The Harmonic Process State (HPS) Transform

Dr. Nelson R. Manohar Alers[1], Member, IEEE

Abstract – In this paper, we introduce a revolutionary class of stationary-based approximations that exhibits potential for high compressibility under rigorous error-control and known confidence. Under small delay and nominal overhead, we generate an approximation signal that is robust, stable, unbiased, and an accurate tracker of finite bursts of stability on the original signal. To this end, statistical filtering is used to identify time segments exhibiting said *approximate* stationary properties. The original signal is transformed into a set of well-behaved tracking signals used to span a decision-making space suitable for robust sequential decision-making. Computationally efficient inferences uncover the *approximate* presence and duration of localized stationary conditions on the original signal. These segments are thus referred to as "Approximate Temporally-Stable" (ATS) segments. Although the location and duration of localized stationary conditions is *random* and *unknown*, the resulting time series of unearthed ATS segments describes a minimal variability trajectory over *long-term* stationary conditions. The approach, referred to as the Harmonic Process State (HPS) transform, exhibits desirable qualities on implementation ease, algorithmic complexity, computational stability, signal compressibility, decision-making robustness, information loss, and error behavior. Although the HPS transform is particularly suited for process control, the results presented here have vast implications to other fields.

Index Terms: measurement techniques (C4C), data compaction and compression (E4A), approximation (G12), mining methods and algorithms (H28D), and control theory (I28C).

I. Introduction

Imagine that you could take any signal and transform it into an operational representation, much more compact but nevertheless functionally equivalent – that is, a representation that made it possible to robustly perform operations (and in particular, inferences) over it with nominal overhead but bearing direct relevance over the original signal. There is *no* such "free lunch"; however, an "approximation equivalency" may be close to it. The success of such would depend on the robustness, accuracy, overhead, and delay associated with the approximation. In this paper, we show one such result. We show that by trading some (but specifiable) delay, we can generate – in a robust, accurate, and under optimal overhead – an "approximation equivalency" possessing the above described features.

Our approach is deceitfully simple yet robust and lush; we generate a stationary-based approximation of any input signal. This approximation models the signal – when feasible and under consistent confidence – as a time series of error-constrained "constant-value states" where consecutive pairs of such random-duration "states" are interconnected by non-stationary fine-tracking of the original signal.

In foresight to presented findings, the above is partly possible because *most* practical signals (e.g., financial time series, manufacturing process indicators, most sensor measurements) exhibit patterns of localized semi-stationary conditions. Often, these patterns are associated with performance states of the random process being observed (e.g., financial indexes, quality

[1] Contact address for the author is Urb. Santa Maria CF8, San German, PR, USA 00683. The author can be contacted at nelsonmanohar@yahoo.com.

[5] Specifically, when compared to the source's outgoing bit rate, the receiver's incoming bit rate is adapted (i.e. distorted) – in accordance to some optimality constraint.

[28] In this discussion, we abstract how an optimal representative value is chosen or updated.

[29] There are O(2[illegible]N (k-1)) such merge processes (and thus, segment-sets) at an arbitrary level k.

[49] Through careful construction, this approximate test unearths both mean and variance departures. See [LAPIN:CP] for a review of sampled population means and variances tests.

[56] Bear in mind the concept of undecidability, asserts that notions of start, continuance, end, etc. can not be used in a deterministic fashion but rather in a probabilistic one.

indicators, temperature readings).[2] However, there are several constraints to the unearthing of these patterns. First, these localized semi-stationary conditions may be randomly scattered. Second, these localized semi-stationary conditions may exhibit widely varying durations. Last, these patterns of localized semi-stationary conditions may be exist only at specific but unknown timescales.

Moreover, by virtue of unearthing patterns of localized semi-stationary conditions from *any* signal, the approach also unearths *approximate* time-scale information. Specifically, a measurements artifact (specified with nominal user input toward the intertwining of delay and confidence) allows the unearthing of timescale information. Caveat emptor; this is possible as long as such timescale is both greater than this measurement artifact as well as greater than the sampling timescale. We are unaware of any (efficient or not) algorithmic work on uncovering of time-scale information or patterns of semi-stationary conditions from an arbitrary signal.

