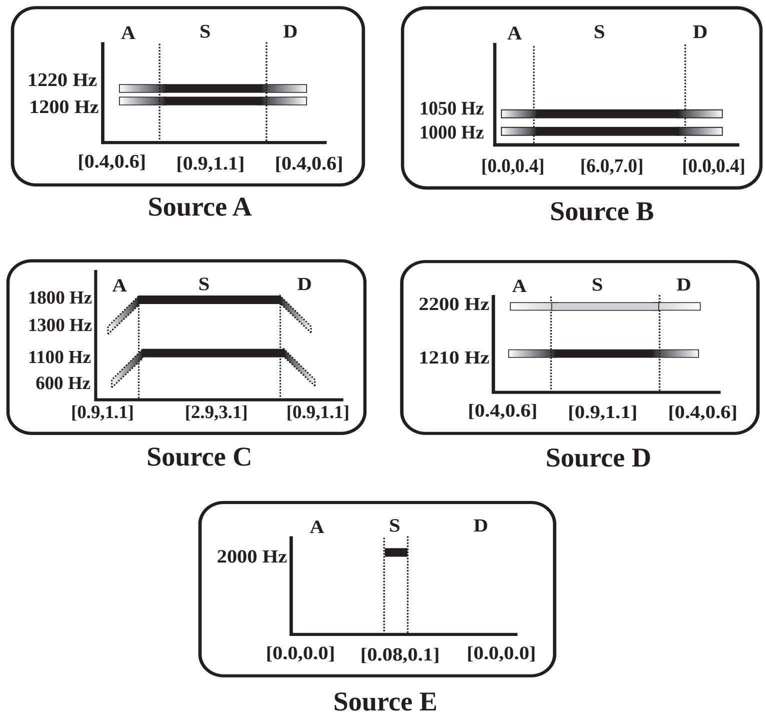

Figure 8: *IPUS Source Database. The vertical axes represent frequency and the horizontal axes represent time. The energy-level changes for each microstream are represented qualitatively by the shading gradations.*

STFT-INTERVAL: 1024

The number of data points to which each FFT in the Short-Time Fourier Transform (STFT) algorithm is applied (≤ FFT-SIZE). This determines the length of segments.

STFT-OVERLAP: 0

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

STFT-PEAK-ENERGY-THRESHOLD: 0.2

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

ABSOLUTE-NOISE-THRESHOLD: 0.001

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

SAMPLING-FREQUENCY: 10KHz

Rate at which the data stream is sampled.

There are several critical actions that IPUS must perform if it is to reasonably analyze the scenario in figure 9. In block 1, IPUS encounters two alternative interpretations of the data in the [1200, 1220] frequency region. That is, there is the possibility that it it could be caused by source A or source D, or even both occurring simultaneously. One reason for this confusion stems from the fact that the energy threshold setting for the peak-picking algorithm is high and would prevent D's low-energy microstream from being detected if in fact it were being produced by the environment. The second reason is that the frequency-sampling provided by the STFT algorithm's FFT-SIZE parameter does not adequately resolve the data in the [1200, 1220] into source A's two microstreams. The uncertainty engendered by this event is resolved through reprocessing under the direction of differential diagnostic reasoning, which increases resolution and decreases the energy threshold.

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

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

4.2 IPUS Trace

The following is a high-level trace of the significant events that occurred as the system processed the scenario in figure 9. The Appendix (page 49) contains the actual trace output created by the IPUS testbed.

4.2.1 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

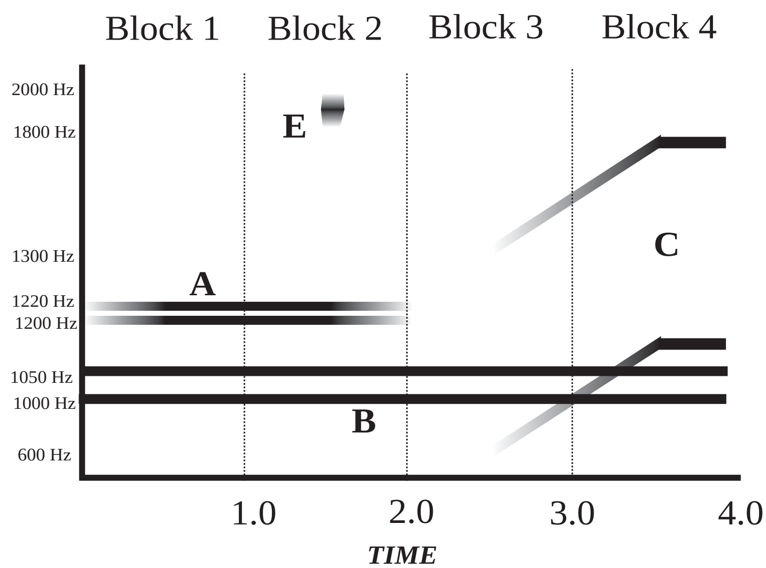

Figure 9: *Signal Events in Testbed Example. Shading indicates attack (light-to-dark) and decay (dark-to-light) regions.*

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 IPUS to act upon current high-level expectations and search for support evidence. At this point in the scenario, however, there are no explicit source expectations. In the source database, though, C's definition identifies it as a high priority source. This priority will impose an ordering on the data examined in the next step.

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

Discrepancy-Detection Clustering: The testbed uses the heuristic that short (two- or three-peak) contours should not be used as evidence for microstreams since short contours could be the result of random noise, and the system should apply as little computing time as necessary to the processing of noise.

Because a large number of short contours relative to the total number of contours is detected in the current block, IPUS performs discrepancy detection to determine if there are short-contour clusters that could indicate distorted sources. The system finds a high-population cluster in the $[1200, 1250]$ Hz range, and then attempts to find a source hypothesis to explain the stream. A query to the source database yields two possible explanations, source A and source D, because at least one of their frequency components overlaps the cluster frequency region. Therefore IPUS posts both sources' hypotheses as alternative explanations for the contour cluster. This use of short contours in place of long contours to support interpretations raises a violation discrepancy, since the a priori expectation that sources are indicated only by long contours is violated.

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

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

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

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

4.2.2 BLOCK 2

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

Fault Discrepancy Resolution: Before any source expectations' components are searched for, the fault discrepancy is selected for handling by the focusing heuristics[3]. The diagnosis KS explanation for this discrepancy is (CONTOUR-TIME-RESOLUTION). That is, the STFT overlap is too low to detect enough peaks to generate contours of significant length to account for the signal energy increase.

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

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

Discrepancy Detection: As occurred in block 1, the front-end processing parameters produce a cluster of short contours in the [1200, 1250] Hz range. Again,

[3] Since the duration of this discrepancy indicates that it is not related to C, the astute reader may wonder why C's high criticality did not override this choice and force IPUS to look for C data first. This is a shortcoming of the heuristic set used for this experiment and will be rectified in future heuristic sets.

IPUS relies on the heuristic that short contours not be used to support microstreams. Thus, the expected microstream extensions for source A are not found. This raises a conflict discrepancy.

Conflict Discrepancy Resolution: The discrepancy diagnosis KS takes as input a desired interpretation state containing the two microstreams and an observed interpretation state containing neither. The KS's means-ends analysis algorithm returns the explanation (COARSE-FREQUENCY-RESOLUTION). That is, the KS proposes that the STFT analysis was done with inadequate frequency sampling, causing the two microstreams to appear as the contour cluster actually observed. The KS also returns a "scenario-specification" indicating that if this diagnosis is correct, in the next block under the same initial parameter settings, A's microstreams will appear like the contour cluster again, and to avoid spending time diagnose the discrepancy, IPUS should accept the cluster's short contours as support without raising a discrepancy. In this scenario, however, the scenario-specification will not be useful because source A's steady region should not extend into block 3.

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

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

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

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

4.2.3 BLOCK 3

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

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

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

Define Expectations: An explicit expectation for B's microstreams in block 4 is posted.

4.2.4 BLOCK 4

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

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

Reprocessing: To find "enough" (60% in this case) of C's attack region, IPUS must search back into block 3 and reinterpret the previously-detected but unrecognized short contours as valid attack-region contours. C's attack region and its chirp characteristics are identified in the previous block's signal data[4].

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

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

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

5 Primary IPUS Components

Developing the testbed to handle examples like the one just described above has given us several insights into the interrelationships among components that we include in IPUS to handle practical applications. In this section we discuss the discrepancy detection, discrepancy diagnosis, differential diagnosis, signal data reprocessing, and system control components of IPUS.

5.1 Discrepancy Detection

An important consideration in discrepancy detection is that the expectation hypotheses are often qualitative: they are not describable with specific numerical values. An example of this is an expectation like "within the next two seconds, a sinusoidal component currently at 1200 Hz will shift to a frequency between 1200 and 2000 Hz and the shift's duration will be between 100 and 500 msec". This implies that discrepancy detection mechanisms must be able to work with ranges of permissible values as well as specific values. This requires a representation in which qualitative calculus can be performed. In [14] we discuss the range calculus used in the testbed implementation described in this report.

The task of detecting discrepancies is distributed among all the knowledge sources responsible for interpreting lower-level data as higher-level concepts (e.g., interpreting 5 contours as 1 microstream). Each such KS, when acting in a goal-directed manner, checks to see if any data can possibly support the sought-after higher-level concept. If none can be found, or if only partially supportive data is available, the KS will record this as a source of uncertainty (SOU) 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, an SPA discrepancy detection KS is executed to check if SPA outputs are consistent with each other. An example of this checking occurs in the trace when IPUS detected a significant energy shift in the time-domain, but no new contours in the frequency domain. Again, when discrepancies are found, SOUs are posted in the problem solving model. The basic types of symbolic SOUs 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 IPUS architecture, an important issue is the relationship between the symbolic SOUs associated with various hypotheses and the discrepancy descriptions generated by the discrepancy detection process. Our testbed implementation defines 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 labelled with a *negative evidence* SOU (in particular, a SUPPORT LIMITATION SOU, which indicates that support for a hypothesis is limited until results of further processing are obtained) connects the less reliable hypothesis to the composite hypothesis.
3. **Violation-type Discrepancies and SOU's**. A violation-type discrepancy occurs when signal processing output data violates the *a-priori* known characteristics of the entire class of possible input signals in the application domain. When such an output data hypothesis is posted on the interpretation blackboard, a *negative evidence* type of SOU is attached to it. This SOU contains a description of the violated condition.

In addition to the discrepancy detection components of the interpretation KS's (that perform conflict discrepancy detection), the testbed contains one KS each for fault discrepancy detection and violation discrepancy detection. We first describe the fault detection algorithm, which in this case compares time-domain processing results with frequency-domain processing results for consistency. Starting at time 0 the fault detection KS monitors the average-energy fluctuations between consecutive 128-point segments of the raw signal data and computes for each segment a weighted variation threshold T_i from the differences:

$$T_i = \omega T_{i-1} + | E_i - E_{i-1} | \tag{1}$$

where E_i is the average energy of the i^{th} 128-point segment. If $|E_i - E_{i-1}| > T_{i-1}$, the segment is marked as having experienced either a positive or negative energy shift, depending on the sign of the difference. When a negative shift follows a positive shift by less than twice the length of the data window (STFT-INTERVAL) used by the frequency-domain processing algorithms, the KS assumes that if the shifts represent an impulsive source, the data window length would preclude the source from creating a significant (e.g., more than 4 frequency peaks over the time period) event in the spectral record. At this point a fault discrepancy is recorded. If at some later point IPUS chooses to resolve this discrepancy, it can use the discrepancy diagnosis KS to explain the energy shifts as an undetected source or as some glitch due to noise or as the overlapping onset and fadeout of two previously recognized sources.

The violation detection KS checks if a large number of short contours contiguous in time could indicate the distorted presence of a source. When this occurs, the assumption that only significant-length contours may support sources is violated. The KS is instantiated by the control when a large number of short contours relative to the total number of contours is detected within the current block. A frequency-bin histogram algorithm is used to identify regions of high contour-density in the spectrum. If these regions contain many short contours, the violation detection KS assumes that short contour clusters in these regions could represent a source's distorted presence in the contouring KS's output.

5.2 Discrepancy Diagnosis

The discrepancy diagnosis KS, which is based on [12], models the reasoning of a signal processing expert and carries out a discrepancies-to-distortions inverse mapping. A major part of the expert reasoning makes use of knowledge regarding the underlying Fourier *theory* for the signal processing algorithms. This diagnostic reasoning is captured within a means-ends analysis framework [16] using multiple levels of abstraction and a verification phase. Furthermore, the reasoning is carried out with a qualitative description of the various signal quantities involved in order to deal with uncertain and approximate information. Figure 10 outlines the plan-and-verify strategy of the diagnostic process.

The formal discrepancy diagnosis task is to generate a sequence of "distortion operators" that can explain the discrepancies between an *initial signal state* that represents the believed information (properties of the input signal class, expectations, or outputs from alternative SPA's whose outputs are less precise but more

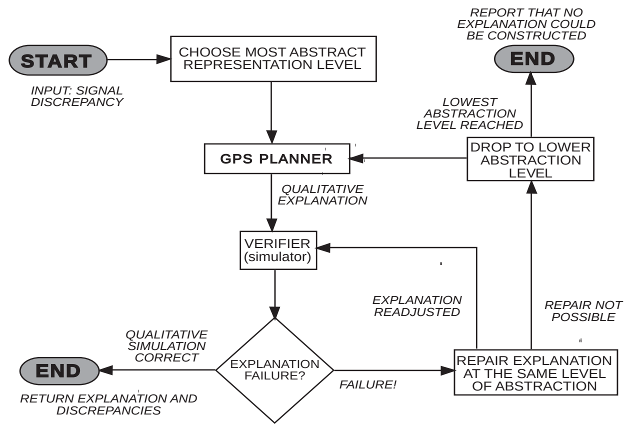

Figure 10: *The plan-and-verify strategy of the IPUS discrepancy diagnosis knowledge source.*

reliable) and a *goal signal state* that represents the SPA output data. The diagnosis knowledge-base contains operators that model various kinds of distortions that can result from improperly-tuned signal-processing control parameters. For example, in the context of the STFT algorithm, one of the distortion operators models a "frequency-resolution" distortion that occurs in the SPA output data when the window-length control parameter is not large enough to resolve two closely-spaced frequency components in a signal (see figure 11). The diagnosis process hypothesizes a sequence of operators, which when applied to the initial signal state will yield the distorted goal state. The search for the sequence is carried out using progressively more complex abstractions of the initial and final states, until finally an abstraction level is reached where an operator sequence can be generated using no more signal information than is available at that level. That is, the diagnosis process mimics the diagnostic reasoning of experts in that they first offer explanations (i.e., operator sequences) that are as uncomplicated as possible[18].

Once a candidate sequence has been obtained, the diagnostic process enters into its *verify phase*. At this point, the diagnostic process "drops" to to the lowest abstraction level at which a description of the initial state is known. Verification proceeds as a degenerate case of the GPS algorithm at this lowest abstraction level. That is, no real "operator search" is carried out: the "search" algorithm simply selects operators in the order they appear in the candidate operator sequence. This phase verifies that the pre- and post-conditions of each operator are met even when all information about the initial and final states is considered. If verification succeeds, the diagnosis process returns the candidate operator sequence as its final answer. If verification fails at some point, however, the diagnosis process attempts to "patch" the operator sequence by building a new sequence that eliminates the unmet conditions observed in the original sequence. This new sequence is then

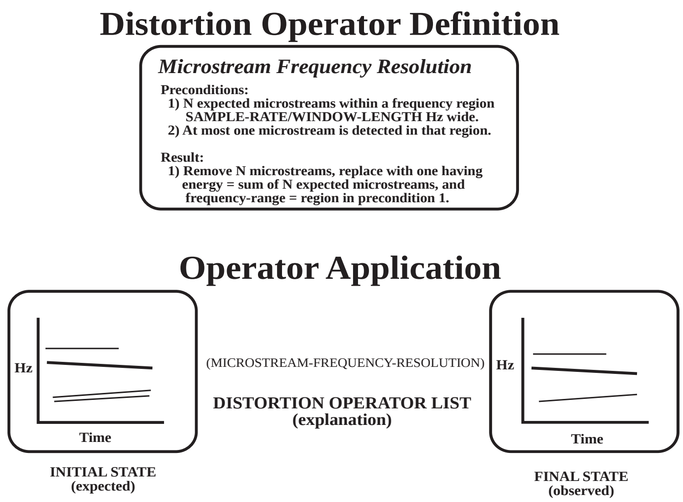

Figure 11: *The microstream-frequency-resolution operator. When applied to a state, the operator replaces each set of expected microstreams whose members are closer than SAMPLE-RATE/WINDOW-LENGTH with a single microstream, reflecting the resolving limits associated with the current value of WINDOW-LENGTH. In the short example illustrated, this operator effectively reduces the differences between the expected state and the observed state.*

inserted into the original operator sequence and verification continues.

One issue not originally dealt with in [12] that arises in the IPUS framework is the problem of incorrect explanations. Sometimes the first explanation offered by the diagnosis process will not enable the reprocessing mechanism to eliminate a discrepancy. In these cases, IPUS may decide to reactivate the diagnostic process and provide the incorrect explanation as one that must not be returned again. To prevent the diagnosis process from repeating the same search it performed when it originally generated the incorrect explanation, the system stores with the explanation the search-tree context it was in when the explanation was produced. Then, the diagnosis process simply "starts up" from that point in the search space when it begins considering operators for a new explanation.

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

5.3 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 processing parameter settings under which signal reprocessing may remove those distortions. Figure 12 illustrates the reprocessing knowledge source's organization. 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 13 shows the definition for one reprocessing plan for the explanation (CONTOUR-TIME-RESOLUTION).

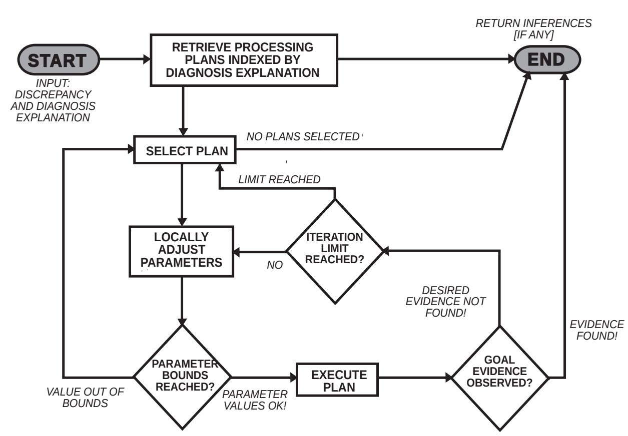

Figure 12: *The IPUS reprocessing knowledge source's framework*

From the retrieved set of applicable plans, one is selected during the plan-selection stage. Selections are governed by "cost" criteria including 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 at least partially

```
(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*
                          *STFT-INTERVAL*
                          *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 13: *The definition for a reprocessing plan 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 STFT-INTERVAL 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.*

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 parameter value outside of its prespecified range, the plan is considered to have failed.

When failure is indicated, the 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), the IPUS system has no further recourse but to annotate the entities involved in the discrepancy with SOUs indicating low confidence due to unresolvable discrepancies. These SOUs' effects on the entities' confidence levels are then propagated to interpretations based on those entities.

5.4 Differential Diagnosis

In the course of processing signal data, IPUS may encounter data that could support several alternative interpretations. This situation occurs when a query to the source database returns more than one source model whose frequency components (or energy levels, or whatever other indexing feature is used) overlap the observed data. An example of this type of event appears in the trace when data in block 1 in the $[1200, 1230]$ range could support both source A and source D's existence. In such cases IPUS (under the RESUN framework) pursues a least-commitment interpretation strategy. For each retrieved model, an interpretation hypothesis supported by the observed data is created, and for each hypothesis, a source of uncertainty (ALTERNATIVE-EXPLANATION-SOU, in this case) is recorded in the problem-solving model. This SOU is left unresolved until the focusing heuristics deem its resolution appropriate to the current problem-solving context.

The role of the differential diagnosis knowledge source is to produce reprocessing plans that will enable IPUS to prune the interpretation search space for ambiguous data. 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.

The KS first compares the interpretation hypotheses to determine their overlapping regions. Any observed evidence in these regions is labelled "ambiguous". The KS then determines the hypotheses' discriminating regions (e.g., `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 14 depicts the differential diagnosis KS's execution for the ambiguous data observed in block 1 of the trace (see page 13).

We should note that the explanatory reasoning performed in the differential diagnosis KS for missing evidence is very primitive compared to that available in the discrepancy diagnosis KS; there is no explicit modeling of distortions via formal operators, nor is there a rich set of explanations available. Only simple distortions like loss of low-energy microstreams due to energy thresholding are considered. The justification for this design is that the differential diagnosis KS's role is to trigger reprocessing that quickly prunes large areas of underconstrained interpretation spaces, *without preference* for any particular interpretation. On the basis of this specification, it is not appropriate to devote time consuming, sophisticated reasoning to the generation of missing-evidence explanations.

In cases where IPUS 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.

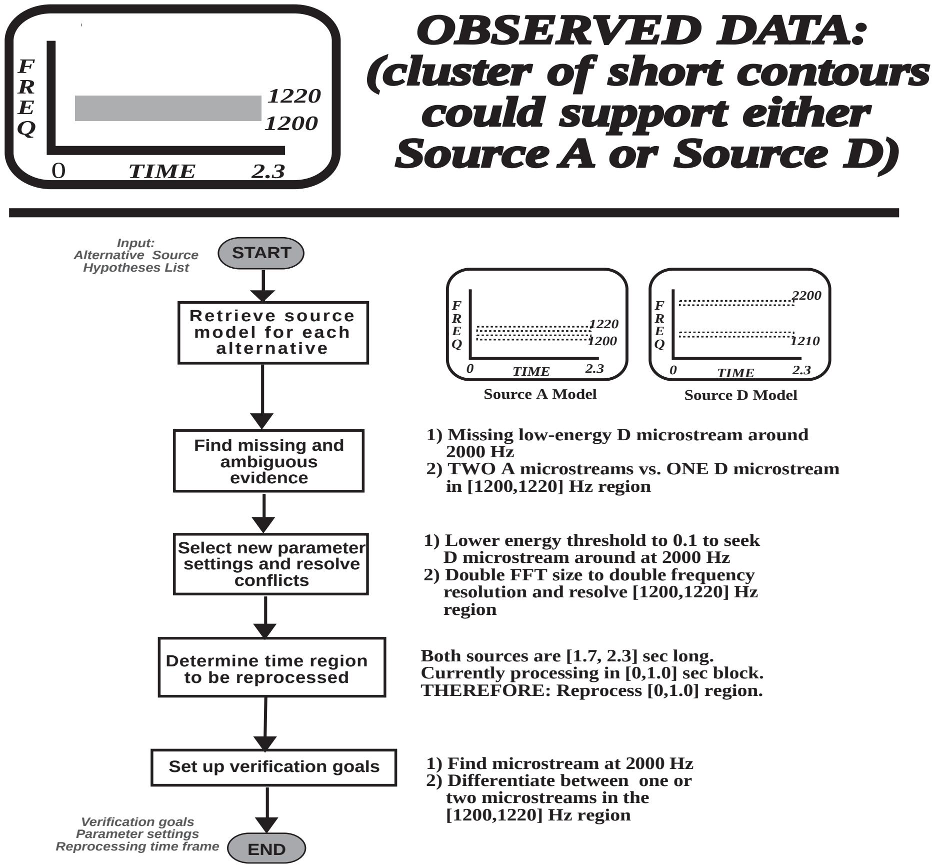

Figure 14: *A flowchart for the IPUS differential diagnosis KS and its execution in the testbed scenario.*

5.5 Source Models and Interpretation Knowledge Sources

The IPUS architecture assumes that sources have discernible structure. That is, sources exhibit features like microstreams and energy variations that can be identified using the output of *at least one* SPA. The system source library uses a grammar to specify this structure at the three highest abstraction levels. This library is indexable by frequency regions or source name. Properties of source features (e.g., frequencies of microstreams) are specified in the same range calculus used in expressing expectations and discrepancies. Figure 15 illustrates the grammar units used to define source A.

The CONTOUR-MICROSTREAM, MICROSTREAM-STREAM, and STREAM-SOURCE interpretation KSs make extensive use of the source models. They can interpret lower-level data as higher-level concepts in a data-driven mode, or search for lower-level data to support higher-level concepts in a goal-driven mode. For example, in the data-driven mode the CONTOUR-MICROSTREAM creates microstream hypotheses linking contours that are similar in frequency and together exhibit energy variations similar to the attack, steady, or decay behavior of a microstream. In the goal-directed mode, however, the KS searches for contours that overlap the time, frequency, and energy region specified by a microstream expectation.

As data is driven upward through interpretation levels, many underconstrained hypotheses are generated. For instance, when a sequence of decaying-energy contours is grouped to hypothesize first a microstream's presence, then a stream's presence, and finally a source's presence, each interpretation's constraints are very broadly specified (e.g., the attack and steady regions of the alleged microstream could lie anywhere in time before the decay contours, any number of duplicate streams could have occurred before the current stream, the source's volume is poorly-defined on the basis of a decaying energy rate alone, etc). Once the source database is queried, however, the returned models' information (number and location of microstreams, relative microstream energies, expected attack and steady time regions, etc) is propagated back through this underconstrained set of related hypotheses in a goal-directed manner. When more than one model can apply, multiple alternative copies of the hypothesis-set, each one reflecting the constraints of a model, are created.

The hypotheses in each set can now be used by the interpretation KSs as expectations when the focusing heuristics choose to search for more evidence to support or disprove the sources specified by the sets. As more evidence is found (or not found), the interpretation KSs record these discoveries by propagating the evidence's constraints (e.g., the maximum time span found for a source's microstream provides a minimum bound on the duration of the source's stream and the source itself, while the lack of a high-energy microstream indicates a greater belief that its source isn't actually present in the monitored environment, etc.) through the hypothesis-set.

5.6 Control Issues

Control in IPUS is based on the RESUN control framework[2, 3], which models interpretation as a process that gathers evidence to resolve particular sources of uncertainty in interpretation hypotheses. This section first provides an overview of the RESUN framework components and then describes how they are used within IPUS in the context of the testbed trace.

The key components of the approach are an *evidential representation* for expressing sources of uncertainty (SOU's) in the evidence hypotheses and a script-based, *incremental reactive control planner*. The control planner is based on *control plan schemas* and *focusing heuristics*. Control plans consist of either primitive actions (knowledge sources), or sets of subgoals to be satisfied, and are used to define available interpretation methods. Focusing heuristics represent strategy knowledge to be applied during the planning process to select the best control plans and control plan