For this and similar reasons, the above represents a fundamental result bearing significance and noteworthy relevance on many fields.[3] For example, it would allow for detecting states and transitions within a signal as well as for desensitizing sensors to noise in a signal, which in turn, makes possible new forms of adaptive decision-making control, signal representation, signal compression, substring search, etc.

The above-described problem relates to an area referred to as sampling inversion. Sampling inversion represents "the process of making *partial* observations of a system, and drawing inferences about the *full* behavior of the system," a process that is constrained "with *minimizing* information loss whilst *reducing* the volume of collected data [**KUROSE:JSAC-CFP**]." Our work in sampling inversion started early on while attempting to address stable rate control for adaptive resource management. In 1995 [**NRM_ACMMM95**], we introduced an approach for adapting the schedule of heterogeneous streams (a "rate control" problem) within the framework of statistical process performance (a "process control" problem). Then, in 1997 [**NRM_MMCN98**], we further extended this framework to distributed multimedia and network measurements. There we introduced the concept of statistical filtering of temporal stability, explored some of its applications to the adaptive rate control problem, and applied it to network measurements (see **Fig. 1**).

Fig. : Data vs. inference transfer models. The left panel models transfer of information in terms of an intermediary representation created for optimal signal reconstruction and high compressibility. Right panel models the transfer of information in terms of interferences about the hidden random process and its consequences over signal compressibility and signal reconstruction. We show that if based on the HPS transform, sampling and measurement applications could greatly benefit from this representation.

This work was seminal to many *unacknowledged* derivative research efforts. Then, in 1999 [**NRM_USPTO99**], we showed in a series of systems, how by taking advantage of the stability properties we could devise its application to scalable, loosely coupled, and autonomous distributed resource management. Here, those *preliminary* reported

[2] For example, temperature is typically stationary within a segment of a given day. Such is a result of constraints associated with the physical process being sampled (i.e., the weather).

[3] See Article VI for a review of these and other applications.

concepts (underlying the above systems research) are now *rigorously* developed into broad theoretical results. Specifically, we rigorously examine the foundation of statistical filtering of "**localized stationary condition**s" that we pioneered early on and significantly extend on this. Specifically, we introduce the concept of "approximate τ-invariance" and describe here what we refer to as to the **Harmonic Process State (HPS) transform**.

The **HPS transform** is shown to have most desirable qualities in complexity, simplicity, robustness, error, compressibility, and information loss. Extraordinarily, the **HPS transform** achieves all this under nominal computational and memory overhead, and more importantly, under consistent confidence levels and bounded error. We analyze these results and meticulously study capabilities, tradeoffs, and limitations.

Information Theory[4]

*"Stems from the work of American mathematician and electrical engineer **Claude E. Shannon** and, in particular, his classic paper **"A Mathematical Theory of Communication,"** published in 1948 in the **Bell System Technical Journal**. Information theory focuses on the problems inherent in sending and receiving messages and information.*

1. *The theory is based on the idea that communication involves uncertain processes, both in the selection of the message to be transmitted and in transmission... Information theory provides a way to measure this uncertainty precisely... [The] actual meaning of the message is unimportant. Instead, the important qualities of communication are the amount of information that the message contains, the accuracy of the transmission, and the quality of the reception... Information theory measures the amount of information in a message by using bits [... and] provides a way to find the minimum number of bits required to communicate a given message... Information theory can provide a way to measure the amount of information produced by a source or to measure the ability of a noisy channel to transmit information reliably.*
2. *[The] theory provides the theoretical basis for data compression, ... a way to squeeze more information into a message by eliminating redundancy, or parts of the message that do not contain any important information... [It] provides a method for determining exactly how many bits are required to specify a given message to a given precision. This method is called the theory of data compression or, more technically, rate distortion theory. As the acceptable distortion becomes smaller and smaller, the required number of bits becomes larger and larger. Conversely, as the allowed distortion becomes larger, the required number of bits decreases. Ultimately, the number of required bits becomes zero. The number becomes zero when the allowed distortion can be achieved by merely guessing at the message. **Shannon's Fundamental Theorem of Data Compression** states that it is possible to compress a message to a given level, but no more."*

Because a bit stream is compressed (i.e., *transformed*) by a source, transmitted, and then uncompressed (i.e., *untransformed*) at a receiver, solutions to an information theory problem are also referred to as "rate distortion". This is key to understanding our approach; distortion is applied in accordance to *some* optimality constraint.[5].