```
CLOS instance of GRAMMAR-UNIT
Local Slots:
LEVEL:                          :SOURCE
NAME:                           A
TYPICAL-UNIT:                  <SOURCE  A   TONAL
                                         duration:[17000 23000]
                                         gram-components:  (<GRAM-UNIT: STREAM  AST5834>)
SPA-PARAMETERS: ((*FFT-SIZE* . 2048) (*STFT-INTERVAL* . 1024)
                 (*RELATIVE-ENERGY-THRESHOLD* .20)
                 (*STFT-OVERLAP* . (*  3/8  *STFT-INTEVAL*))
SUB-UNITS:      (<GRAM-UNIT:STREAM AST5835>)
DURATION:        [17000 23000]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
CLOS instance of GRAMMAR-UNIT
Local Slots:
LEVEL:                          :STREAM
NAME:                           AST5835
TYPICAL-UNIT:                  <STREAM:  cat:TONAL
                                          comp-microstreams: (< freq:[1218 1222]
                                                               energy:[0.40 0.46]>
                                                              <freq:[1198 1202]
                                                               energy:[0.40 0.46]>)
SUB-UNITS:                    (<GRAM-UNIT: MICRO Ams5834>
                               <GRAM-UNIT:MICRO Ams5833>)
RELATIVE-TEMPORAL-POSN:  0.0
DURATION:       [17000 23000]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
CLOS instance of GRAMMAR-UNIT
Local Slots:
LEVEL:              :MICROSTREAM
NAME:               Ams5833
TYPICAL-UNIT: <microstream: freq: [1218 1222]
                            energy: [0.40 0.46]
                            attack-start-time: [0 0]
                            steady-start-time:[4000 6000]
                            decay-start-time: [13000 17000]
                            decay-end-time: [17000 23000]
                            attack-freq-behavior: (CONSTANT . [1218 1222])
RELATIVE-TEMPORAL-POSN:  0.0
DURATION:      [17000 23000]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
CLOS instance of GRAMMAR-UNIT
Local Slots:
LEVEL:                          :MICROSTREAM
NAME:                           Ams5834
TYPICAL-UNIT: <microstream: freq: [1198 1202] energy: [0.40 0.46]>
RELATIVE-TEMPORAL-POSN:  0.0
DURATION:       [17000 23000]
```

Figure 15: *Grammar definition units for Source A. The source grammar unit (top) specifies the range of durations and the component streams for A. It also contains a list of ideal parameter settings for monitoring the source in isolation. The stream grammar unit specifies its offset relative to the start of the source's signal emission, and the component microstreams' frequency and energy regions. The microstream grammar units (bottom two cells) specify the attack, steady, and decay regions for each microstream, as well as their temporal offsets within their stream.*

```
#<PS-MODEL-UNIT 26224753>
Named structure of type PS-MODEL-UNIT

ANSWERS:(<ANSWER-PS-MODEL Answer:#<source-hyp.00003 D>>
            <ANSWER-PS-MODEL Answer:#<source-hyp.00002 A>>
            <ANSWER-PS-MODEL Answer:#<source-hyp.00001 B>>)
SOUS:(<UNCERTAIN-NONANSWER-SOU (#<contour-hyp.00019>)>
        <UNCERTAIN-ANSWER-SOU #<source-hyp.00003 D>>
        <UNCERTAIN-ANSWER-SOU #<source-hyp.00002 A>>
        <UNCERTAIN-ANSWER-SOU #<source-hyp.00001 B>>
        <UNCERTAIN-NONANSWER-SOU (#<contour-hyp.00010>)>
        <UNCERTAIN-NONANSWER-SOU time:[1,10000]>
        <NO-EVIDENCE-SOU [10000,  ])
```

Figure 16: The first slot contains the answers, in this case three source hypotheses. The second slot contains all the SOUs. The NONANSWER-SOUs represent short contours, or areas in the time-frequency space where no contours were found, or hypotheses at some other level that for some reason have become disbelieved. The UNCERTAIN-HYPOTHESIS-SOUs represent hypotheses not at the answer level (unexplained contours, unexplained microstreams, or unexplained streams). And the UNCERTAIN-ANSWER-SOUs represent the answers (source hypotheses) found so far. Each of these SOUs has a summary unit, which indicates where the uncertainty stems from.

instances to be pursued. These heuristics may require partial expansion of control plans to be able to make a decision. To do this, the RESUN framework provides a *refocusing mechanism* that allows decisions to be reconsidered.

The other major component of the architecture is the *problem-solving model* (PSM). The PSM is the basis for generating the high-level goals that drive the planning process. The PSM summarizes the system's current interpretation of its data and the uncertainties associated with the interpretation. This model is updated as hypotheses on the interpretation blackboard are added or modified. The PSM is a structure with two fields: answers and SOUs. The *answers* field contains a list of hypotheses currently considered as answers. That is, the hypotheses whose current levels of uncertainty indicate that they could potentially be answers. The *SOUs* field contains an abstract description of the set of uncertainties in the system interpretation covering the entire data stream examined thus far. Figure 16 shows how the problem solving model appeared after all unexplained data have been explained as part of sources but before any diagnosis had been performed in the first block in the experiment trace. This is the situation that confronts the control when it decides to obtain evidence through differential diagnosis to resolve uncertainty associated with the alternative answers *source-hyp.00003* (source D) and *source-hyp.00002* (source A).

The following is a description of the SOU types in the PSM and the control plans available to IPUS for solving them.

- *NO-EVIDENCE-SOU:* denotes the fact that no spectral information has been gathered for the period of time specified. There is always a NO-EVIDENCE-SOU in the PSM, representing the lack of evidence for the next block. Figure 17 shows the contents of this SOU. The territory slot represents the region, one block at a time, for which there is no evidence. When this SOU is solved, a new NO-EVIDENCE-SOU for the next block is posted.

```
<NO-EVIDENCE-SOU>
Named structure of type NO-EVIDENCE-SOU

PS-MODEL: #<NO-EVIDENCE-PS-MODEL
                  Territory: #<COMPLEX-REGION-UNIT
                              (#<REGION-UNIT  T:[10001, 20000]
                                              F:[0, 10000]
                                              E:[0, 1.0]>)>>
```

Figure 17: *NO-EVIDENCE-SOU from the PSM in figure 16*

To solve this SOU, new data should be gathered and processed up to the contour level. This is done by the *SOLVE-NO-EVIDENCE-SOU* plan, which is formed from the following sequence of subgoals:

- Find appropriate parameters to process the data[5]
- Get waveform data
- Have spectral information gathered [6]
- Have contours from spectral information
- Have discrepancies between TD and STFT solved [7]

- *UNCERTAIN-ANSWER-SOU:* represents an uncertain hypothesis at the answer level. The uncertainty is summarized and explained in the summary unit of the SOU. Figure 18 shows an instance of this SOU.

  There are two control plans whose goal is to reduce the uncertainty in answer hypotheses:

  DIFFERENTIAL-DIAGNOSIS-FOR-SOURCES control plan is used when the answer hypothesis is uncertain because there are alternative explanations (in terms of other answer hypotheses) for its supporting data. This method will call the *differential-diagnosis-KS* that will propose a plan to find the differentiating features of the alternative sources.

  SOLVE-UNCERTAIN-ANSWER-SOU control plan is used when the answer hypothesis is uncertain for other reasons (figure 19). This control plan is formed of the sequence of subgoals:

  - *Have-hypothesis-SOU* goal is satisfied by a primitive control plan that determines the set of uncertainties in the hypothesis. A *focusing heuristic* will decide which SOU(s) of that set is to be solved.

[5] the Global parameter adjustment is not currently implemented

[6] Currently the only algorithms available to IPUS are the Short Time Fourier Transform (STFT) and time-domain (TD) energy measurement algorithms.

[7] The action of satisfying this goal includes checking for fault discrepancies between TD and STFT results. If a discrepancy is detected it is solved immediately.

```
<UNCERTAIN-ANSWER-SOU>
Named structure of type UNCERTAIN-ANSWER-SOU

SUMMARY:  <SUMMARY-UNIT
                RATING: 0.12307032
                PARTIAL-EVIDENCE-UNCERTAINTY: 0.29800
                POSSIBLE-ALT-EXPLANATION-UNCERTAINTY: 0.74052006
                POSSIBLE-ALT-SUPPORT-UNCERTAINTY: 0.0
                CONSTRAINT-UNCERTAINTY: 0.0
                 ALT-EXTENSIONS-UNCERTAINTY:  0.0045408897
                 ALTERNATIVE-EXTENSIONS: (<ALT-EXT-UNIT Alt:#<source-ext.00004 a>>)
                NEGATIVE-EVIDENCE-UNCERTAINTY: 0.18336463
                NEGATIVE-EVIDENCE-EXPLANATION: 0.09136386
PS-MODEL: <ANSWER-PS-MODEL
                Answer hyp: #<source-hyp.00003 D>
                Territory: #<COMPLEX-REGION-UNIT
                             (#<REGION-UNIT T:[1024, 9216]
                                              F:[1191, 1230]
                                              E:[0.22, 0.30]> ) > >
```

Figure 18: *UNCERTAIN-ANSWER-SOU from the PSM in figure 16*

– *Have-hypothesis-SOU-solved* goal is satisfied by several control plans. The choice of control plan depends on the chosen SOU.

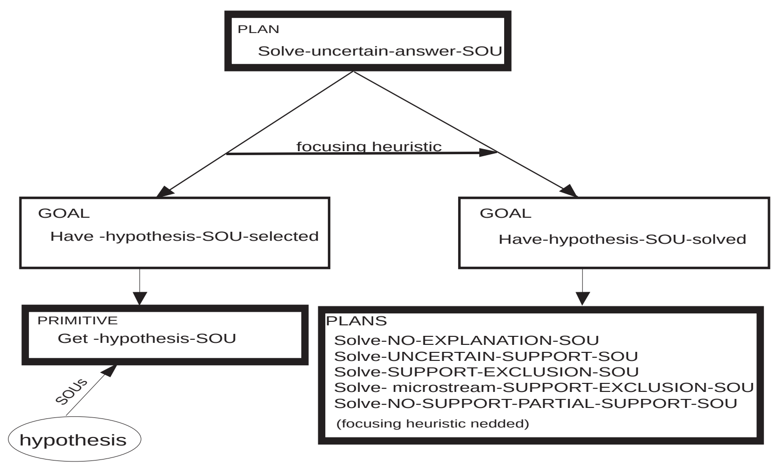

Figure 19: *Solve uncertain answer SOU plan*

- *UNCERTAIN-HYPOTHESIS-SOU:* has the same structure as the previous one, and can represent unexplained contour, microstream, and stream hypotheses. The control plan *SOLVE-UNCERTAIN-HYPOTHESIS-SOU* is instantiated to solve this SOU. This control plan is the same as the SOLVE-UNCERTAIN-ANSWER-SOU plan mentioned before.

- *UNCERTAIN-NONANSWER-SOU:* represents one or several hypotheses that are considered nonanswers[8], or regions of the problem space considered not to have any information of interest. The nonanswer's uncertainty is explained in the summary unit. Figure 20 shows an instance of this SOU from the PSM in figure 16.

[8]A hypothesis is considered nonanswer when it isn't of interest to the system or when it has been disbelieved (based on evidence)

```
<UNCERTAIN-NONANSWER-SOU>
Named structure of type UNCERTAIN-NONANSWER-SOU

SUMMARY: <SUMMARY-UNIT RATING:0.857114>
PS-MODEL:<NONANSWER-PS-MODEL
                Type:         :HYPOTHESES
                Time slice: [1, 10000]
                 Nonanswer-hyps: (#<contour-hyp.00009>)
                 Territory:  #<COMPLEX-REGION-UNIT
                               (#<REGION-UNIT T:[5120, 6114]
                                                   F:[1191, 1211]
                                                   E:[0, 0.1]>
                                #<REGION-UNIT T:[6144, 7168]
                                                   ...  >
                                 ...) > >
```

Figure 20: *UNCERTAIN-NONANSWER-SOU from the PSM in figure 16*

> *CONTOUR-VIOLATION-DISCREPANCY-DETECTION* is the only implemented control plan for solving this SOU. This method is called when the SOU represents a large number of unexplained short contours in a contiguous time region. The plan will check if these short contours, previously classified as noise, could be evidence for a source. This is done by contour clustering into frequency bins. If evidence for a source is found, a violation discrepancy is raised, indicating that finding short contours as evidence for a source violates the assumption of long contours supporting sources.

The contour level is the lowest abstraction level represented in the PSM, because waveform data is processed uninterruptedly up to the contour level by the signal processing algorithms. Although no interpretation is made at this point, the selection of parameters and algorithms can be affected by the interpretation process.

The PSM is updated every time a hypothesis is posted or modified at the contour, microstream, stream or source levels of the blackboard. The reason for updating the PSM so often is that refocusing may potentially occur at any step of the control plans, and if this happens, it is important for the PSM to have the most recent summary of the problem solving situation, in order to decide which goals/actions to pursue next.

Figure 21 shows the initial control plan *SOLVE-PROBLEM*. This control plan satisfies the generic goal *Have-Problem-Solved*, which has one input parameter *ps-goal* that specifies to the system *1) priority sources* indicating those sources the system should detect quickly if they appear, *2) initial contents of the PSM* allowing the system to start with some explicit expectations, *3) minimum answer belief*, *4) minimum nonanswer belief* and *5) answer level*, indicating the level of abstraction of the answer (typically the source level).

The *SOLVE-PROBLEM* control plan has the two subgoals:

- *Have-initialized-PSM:* This is satisfied by a primitive control plan that initializes the PSM with the contents specified in the input parameter or a NO-EVIDENCE-SOU for the first block, when no initial contents are specified.

- *Have-Resolved-PSM-Uncertainty:* The RESOLVE-PSM-UNCERTAINTY control plan satisfies this goal.

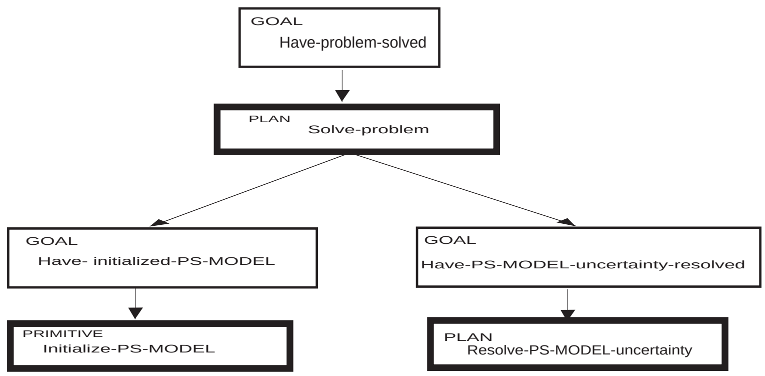

Figure 21: *Initial control plan*

The control plan *RESOLVE-PSM-UNCERTAINTY* (Figure 22) reduces the uncertainty in the problem solving model. To do so, this control plan will iterate on the two following subgoals until the termination criteria is reached:

- *Have-PSM-SOU* goal is satisfied by a primitive control plan that determines the set of uncertainties in the PSM. A *focusing heuristic* will decide which SOU(s) of that set are to be solved in this iteration.

- *Have-PSM-SOU-solved* goal is satisfied by several control plans[9] that will attempt to solve the selected SOU. In doing so, data will be changed on the blackboard and as a consequence, the PSM will get updated.

The first decision faced by the system is which SOU from the PSM to reduce. This decision depends on the status of the system (e.g., what are the priorities of the system, which sources have most evidence, etc). Refocusing is necessary here, because sometimes the choice may prove to be incorrect after further processing at lower abstraction levels on the blackboard concludes. This can occur when IPUS decides to solve the partial evidence uncertainty in a source hypothesis and finds that the microstream supporting the source has only partial support. Examination of the source hypothesis enables the system to determine that the interpretation's uncertainty is due to having partial evidence, but information about the abstraction level or time region of the uncertainty is unavailable. As part of the process for solving this uncertainty, the system examines the microstream level, where the PARTIAL SUPPORT SOU is specified. If no data has been gathered for the time the hypothesis needs support, nothing can be done until the NO EVIDENCE SOU

[9]The choice of plan will depend on which SOU is selected by a focusing heuristic for resolution.

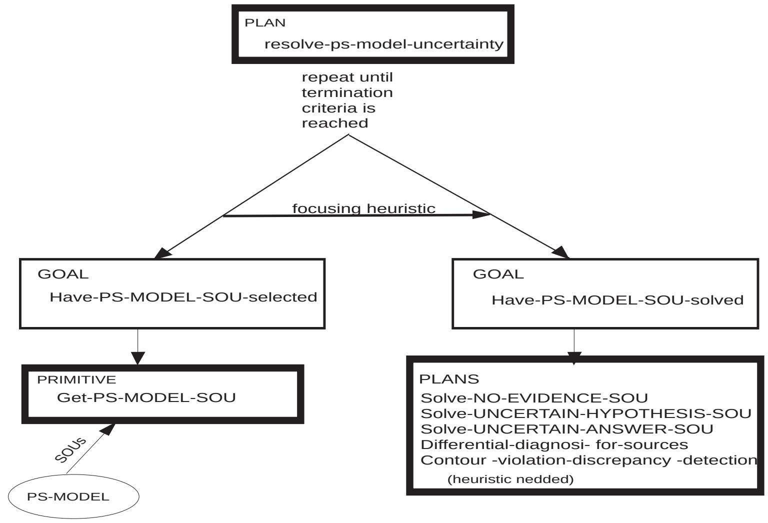

Figure 22: *Top level iteration*

is solved for that region. This causes the refocusing condition to be true and the control focusing decision to be reevaluated.

In the Resolve-PSM-uncertainty focusing heuristic, the system has to choose one or several SOUs from the PSM. The strategy embedded in the control plans and focusing heuristics will try to do reasonable work[10] in each block, and once finished with one block, will go ahead to process the next one.

An implicit strategy in this focusing heuristic is that expectations should be used as much as possible. In this way, after trying to support all the expectations, the unexplained data left will be minimized and easier to explain. This focusing heuristic defines the following order for selecting SOUs to be resolved:

1. UNCERTAIN-ANSWER-SOUs for priority sources.

2. UNCERTAIN-HYPOTHESIS-SOUs for unexplained contours that could belong to a priority source (only in the frequency domain).

3. UNCERTAIN-ANSWER-SOUs for source hypotheses. If there are multiple such hypotheses, choose the one with maximum belief and uncertainty in the current block, giving priority to the ones with uncertain negative evidence.

4. UNCERTAIN-HYPOTHESIS-SOUs for unexplained contours.

5. UNCERTAIN-HYPOTHESIS-SOUs for unexplained contour clusters.

[10] "Reasonable work" means finding support for expectations, solving the negative evidence uncertainty (due to possibly incorrect parameter settings) and explaining all the data up to the highest possible abstraction level (this includes checking for the presence of new sources in the scenario)

6. UNCERTAIN-NONANSWER-SOUs with too much uncertainty. This may represent a situation that may involve a discrepancy due to incorrect parameter setting.

7. NO-EVIDENCE-SOU (gather new data).

This heuristic will result in reducing the uncertainty of the existing priority sources first. In the case of a new block of data that has been processed up to the contour level, contour data will be matched against expected priority source definitions. If there is unexplained data that could possibly support a priority source, this data will be processed further. The next phase will try to reduce the uncertainty in existing sources, giving priority to the ones with solvable negative evidence. It wouldn't make sense to work on a source that could become disbelieved because of negative evidence.

After all possible work has been done with the existing sources, unexplained hypotheses are chosen to be explained. If there is too much data classified as noise (i.e., UNCERTAIN-NONANSWER-SOU), the system tries to identify frequency clusters that may indicate the presence of a source due to incorrect parameter settings. Finally, if there is nothing left to be done in this block, the NO-EVIDENCE-SOU is chosen to start processing the next block.

The focusing heuristic used in the *SOLVE UNCERTAIN-HYPOTHESIS-SOU* and *SOLVE UNCERTAIN-ANSWER-SOU* control plans decides which SOU in the hypothesis to solve. The strategy implemented in this heuristic tries to choose the most relevant SOUs[11] first, and after, if work is still needed on the hypothesis, other SOUs will be chosen giving priority to the negative evidence SOUs. The first SOUs to be resolved will be the NO-EXPLANATION-SOU, NO-SUPPORT-SOU or PARTIAL-SUPPORT-SOU for regions where the data has been already gathered; this seems to be a reasonable approach, since if a hypothesis is uncertain due to the lack of processing, further processing is easy to do at this point and may indicate the future direction of processing. If the uncertainty doesn't come from any of these SOUs, the negative evidence SOUs (SUPPORT-EXCLUSION-SOU, SUPPORT-LIMITATION-SOU or ALTERNATIVE-EXTENSION-SOU) are the next ones to be checked. Each negative evidence SOU has a summary unit that explains where the uncertainty comes from, and how important and certain it is. This summary unit is used by the focusing heuristic to decide which negative evidence to solve. Because solving negative evidence SOUs usually requires reprocessing, only very uncertain negative evidence will be solved.

When none of the previous SOUs are present (or weren't suitable to be solved) in the hypothesis, the heuristic will check the UNCERTAIN-SUPPORT-SOUs. Each UNCERTAIN-SUPPORT-SOU has a summary unit that explains where the uncertainty comes from (why the support is uncertain). Deciding which uncertain support

[11]The most relevant SOUs are those that most affect the evaluation of the sources of uncertainty and therefore the belief in the hypothesis.

to solve depends on the importance of each support for the hypothesis, and on the type of uncertainty in the support.

To conclude, we briefly summarize the set of methods used to solve the SOUs associated with individual hypotheses.

- *SOLVE-NO-EXPLANATION-SOU:* solves the NO-EXPLANATION-SOU in a hypothesis. This plan will explain the hypothesis up to the highest possible abstraction level. Knowledge sources[12] are called to drive the data up and focusing heuristics are called when more than one explanation is possible for a hypothesis[13].

- *SOLVE-NO-SUPPORT-PARTIAL-SUPPORT-SOU:* this method solves a NO-SUPPORT-SOU or PARTIAL-SUPPORT-SOU in a hypothesis. Expectations are created down to the microstream level. This process of expectation generation stops at the microstream level because microstreams are supported by contours generated by the signal processing algorithms. SUPPORT-EXCLUSION-SOUs will be posted for those regions of the microstream not supported by existing contours. The main focusing decision in this method is deciding which support to look for or generate. The focusing heuristic making this decision uses domain specific knowledge (e.g., the steady of a microstream is more important than the attack or decay) and general strategies (e.g., look for support around already supported regions).

- *SOLVE-SUPPORT-EXCLUSION-SOU:* solves the SUPPORT-EXCLUSION-SOU in a hypothesis. This SOU represents a conflict discrepancy and is usually is usually present at the microstream level (at higher levels, supporting expectations are created). The only implemented plan for solving this SOU consists of discrepancy diagnosis followed by reprocessing (if the diagnosis is able to generate an explanation.)

RESUN's evidential representation system also includes a framework for numerically summarizing the SOUs. The symbolic SOUs allow the system to understand the reasons why hypotheses are uncertain so that the system can identify appropriate methods to resolve its uncertainty. However, the system still needs numeric evaluations of the degree of belief in the hypotheses in order to evaluate the termination criteria and reason about control decisions. For example, in deciding which hypotheses to work on next, the system must consider whether each hypothesis satisfies the termination criteria, how close it is to satisfying the criteria, and what its rating is relative to other hypotheses. In addition to computing hypothesis belief

[12] The knowledge sources used to drive the data up are from the sequence: *contour-to-microstream-ks*, *microstream-to-stream-ks* and *stream-to-source-ks*

[13] When more than one possible explanation for a hypothesis is possible, the system has three choices: **1)** create alternative explanations or **2)** delay the decision until more information is available or **3)** create the most likely explanation and leave an SOU about remaining alternatives.