Section 1.1: Background

sampling represents a function $f(\cdots)$ (performed by an observer A) which selects and/or collects time-ordered observations $\langle y(i)\rangle=\langle f\langle x(i)\rangle\rangle$ from some random process $\{X\}$.[6] Sampling inversion is therefore, the process of attempting to reconstruct the random process $\{X\}$ from said observations $\langle y(i)\rangle$. This process is particularly constrained by the fact that the random process is typically hidden (i.e., unknown) while the sampling function $f(\cdots)$ itself may constitute an unqualified hindrance (e.g., it may be preset, ad-hoc, or unknown) to signal fidelity. Consequentially, observations $\langle y(i)\rangle$ could be subject to error, correlation, distortion, and/or bias during sampling.[7] That is, sampling inversion is typically an incompletely

[4] Extract from article by Prof. Robert J. McEliece from Microsoft Encarta 2003 ©.

[6] This often trivialized sampling function $f(\cdots)$ is supposed to be applied according to some criteria (e.g., randomness, time scaling, etc.); which often it is not (or sometimes it cannot be) formulated a-priori.

[7] See [REF:SAMPLING ERROR] for an introduction to error sources in sampling.

specified problem.

Fortunately, we are able to approach formally the problem through a digression. To this end, note that sampling inversion is inherently an information theory problem. information theory deals with encoding information transmitted (from sources to receivers) subject to the "minimization" of information loss possibly due to the presence of noise across the delivery channel.[8] **Sidebar *"Information Theory"*** provides insight into information theory.

The fundamental information theory problem deals with two sub-problems: (1) optimality of signal fidelity and (2) optimality of signal compressibility. From an information theory perspective, the conveying of knowledge about ***{X}*** from ***A*** to some other, say ***B***, could take place just as well in terms of the sampled ***‹y(i)›*** or in terms of some more compact (and more resilient to noise) form – for example, an optimal compression ***{I$_{AB}$}*** of ***‹y(i)›***. Now, recall that the *ultimate* goal of sampling inversion is to enable inferential reasoning regarding the random process being sampled. This retrospection encourages us to transform the underlying transmission model from raw data into one centered around inferences.

Stationary-Based Encoded Representation

Consider an arbitrary signal ***‹y(i)›.*** As shown above, given a **localized stationary condition**, the approach encodes awareness of such random-length stationary time segment into a constant-valued approximation. As shown, such can make approachable theoretical limiting values of signal compressibility (see "*compressed signal*") – *even* for common signals. However, such requires the robust detection of **localized stationary conditions** (see "*decision bit*") and the robust selection of the constant-valued approximation (see "*stationary approximation*"). We show such result.

Note that the implicit compression dictionary is most minimal; a single bit codes the presence (or absence) of **localized stationary condition**s. Moreover, pairs of instances of this bit allow to determine the duration of any such **localized stationary condition**.

Specifically, in our approach, transmission of information takes place through robust inferences[9] ***{I$^*_{AB}$}*** rather than on just sampled observations ***‹y(i)›*** about the underlying random process ***{X}*** (see **Fig. 1**). This way, theoretical limiting values of signal compressibility (i.e., c ***{I$_{AB}$}*** vs. ***{I$^*_{AB}$}***) become plausible through (inference) redundancy tradeoff *w.r.t* (*with respect to*)[10] tolerable values of signal fidelity (i.e., compare ***{I}*** vs. ***{I$^*_{AB}$}***) for arbitrary signal ***‹y(i)›***. By choosing a robust inference, we take advantage of an inference-based approximation to control information loss in such a way to achieve reproduction fidelity suited for certain types of decision-making. Specifically, we formulate signal compressibility of ***‹y(i)›*** in terms of robust inferential arguments made *w.r.t.* the approximate presence (or not) of localized stability conditions on the underlying random process ***{X}***.

Our approach transforms sampling inversion into an information theory problem through a data reduction applied *w.r.t.* to

1. the approximate presence of **localized stationary condition**s (i.e., a signal compressibility constraint) and
2. the accuracy of a constant-valued approximation to

[8] See [REF:INFORMATION THEORY] for an introduction to information theory.