ratings, the summarization process computes ratings for each SOU that identify the effect that the SOU has on hypothesis uncertainty. These SOU ratings are used in making certain control decisions; the focusing heuristics do not have to incorporate this reasoning about SOU weight.

Instead of just computing a single-number belief rating, the summarization process produces a composite characterization of the uncertainty in a hypothesis in terms of an overall belief rating and the amount of uncertainty contributed by the different classes of SOUs (associated with the hypothesis). The major elements of the composite rating are: possible alternative explanation uncertainty, possible alternative support uncertainty, constraint uncertainty, negative evidence uncertainty, and alternative extension uncertainty. Thus, for any belief rating b, where $b < 1$, the sum of the ratings for these uncertainty classes would add up to $1 - b$ (actually the situation for the negative evidence is a bit more complicated—as we will see below). In addition to these categories, the composite includes a partial support uncertainty rating that indicates how much of the remaining uncertainty might potentially be reduced by gathering additional evidence for the hypothesis.

These SOU-class ratings summarize the SOUs by giving an abstract indication of the reasons why the hypothesis is uncertain (i.e., not fully believed). Having the composite rating allows for more detailed reasoning than would be possible with a single number rating. For example, it can distinguish between a hypothesis that has low belief due to a lack of evidence and one for which there is negative evidence. This capability is used in distinguishing between potential answer hypotheses that should be modeled as "answers" and those that should be modeled as "nonanswers." Potential answer hypotheses may not be currently believed (i.e., belief rating < 0.5—more likely wrong than right) simply because not enough evidence has been gathered to resolve the inherent abductive uncertainty (resulting from alternative possible explanations for the supporting data). These hypotheses are represented in the problem solving model (PSM) as potential answers so that the system will attempt to prove them correct. On the other hand, potential answer hypotheses may not be believed because more evidence has been gathered against them than for them. Such hypotheses are modeled as nonanswers so that the system will attempt to disprove them. In either case, of course, the system may pursue the hypotheses until sufficient evidence is gathered to reach belief levels specified in the system goals.

The summarization process represents a hypothesis' uncertainty in a *summary unit* structure with the fields:

- *Rating* represents the hypothesis overall belief.
- *Possible-alt-explanation-uncertainty* indicates the uncertainty in the hypothesis due to the POSSIBLE-ALT-EXPLANATION-SOUs.
- *Possible-alt-support-uncertainty* indicates the uncertainty in the hypothesis due to the POSSIBLE-ALT-SUPPORT-SOUs.

- *Constraint-uncertainty* indicates the uncertainty in the hypothesis due to the constraint SOUs.

- *Alt-extensions-uncertainty* indicates the uncertainty in the hypothesis due to the alternative extensions (i.e., hyp representing alt explanations for the data).

- *Top-level-alt-exts* contains a list with the alternative hypotheses to the hypothesis this summary unit represents (used in computing alt-extensions-uncertainty.

- *Negative-evidence-uncertainty* indicates the uncertainty in the hypothesis due to negative evidence.

- *Negative-evidence-explanation* indicates the likelihood of there being explanations for the negative evidence. The effect of the negative evidence is reduced by this amount.

- *Partial-evidence-uncertainty* indicates by which percentage the belief in the hypothesis would increase if the PARTIAL-SUPPORT-SOUs were to be perfectly solved.

These ratings summarize to the hypothesis' evidence at all levels of abstraction (e.g., if a source is supported by a microstream which has partial support, this would be reflected in the source hypothesis' summary unit). This composite rating allows the focusing heuristics to make decisions based on the overall uncertainty of the hypothesis without accessing its support.

The summarization process is called every time a hypothesis is posted (or modified) at the contour, microstream, stream or source levels of the blackboard. The reason for this is that refocusing may potentially occur at any step of the control plans, and if this happens, it is important that the hypotheses are correctly rated in order for the focusing heuristics to make the best decision about which goals/actions to pursue next.

The summarization process operates by recursively summarizing the support evidence for a given top-level hypothesis. Summarization is carried out using application-specific evaluation functions because neither Bayes' Rule nor Dempster's Rule are generally applicable to interpretation due to the lack of independence of hypothesis evidence. Nonetheless, the application-specific evaluation functions effectively compute conditional probabilities and the composite rating permits these evaluation functions to be quite modular.

When a hypothesis is summarized, its SOUs are themselves summarized and combined into a single summary unit. This process has the following steps:

- *Uncertain Support Summarization:* This step creates a summary unit for each UNCERTAIN-SUPPORT-SOU in the hypothesis. These units rate the uncertainty stemming from the support evidence. The summarization of an

UNCERTAIN-SUPPORT-SOU is defined recursively in terms of the summarization of the support hypothesis it represents. For example, consider a source hypothesis supported by one stream, which is itself supported by two microstreams. To compute the rating for the source's UNCERTAIN-SUPPORT-SOU, the stream hypothesis must be summarized. But to compute the rating for the stream, its supporting microstreams must be summarized. This in turn requires that the microstreams' supporting contours be summarized, etc. This process terminates at the summarization of supporting spectrum hypotheses (spectrum hypotheses all have a rating of 1.0 – no uncertainty). Once the evaluation for the support hypotheses is finished, the rating for the UNCERTAIN-SUPPORT-SOU is computed in general by reducing the belief of its support hypothesis based on the ratings of the SOUs (constraint SOUs and POSSIBLE-ALT-EXPLANATION-SOUs) associated with the support inference.

- *Negative Evidence Summarization:* This step creates a summary unit for each negative evidence SOU in the hypothesis. Each unit has (1) an overall rating that represents the SOU's actual mitigating effect on the hypothesis' belief, (2) a negative-evidence-explanation rating that measures the uncertainty in the negative evidence represented by the SOU, and (3) a negative-evidence-uncertainty rating that measures the effect the negative evidence would have on hypothesis belief if the negative-evidence-explanation rating were 0.0 (i.e., no uncertainty).

- *Partial Support Summarization:* This step creates a summary unit for each PARTIAL-SUPPORT-SOU in the hypothesis. Each unit indicates by how much the hypothesis' rating would be increased if the SOU were solved.

- *Support Evidence Combination:* This step computes a summary unit for the hypothesis based on the previously summarized SOUs. The unit is initialized with the slot values from the minimum-rated UNCERTAIN-SUPPORT-SOU's summary unit[14]. This summary unit's overall rating is increased based on the

[14] We start with the minimum-rated UNCERTAIN-SUPPORT-SOU because for the particular hypothesis (extension) to be correct, all of its support inferences must be correct. Thus, the hypothesis (extension) cannot be more certain than any of its individual supports. However, the rating on the UNCERTAIN-SUPPORT-SOU only reflects the support evidence for this inference—it does not include the explanation evidence (the hypothesis being rated). For this reason, the minimum support is increased based on the existence of corroborating evidence—i.e., the other supports for the hypothesis. This is the basis of the hypothesize and test strategy for resolving hypothesis uncertainty. That is, the uncertainty in a piece of support evidence due to the possibility of alternative explanations for its supporting data (the basic uncertainty of abductive interpretation inferences) can be reduced by gathering corroborating evidence. Note though, that the extent to which this occurs depends on the amount and quality of the corroborating support—as well as the characteristics of the minimum-rated support. For example, if the minimum-rated support is low rated because of negative evidence rather than the possibility of alternative explanation for

amount of support evidence and the quality of this evidence as summarized in the other UNCERTAIN-SUPPORT-SOUs. The summary unit's overall rating is decreased based on the ratings of the negative evidence SOUs.

The remainder of this section briefly describes the rating evaluation functions used in the IPUS system. These functions are preliminary and are designed to work with the library of artificial sources described in this report. Generalization to real sources will be done when the source library is extended.

As stated earlier, POSSIBLE-ALT-EXPLANATION-SOUs are evaluated during the propagation of a hypothesis' summary to the rating of its explanation's UNCERTAIN-SUPPORT-SOU. A different computation has been implemented for each support evidence type. When the support is a contour hypothesis, this process computes two ratings: (1) a likelihood that the contour cannot be explained by *any* microstream (e.g., it is actually a noise contour), and (2) a likelihood that the contour does not actually support the microstream it is alleged to support. The first rating is based on the length and density of the contour, and the second rating is based on the length of the microstream and the length of the contour.

When the support is a microstream, this process computes the likelihood that the microstream does not support *any* stream (this is assumed to be very low) and when the support is a stream, this process computes the likelihood that the stream does not support *any* source (which, again, is assumed to be very low).

Each negative evidence SOU has an associated uncertainty that indicates the possibility there exists an explanation for it (e.g., microstream data may be missing as a result of improper parameter settings). The likelihood of this possibility is expressed in the negative-evidence-explanation rating of each negative evidence SOU's summary unit. The effect of negative evidence at the microstream level is computed using the time region covered by the negative evidence, the total length of the microstream (a function of the form $1 - e^{-x}$ where we map the total length of the microstream to the interval $[1, 8]$ and x is the SOU's time coverage) and the microstream region (attack, steady or decay) in which the negative evidence is located. The steady region is considered the most important region in a microstream and therefore, negative evidence for the steady region has a stronger effect on belief than negative evidence for the attack or decay regions. The negative evidence SOUs at the stream level are evaluated based on the number of microstreams in the stream definition and the importance of the missing microstream. In the current implementation, all the microstreams of a stream are considered to have equal importance. The negative evidence SOUs at the source level and the PARTIAL-SUPPORT-SOUs are evaluated in a similar manner.

Once all the SOUs have been rated, their summaries are combined into a single summary unit for the hypothesis. This unit's slots are initialized with the slot values of the minimum rated UNCERTAIN-SUPPORT-SOU. For each remaining

its support, then the existence of corroborating evidence has much less effect on the rating of the hypothesis extension.

UNCERTAIN-SUPPORT-SOU, the function for evaluating the negative evidence SOUs is used to compute a factor τ that will increase the summary unit's overall rating w by the relation $w_1 = w_0(1+\tau)$. For each negative evidence SOU, the summary unit's overall rating w is decreased by the relation $w_1 = w_0(1-\delta)$, where δ is the negative evidence SOU's overall rating. The PARTIAL-SUPPORT SOUs' ratings are also combined into a single number, in an algorithm similar to the one for the combination of UNCERTAIN-SUPPORT-SOUs.

```
#<source-ext.00006 D>  RATING: 0.12307032
                       PARTIAL-EVIDENCE-UNCERTAINTY: 0.29800
                       POSSIBLE-ALT-EXPLANATION-UNCERTAINTY: 0.74052006
                       POSSIBLE-ALT-SUPPORT-UNCERTAINTY: 0.0
                       CONSTRAINT-UNCERTAINTY: 0.0
                        ALT-EXTENSIONS-UNCERTAINTY:  0.045408897
                        TOP-LEVEL-ALT-EXTENSIONS: (<ALT-EXT-UNIT Alt:#<source-ext.00004 A>>)
                       NEGATIVE-EVIDENCE-UNCERTAINTY: 0.18336463
                       NEGATIVE-EVIDENCE-EXPLANATION: 0.09136386
```

Figure 23: The belief in this hypothesis is 0.12307032; it would increase by 0.29800 if the partial support were to be perfectly solved. 0.74052006 of the belief was reduced because off possible alternative explanations for the support, 0.045408897 because source A is an alternative source, and 0.09200077 because somewhere in the support there is negative evidence of strength 0.18336463 with likelihood of 0.09136386 to have an explanation.

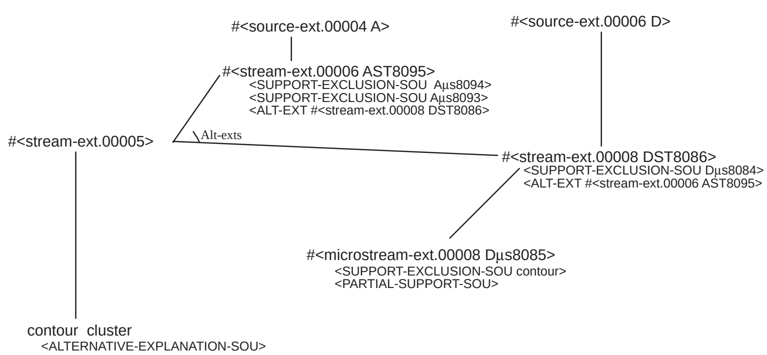

Figure 24: *The network of high level hypotheses that contributes to source D's belief.*

Figure 23 shows an answer hypothesis from the PSM in figure 16 with its summary unit. The network of highlevel hypotheses that contribute to this hypothesis' belief appears in figure 24. The partial-evidence-uncertainty slot value represents the PARTIAL-SUPPORT-SOU in the microstream Dμs8085, the negative-evidence-uncertainty slot value represents the SUPPORT-EXCLUSION-SOUs in the stream DST8086 and microstream Dμs8085, the alt-extensions-uncertainty slot value represents the ALT-EXTENSION-SOU in the stream DST8086, and the possible-alt-explanation-uncertainty slot value represents the POSSIBLE-ALTERNATIVE-EXPLANATION-SOU in the contour cluster.

This section represents our initial formulation of control strategies for IPUS. We expect to augment these strategies as we expand the source library to include real-world signals and increase the noise levels in the scenarios submitted to IPUS. The addition of highly context-specific SPAs will also require control strategy enhancements.

6 Conclusions and Future Research

This report presents an architecture for addressing signal understanding problems where the variety of possible input signal types makes it impossible to use one signal processing algorithm to process all the input signal types correctly. The IPUS paradigm provides a framework for structuring the cooperation that must take place between the search for appropriate SPAs and the search for interpretation models to explain the SPAs' output data.

We are presently evaluating the feasibility of the approach as implemented in our testbed. Initial experiments performed on the IPUS testbed indicate that the basic functionality of the major components and their interrelationships are realizable. Additionally, the system time performance seems reasonable given the unoptimized stage of its development. For example, on a TI Explorer II+ with 24 MB of physical memory, 301 MB of virtual memory, and garbage collection *disabled*, the 4-second scenario in section 4.2 required 240 seconds of real time for analysis by the unoptimized testbed. Of this, 89% (214 seconds) was spent executing signal processing algorithms[15]. We believe that SPAs implemented in hardware and optimized testbed code will bring the system closer to real-time performance levels. As future generations of hardware with faster timings become available, the "SPA bottleneck effect" will diminish.

The current implementation of the IPUS architecture has generated several research problems in areas such as sensor fusion, real-time system design, and system control. Our future IPUS research in the sensor fusion area will focus on how to unify interpretations of the same data obtained under different signal processing parameters. This is the **model-synthesis problem**. In the current implementation, our "synthesis" simply uses the results of the latest reprocessing to the exclusion of earlier processings. This approach works in the case where reprocessing produces results that only *refine* those of earlier processings. An instance of this occurs in the testbed example described earlier when "new" contours are found in block 2 to support microstream expectations for source A. Since they covered frequency regions *contained* within those of the short contours originally found in block 1, we were able to use the new contours without having to "integrate" them explicitly with the original contours.

There are certain situations where no single set of values for the control parameters of the SPA will eliminate all distortions in the SPA output at the same time. In these situations it will be necessary for the reprocessing KS to integrate the results of several reprocessings, each of which removes only *some* of the observed

[15] Coded in LISP

distortions. As an example of a situation where a more sophisticated approach to the model-synthesis problem is required, consider the processing of an electric motor sound that contains a speed-change transition. Let us assume that there have been no earlier speed-changes in the electric motor sound and that the front-end signal processing has therefore had its control parameters set to values that ensure detection of two steady-frequency components. Let us further assume that the system has no expectation for a speed change at the current time; instead it expects the two steady-frequency components to continue. However, when the portion of the signal containing the speed change is processed, the result shown in figure 25 is produced.

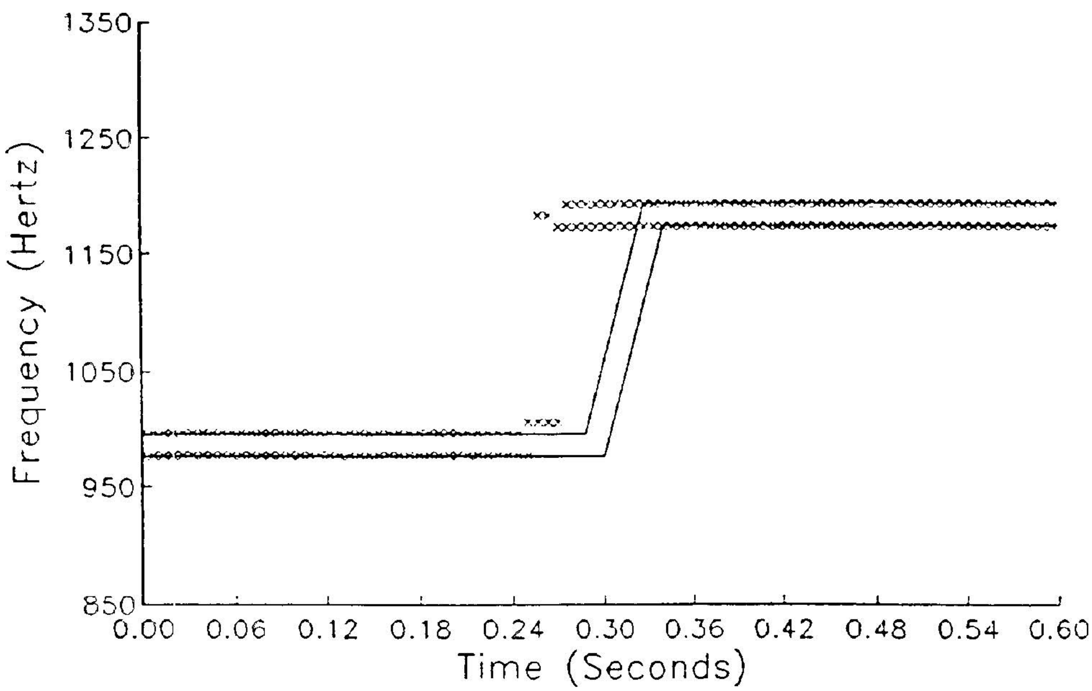

Figure 25: *A signal-processing output for a motor sound. The solid lines represent the actual frequency tracks while the* $\times$*'s represent the STFT output data. Time-resolution distortion causes the motor speed-change interval not to be tracked.*

The signal processing output is in conflict with the expectation of two steady-frequency components. The diagnosis process hypothesizes that the upper two microstreams in figure 25 are connected with the lower two microstreams and that this connection is missing in the SPA output due to a time-resolution distortion. This distortion arises when the STFT-instance has its window-length parameter set to a relatively large value. The signal re-processing planner then concludes that the STFT window-length should be decreased.

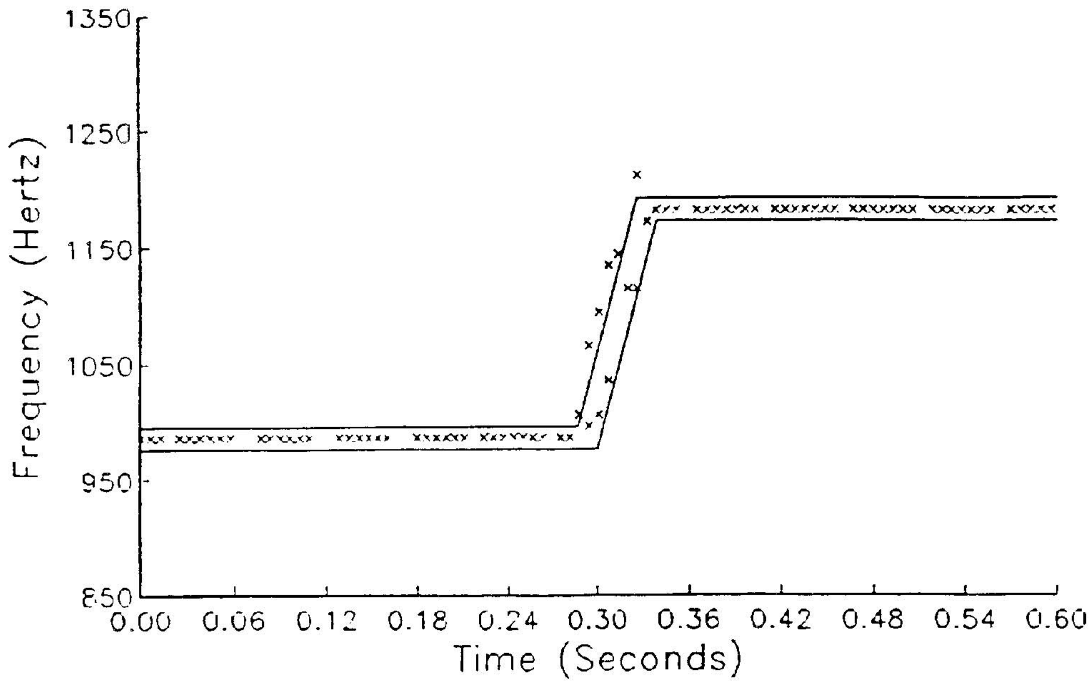

Figure 26: *Signal-processing output for a motor sound. The solid lines represent the actual frequency tracks while the* $\times$*'s represent the STFT output data. Frequency-resolution distortion causes the two frequency tracks to be merged into a single ghost-track.*

The subsequent execution of the re-processing plan results in the data shown in Figure 26. As expected, evidence is obtained for the speed-change, but now poor frequency resolution resulting from the shorter window-length does not resolve the two constituent frequency components. If we were to register this as a discrepancy, the system would become trapped in a discrepancy loop, since there is no single SPA instance that can capture both aspects of the signal. Thus, there is a need for the system to anticipate the new distortion and *integrate* the data from figures 25 and 26 as jointly representing evidence for the interpretation-model represented by the solid-lines. Preliminary research plans call for incorporating knowledge about this kind of model-synthesis in the reprocessing knowledge source. How the knowledge will be incorporated and how it will interact with other reprocessing knowledge based on SPA theory are open questions.

In the area of real-time system design, our IPUS research will focus on developing a framework for parameterizing knowledge sources to engage in approximate processing [6, 11]. The goal of this work will be to develop knowledge sources that are responsive to real-time constraints; if time constraints are tight, the knowledge sources should utilize approximate search algorithms and approximate data representations to provide less precise interpretations that are still useful to the system. Preliminary work in this area has been done for the diagnosis knowledge source. It is believed that adding a parameter to this knowledge source describing at what level in the data abstraction hierarchy the system is to perform explanation verification will provide a clean mechanism for expressing search and data approximation constraints. How the control plans will decide appropriate constraints, and what the exact time requirements for each abstraction level are remain open questions.

In complex signal-generating environments, it is possible that the particular scenario being monitored will generate signals whose characteristics (e.g., frequency shifts, volume changes, new sources) gradually depart from those that can be adequately processed by the front-end SPAs. Our current IPUS framework relies on localized reprocessing to remedy recognized inadequacies. We will explore an enhancement to the framework, **global parameter adaptation**, which adapts the front-end parameters as overall scenario characteristics change. The problem of deciding when to shift front-end processing parameter values at a global level (i.e., the default parameter values as opposed to the localized changes made within the reprocessing knowledge source) is a key issue in the area of system control. Certain cues such as increased number of reprocessings, identification of source behavioral changes, etc. should be exploited by the system in order to decide when to adapt front-end processing parameters instead of simply relying on reprocessings with parameter-value changes localized to distorted signal regions. Changes in the set of scenario expectations and the priorities attached to recognizing particular sources should also play major roles in selecting new front-end processing configurations. Future work in this area will focus on selecting the cues and on developing a formal framework for relating changes in these cues, source priorities, and scenario expectations to decisions on switching front-end processing parameter values and/or SPAs. A cost-benefit model based on decision theory or signal-detection theory may prove useful in this research problem [4].

The development of this testbed thus far has had a strong empirical nature. That is, given knowledge of the range of scenarios we could generate, we supplied IPUS with a set of SPAs we believed adequate for the interpretation tasks it could face. However, no formal analysis was ever performed on the scenarios to determine a priori what algorithms would be needed and *where* in the data streams processing by two or more SPAs would be required. Work in this area is closely linked to the **SPA model variety problem** [15]. This problem focuses on the relationship between SPAs and the classes of signals for which they can produce undistorted outputs. A signal understanding system needs to use more than one SPA if there does not exist a single SPA that can produce undistorted output for all the possible input signals in the given application domain. In other words, the input signal must satisfy the conditions in the data-model for that SPA. If the SPA does not satisfy the conditions, it is said to suffer from a model variety problem with respect to the input signals. Formal analysis of this relationship between SPAs and input signal characteristics will permit us to predictably tailor IPUS's algorithm database to specific scenario classes. This ability will enable us to formally evaluate the focusing heuristics in terms of the ratio between the amount of data reprocessed and the minimum required amount of reprocessing.

7 Acknowledgments

We thank Erkan Dorken and Avram Weiss at Boston University for their work in developing SPA theory used in demonstrating IPUS.

Given the large number of contributors to this effort it seems appropriate to give specific credit to individuals for their contributions. Victor Lesser and Hamid Nawab were responsible for the overall concept of IPUS and the management of project details. Malini Bhandaru designed and implemented the interpretation and differential diagnosis knowledge sources. Norman Carver was responsible for the RESUN architecture and provided invaluable assistance in the mapping of IPUS concepts onto the RESUN framework. He also assisted in the design of several IPUS architecture components. Zarko Cvetanovic designed and implemented the original version of the signal reprocessing knowledge source. Izaskun Gallastegi designed and implemented the control plans and focusing heuristics, and also played a major role in debugging the system to obtain its current performance level. Finally, Frank Klassner designed and implemented the discrepancy detection and diagnosis knowledge sources. He also implemented the signal reprocessing knowledge used in the IPUS version described here.

References

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

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

[3] N. Carver, *Sophisticated Control for Interpretation: Planning to Resolve Uncertainty*, Ph.D. Thesis, Computer and Information Science Department, University of Massachusetts, 1990.

[4] Coombs, Dawes, Tversky, ch 6, *Mathematical Psychology, An Elementary Introduction*, Prentice-Hall, 1970.

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

[6] K. Decker, V. Lesser, R. Whitehair, "Extending a Blackboard Architecture for Approximate Processing", *Journal of Real-Time Systems 2*, pp 47–79, Kluwer Academic Publishers, 1990.

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

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

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

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

[11] V. Lesser, J. Pavlin, and E. Durfee, "Approximate Processing in Real-Time Problem Solving", AI Magazine, Spring 88, Vol. 9, No. 1, pp 49–61.

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

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

[14] H. Nawab and V. Lesser, "High-Level Adaptive Signal Processing", *NAIC Final Report*, vol 17, October 1989.

[15] H. Nawab and V. Lesser, "Integrated Processing and Understanding of Signals", *Knowledge-Based Signal Processing*, A. Oppenheim and H. Nawab, eds. 1991.

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

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

[18] Y. Peng and J. Reggia, "Plausibility of Diagnositic Hypotheses: The Nature of Simplicity," *AAAI '86 Proceedings*, pp 140–145, 1986.

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

Appendix

The following is a hardcopy of the trace file for the experiment described in this paper.

```
   PROBLEM SOLVING MODEL:

    ANSWERS:
    SOUS:   <NO-EVIDENCE-SOU>



**** Processing Block  # 1 from time 1 to time 10000



 Loading next data set........9999 data points read

     Gathering Spectral Information
     Contours found: #<contour-ext.00001>
                     #<contour-ext.00002>
                     14 short contours found

 **** Explaining hypothesis #<contour-ext.00002>
     1 possible source
      Source B,
     based in 1 microstream
      - confirmed microstreams at ([986 1006]) Hz
        from time 0 to time 9216
      - unconfirmed microstreams at ([1041 1059]) Hz[1041 1059] Hz

**** Attempting to confirm source B MICROSTREAM at [1041 1059] Hz
     contour #<contour-ext.00001> confirms microstream at [1042 1061] Hz

**** Attempting to explain unexplained data
     to resolve uncertainty by too many unexplained data
     Frequency cluster indicative of a source with
     13 short contours found at [1200 1300] Hz

**** Explaining Contour Cluster
     2 possible sources
      Source A,
```

```
    - missing microstreams at ([1211 1229] [1191 1209]) Hz
    Source D,
    - unconfirmed microstreams at ([2191 2209]) Hz
    - missing microstreams at ([1201 1219]) Hz[2191 2209] Hz

**** Attempting to confirm source D MICROSTREAM at [2191 2209] Hz
     Unconfirmed microstream at [2191 2209] Hz

**** Differential Diagnosis between sources A and D
     - Reprocess in time [0 10000]
     - Reprocessing goals:
        1- Differentiate between
            -source A microstreams
              at ([1200 1200] [1220 1220]) Hz
            -source D microstreams
              at ([1201 1219]) Hz
        2- Seek source D microstream at [2191 2209] Hz

  Performing Reprocessing for Differential Diagnosis
  Parameter *STFT-PEAK-ENERGY-THRESHOLD* has local value 0.09
  Parameter *FFT-SIZE* has local value 2048
  Differential Diagnosis Reprocessing completed.

    Source D has become disbelieved
    Source A microstreams confirmed
    at [1211 1229] Hz[1191 1209] Hz
     contour #<contour-ext.00024> confirms microstream at [1196 1206] Hz
     contour #<contour-ext.00023> confirms microstream at [1216 1226] Hz
    Source D microstream not found at ([2191 2209]) Hz



   PROBLEM SOLVING MODEL:

    ANSWERS:<ANSWER-PS-MODEL Answer:#<source-hyp.00002 A>>
            <ANSWER-PS-MODEL Answer:#<source-hyp.00001 B>>

    SOUS:   <UNCERTAIN-ANSWER-SOU #<source-hyp.00002 A> rating: 0.48110092
                     possible-alt-explanation-uncertainty: 0.337713
                     alt-extensions-uncertainty:           0.08670694
                     alternative-extensions:               (#<source-ext.00006 D>)
                     negative-evidence-uncertainty:        0.13634779
```

```
        negative-evidence-explanation:        0.041868664
        partial-evidence-uncertainty:         0.44275594
<UNCERTAIN-ANSWER-SOU #<source-hyp.00001 B> rating: 0.7440413
        possible-alt-explanation-uncertainty: 0.25171039
        negative-evidence-uncertainty:        0.014161205
        negative-evidence-explanation:        0.009912843
        partial-evidence-uncertainty:         0.54788303
<NONANSWER-PS-MODEL hyp: #<source-hyp.00003 D> rating: 0.086706944
        possible-alt-explanation-uncertainty: 0.73317057
        possible-alt-support-uncertainty:     0.009000005
        alt-extensions-uncertainty:           0.015626855
        alternative-extensions:               (#<source-ext.00010 A>)
        negative-evidence-uncertainty:        0.17133908
        negative-evidence-explanation:        0.0068433946
<NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00022> #<contour-hyp.00021>
                         #<contour-hyp.00020> #<contour-hyp.00019>
                         #<contour-hyp.00018> #<contour-hyp.00017>))>
        rating: 0.75
        possible-alt-explanation-uncertainty: 0.25
<NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00009>))>
        rating: 0.875
        possible-alt-explanation-uncertainty: 0.125
<NONANSWER-PS-MODEL Time:[1 10000]>
<NO-EVIDENCE-SOU>


**** Processing Block  # 2 from time 10001 to time 20000


 Loading next data set........9999 data points read

     Gathering Spectral Information
     Contours found: #<contour-ext.00027>
                     #<contour-ext.00028>
                     15 short contours found

     Data-Data discrepancy detected between TD and STFT results
     Energy increase detected by TD and not detected by STFT
    in region [12049 13201]
     Diagnosis explanation: <(CONTOUR-TIME-RESOLUTION)>
```

```
 Performing Discrepancy Reprocessing for MISSING-STFT-CONTOUR-PRESENT-TD-CONTOUR
 of type FAULT
      Parameter  *STFT-OVERLAP*  has new local value: 128
      Parameter  *STFT-INTERVAL*  has new local value: 256
      Parameter  *STFT-PEAK-ENERGY-THRESHOLD*  has new local value: 0.9
 Synthesizing the following contours:
 (#<contour-ext.00042> #<contour-ext.00043>)
 Discrepancy Reprocessing Completed


**** Attempting to confirm source B MICROSTREAM at [1042 1061] Hz
     contour #<contour-ext.00027> confirms microstream at [1042 1062] Hz

**** Attempting to confirm source B MICROSTREAM at [986 1006] Hz
     contour #<contour-ext.00028> confirms microstream at [986 1006] Hz

**** Attempting to confirm source A MICROSTREAM at [1196 1206] Hz
     Unconfirmed microstream at [1196 1206] Hz

**** Attempting to confirm source A MICROSTREAM at [1216 1226] Hz
     Unconfirmed microstream at [1216 1226] Hz

**** Attempting to confirm source A MICROSTREAM at [1196 1206] Hz
     Unconfirmed microstream at [1196 1206] Hz

**** Attempting to confirm source A MICROSTREAM at [1216 1226] Hz
     Unconfirmed microstream at [1216 1226] Hz

****Performing Discrepancy Diagnosis.
    INITIAL: (#<microstream-ext.00015> #<microstream-ext.00014>)
    FINAL: NIL

      Discrepancy Explanation Proposed: (#<PLAN-OPERATOR 55271727>).


 Performing Discrepancy Reprocessing for MISSING-MICROSTREAM of type CONFLICT
      Parameter  *FFT-SIZE*  has new local value: 2048
      Parameter  *STFT-PEAK-ENERGY-THRESHOLD*  has new local value: 0.27
   Synthesizing the following microstreams:
        (#<microstream-ext.00020> #<microstream-ext.00022>)
```

```
Discrepancy Reprocessing Completed


 **** Explaining hypothesis #<contour-ext.00042>
     1 possible source
      Source E,
     based in 1 microstream
      - confirmed microstreams at ([1992 2012]) Hz
        from time 12073 to time 12949

**** Attempting to explain unexplained data
    to resolve uncertainty by too many unexplained data
     Frequency cluster indicative of a source with
     13 short contours found at [1200 1300] Hz

**** Explaining Contour Cluster
      Source A,



   PROBLEM SOLVING MODEL:

    ANSWERS:<ANSWER-PS-MODEL Answer:#<source-hyp.00004 E>>
            <ANSWER-PS-MODEL Answer:#<source-hyp.00002 A>>
            <ANSWER-PS-MODEL Answer:#<source-hyp.00001 B>>

    SOUS:   <UNCERTAIN-ANSWER-SOU #<source-hyp.00004 E> rating: 0.7582098
                     possible-alt-explanation-uncertainty: 0.24179024
                     partial-evidence-uncertainty:         0.17878717
            <UNCERTAIN-ANSWER-SOU #<source-hyp.00002 A> rating: 0.59019625
                     possible-alt-explanation-uncertainty: 0.19123594
                     alt-extensions-uncertainty:           0.06798804
                     alternative-extensions:               (#<source-ext.00006 D>)
                     negative-evidence-uncertainty:        0.2052753
                     negative-evidence-explanation:        0.054695465
                     partial-evidence-uncertainty:         0.024274237
            <UNCERTAIN-ANSWER-SOU #<source-hyp.00001 B> rating: 0.7619481
                     possible-alt-explanation-uncertainty: 0.2291733
                     negative-evidence-uncertainty:        0.029595304
                     negative-evidence-explanation:        0.020716712
                     partial-evidence-uncertainty:         0.5210847
            <NONANSWER-PS-MODEL hyp: #<source-hyp.00003 D> rating: 0.06798807
```

```
                possible-alt-explanation-uncertainty: 0.73317057
                possible-alt-support-uncertainty:     0.009000005
                alt-extensions-uncertainty:           0.03434573
                alternative-extensions:               (#<source-ext.00016 A>)
                negative-evidence-uncertainty:        0.17133908
                negative-evidence-explanation:        0.0068433946
         <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00043>)>
                rating: 0.875
                possible-alt-explanation-uncertainty: 0.125
         <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00029> #<contour-hyp.00032>)>
                rating: 0.88235295
                possible-alt-explanation-uncertainty: 0.11764705
         <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00022> #<contour-hyp.00021>
                                  #<contour-hyp.00020> #<contour-hyp.00019>
                                  #<contour-hyp.00018> #<contour-hyp.00017>)>
                rating: 0.75
                possible-alt-explanation-uncertainty: 0.25
         <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00009>)>
                rating: 0.875
                possible-alt-explanation-uncertainty: 0.125
         <NONANSWER-PS-MODEL Time:[10001 20000]>
         <NONANSWER-PS-MODEL Time:[1 10000]>
         <NO-EVIDENCE-SOU>


**** Processing Block  # 3 from time 20001 to time 30000


 Loading next data set........9999 data points read

     Gathering Spectral Information
     Contours found: #<contour-ext.00051>
                     #<contour-ext.00052>
                     7 short contours found

**** Attempting to confirm source B MICROSTREAM at [1042 1062] Hz
     contour #<contour-ext.00051> confirms microstream at [1042 1062] Hz

**** Attempting to confirm source B MICROSTREAM at [986 1006] Hz
     contour #<contour-ext.00052> confirms microstream at [986 1006] Hz
```

```
**** Attempting to explain unexplained data
    to resolve uncertainty by too many unexplained data



   PROBLEM SOLVING MODEL:

    ANSWERS:<ANSWER-PS-MODEL Answer:#<source-hyp.00004 E>>
            <ANSWER-PS-MODEL Answer:#<source-hyp.00002 A>>
            <ANSWER-PS-MODEL Answer:#<source-hyp.00001 B>>

    SOUS:   <UNCERTAIN-ANSWER-SOU #<source-hyp.00004 E> rating: 0.7582098
                     possible-alt-explanation-uncertainty: 0.24179024
                     partial-evidence-uncertainty:         0.17878717
            <UNCERTAIN-ANSWER-SOU #<source-hyp.00002 A> rating: 0.59019625
                     possible-alt-explanation-uncertainty: 0.19123594
                     alt-extensions-uncertainty:           0.06798804
                     alternative-extensions:              (#<source-ext.00006 D>)
                     negative-evidence-uncertainty:        0.2052753
                     negative-evidence-explanation:        0.054695465
                     partial-evidence-uncertainty:         0.024274237
            <UNCERTAIN-ANSWER-SOU #<source-hyp.00001 B> rating: 0.7803583
                     possible-alt-explanation-uncertainty: 0.20576218
                     negative-evidence-uncertainty:        0.046264954
                     negative-evidence-explanation:        0.032385465
                     partial-evidence-uncertainty:         0.43503252
            <NONANSWER-PS-MODEL hyp: #<source-hyp.00003 D> rating: 0.06798807
                     possible-alt-explanation-uncertainty: 0.73317057
                     possible-alt-support-uncertainty:     0.009000005
                     alt-extensions-uncertainty:           0.03434573
                     alternative-extensions:              (#<source-ext.00016 A>)
                     negative-evidence-uncertainty:        0.17133908
                     negative-evidence-explanation:        0.0068433946
            <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00053> #<contour-hyp.00054>
                                     #<contour-hyp.00055> #<contour-hyp.00056>
                                     #<contour-hyp.00057> #<contour-hyp.00058>
                                     #<contour-hyp.00059>))
                     rating: 0.7
                     possible-alt-explanation-uncertainty: 0.3
            <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00043>))
                     rating: 0.875
```

```
                possible-alt-explanation-uncertainty: 0.125
        <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00029> #<contour-hyp.00032>)>
                rating: 0.88235295
                possible-alt-explanation-uncertainty: 0.11764705
        <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00022> #<contour-hyp.00021>
                                 #<contour-hyp.00020> #<contour-hyp.00019>
                                 #<contour-hyp.00018> #<contour-hyp.00017>)>
                rating: 0.75
                possible-alt-explanation-uncertainty: 0.25
        <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00009>)>
                rating: 0.875
                possible-alt-explanation-uncertainty: 0.125
        <NONANSWER-PS-MODEL Time:[20001 30000]>
        <NONANSWER-PS-MODEL Time:[10001 20000]>
        <NONANSWER-PS-MODEL Time:[1 10000]>
        <NO-EVIDENCE-SOU>


**** Processing Block  # 4 from time 30001 to time 40000


 Loading next data set........9999 data points read

     Gathering Spectral Information
     Contours found: #<contour-ext.00062>
                     #<contour-ext.00077>
                     #<contour-ext.00081>
                     #<contour-ext.00082>
                     #<contour-ext.00083>
                     19 short contours found

     Data-Data discrepancy detected between TD and STFT results
     Energy increase detected by TD and not detected by STFT
    in region [32817 33457]
     Diagnosis explanation: <(CONTOUR-TIME-RESOLUTION)>

 Performing Discrepancy Reprocessing for MISSING-STFT-CONTOUR-PRESENT-TD-CONTOUR
 of type FAULT
      Parameter  *STFT-OVERLAP*  has new local value: 128
      Parameter  *STFT-INTERVAL*  has new local value: 256
```

```
     Parameter  *STFT-PEAK-ENERGY-THRESHOLD*  has new local value: 0.9
Synthesizing the following contours:
(#<contour-ext.00085> #<contour-ext.00084>)
Discrepancy Reprocessing Completed


**** Explaining hypothesis #<contour-ext.00081>
    1 possible source
     Source C,
    based in 1 microstream
     - confirmed microstreams at ([1094 1113]) Hz
     - unconfirmed microstreams at ([1791 1809]) Hz[1791 1809] Hz

**** Attempting to confirm source C MICROSTREAM at [1791 1809] Hz
     contours (#<contour-ext.00077> #<contour-ext.00053> #<contour-ext.00054>
               #<contour-ext.00060> #<contour-ext.00066> #<contour-ext.00067>
               #<contour-ext.00070> #<contour-ext.00071> #<contour-ext.00074>)
     confirm microstream at [1784 1804] Hz

**** Attempting to confirm source B MICROSTREAM at [1042 1062] Hz
     contour #<contour-ext.00082> confirms microstream at [1042 1064] Hz

**** Attempting to confirm source B MICROSTREAM at [986 1006] Hz
     contour #<contour-ext.00083> confirms microstream at [986 1006] Hz



  PROBLEM SOLVING MODEL:

   ANSWERS:<ANSWER-PS-MODEL Answer:#<source-hyp.00005 C>>
           <ANSWER-PS-MODEL Answer:#<source-hyp.00004 E>>
           <ANSWER-PS-MODEL Answer:#<source-hyp.00002 A>>
           <ANSWER-PS-MODEL Answer:#<source-hyp.00001 B>>

   SOUS:   <UNCERTAIN-ANSWER-SOU #<source-hyp.00005 C> rating: 0.5497634
                    possible-alt-explanation-uncertainty: 0.45023665
                    partial-evidence-uncertainty:         0.6916429
           <UNCERTAIN-ANSWER-SOU #<source-hyp.00004 E> rating: 0.7582098
                    possible-alt-explanation-uncertainty: 0.24179024
                    partial-evidence-uncertainty:         0.17878717
           <UNCERTAIN-ANSWER-SOU #<source-hyp.00002 A> rating: 0.59019625
                    possible-alt-explanation-uncertainty: 0.19123594
```

```
        alt-extensions-uncertainty:          0.06798804
        alternative-extensions:             (#<source-ext.00006 D>)
        negative-evidence-uncertainty:       0.2052753
        negative-evidence-explanation:       0.054695465
        partial-evidence-uncertainty:        0.024274237
<UNCERTAIN-ANSWER-SOU #<source-hyp.00001 B> rating: 0.7194943
        possible-alt-explanation-uncertainty: 0.2436752
        negative-evidence-uncertainty:       0.12276849
        negative-evidence-explanation:       0.08593795
        partial-evidence-uncertainty:        0.30271155
<UNCERTAIN-HYPOTHESIS-SOU hyp: #<contour-hyp.00084> rating: 0.99
        possible-alt-explanation-uncertainty: 0.010000005
<UNCERTAIN-HYPOTHESIS-SOU hyp: #<contour-hyp.00062> rating: 0.99
        possible-alt-explanation-uncertainty: 0.010000005
<NONANSWER-PS-MODEL hyp: #<source-hyp.00003 D> rating: 0.06798807
        possible-alt-explanation-uncertainty: 0.73317057
        possible-alt-support-uncertainty:    0.009000005
        alt-extensions-uncertainty:          0.03434573
        alternative-extensions:             (#<source-ext.00016 A>)
        negative-evidence-uncertainty:       0.17133908
        negative-evidence-explanation:       0.0068433946
<NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00085>)>
        rating: 0.8
        possible-alt-explanation-uncertainty: 0.19999999
<NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00061> #<contour-hyp.00063>
                         #<contour-hyp.00064> #<contour-hyp.00065>
                         #<contour-hyp.00068> #<contour-hyp.00069>
                         #<contour-hyp.00072> #<contour-hyp.00073>
                         #<contour-hyp.00075> #<contour-hyp.00076>
                         #<contour-hyp.00078> #<contour-hyp.00079>
                         #<contour-hyp.00080>)>
        rating: 0.4583333
        possible-alt-explanation-uncertainty: 0.5416667
<NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00055> #<contour-hyp.00056>
                         #<contour-hyp.00057> #<contour-hyp.00058>
                         #<contour-hyp.00059>)>
        rating: 0.78571427
        possible-alt-explanation-uncertainty: 0.21428573
<NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00043>)>
        rating: 0.875
        possible-alt-explanation-uncertainty: 0.125
<NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00029> #<contour-hyp.00032>)>
```

```
          rating: 0.88235295
          possible-alt-explanation-uncertainty: 0.11764705
  <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00022> #<contour-hyp.00021>
                           #<contour-hyp.00020> #<contour-hyp.00019>
                           #<contour-hyp.00018> #<contour-hyp.00017>))>
          rating: 0.75
          possible-alt-explanation-uncertainty: 0.25
  <NONANSWER-PS-MODEL Hyp:(#<contour-hyp.00009>))>
          rating: 0.875
          possible-alt-explanation-uncertainty: 0.125
  <NONANSWER-PS-MODEL Time:[30001 40000]>
  <NONANSWER-PS-MODEL Time:[20001 30000]>
  <NONANSWER-PS-MODEL Time:[10001 20000]>
  <NONANSWER-PS-MODEL Time:[1 10000]>
  <NO-EVIDENCE-SOU>
```

*NOTE: all the summary-unit slots with rating 0.0 have been edited out from this trace to improve the readability.