[9] Example of inferences are "the random process {X} is (or is not) in a state of statistical quality control within 3σ limits [MONTGOMERY:SQC]", or "the random process {X} exhibits semi-ergodic properties w.r.t. the first and second moments of the input signal."

[10] For conciseness, the term "w.r.t." abbreviates the preposition "with respect to".

said localized time segment; said accuracy constraint described as "bounded on error with confidence" (i.e., a signal fidelity constraint).

Both are criteria defined within.

Sidebar *"Stationary-Based Encoded Representation"* provides insight into the basic idea behind a stationary-based encoding of an arbitrary signal. As shown in this paper, in addition to capturing valuable inferential knowledge about the underlying random process ***{X}***, our approach results in significant signal compressibility (by removing resulting redundancy on said inferences about localized stability conditions) while enforcing signal fidelity through a rigorously derived autonomously adapted bound over the accumulated quantization error that results from said constant-valued approximation.

Moreover, it is highly desirable for *any* characterization approach to provide insight into the timescale of the observed random process. The approach we present here accomplishes this feat; it uncovers hidden timescales present in sampled observations of a random process. Moreover, it does so in *bounded* time and *known* confidence. Timescale information (partly) manifests itself through the (*random*) duration of stationary-based artifacts (referred to as ***WSS*/ATS segments**) that result from the unearthing of **localized stationary condition**s produced by the **HPS transform**.

Notational Conventions

1. For conciseness, "*w.r.t.*" abbreviates the prepositional phrase "with respect to".
2. Mathematical terms are denoted in ***bold italics***; **HPS** terminology is highlighted in **bold**; qualifying emphasis via *italics*, and key terms highlighted via small caps.
3. Footnotes document peripheral arguments presented for completeness.
4. A random variable (***r.v.***) ***x*** is identified as **‹*x*›**, a random process ***X*** by ***{X}***, and the ***i***-th element in a time series ***y*** by ***y(i)***. For example, the ***i***-th observation in a random variable time series **‹*x*›** will then be **‹*x(i)*›**. For emphasis *w.r.t.* time, a time series named ***y*** is (implicitly) referenced to as **‹*y(i)*›** on left hand side (***LHS***) terms.
5. The prior conditioning ***w*** of an operand ***z*** is specified using a conditional operator │ as in ***z│w***. In particular, at time ***i***, the application of a windowed outlook of size ***m*** over **‹*y*›** is specified as ***[‹y(i)›│m]***. Specifically, such spans the elements y_{i-m}•••$y_{i.}$.[11]
6. Moving window operators are applied over a past outlook of some size ***m*** over a time series. A moving window operator (such as the sampling average **‹*μ[•••]*›**) over a time series **‹*y*›** at time index ***i*** would then be specified as simply **‹*μ[‹y(i)›|m]*›**.[12]
7. A fixed length interval in time is specified as a vector as in . In contrast, a random length interval is specified as a ***r.v.*** bracketed vector as in ‹›.
8. The number of elements (i.e., the size) of a time series ***x*** is given by ║***x***║

Section 1.2: Approach

Although our approach, the **HPS transform**, is palpably well suited for adaptive process control, the results presented have revolutionary implications across many other fields. Next, we preview important concepts about the **HPS transform**. Before introducing the theory of the **HPS transform**, the reader is referred to **Sidebar *"Notational Conventions"*,** which reviews conventions used in this paper.

[11] Formally, the elements of this array are (‹y(i-m)›•••‹y(i)›].

[12] That is, the operator followed by a notation that succinctly describes the moving window operand: time series name (y), the time index (i), and the size of the moving window (m).

Our sampling inversion approach applies *online* statistical filtering over a signal ***‹y(i)›*** (representing a realization of a random process ***{X}***) to identify time segments exhibiting what we refer to as “approximate τ-invariance” – a concept related to the theory of stationariness.[13] We refer to these time segments to as **ATS segments**.

Subsection 1.2.1: Stationary-based Encoding

The approach can be conceptualized as extracting what we refer as to the “fundamental frequencies” of a random process ***{X}***. A fundamental frequency corresponds to a weakly stationary state (referred to as a “process state”) of the sampled random process ***{X}***. For example, measurements from physical sciences, financial sector, earth sciences, manufacturing processes, medicine, etc. while all representing quite different random processes and distributions; all exhibit these segments of **localized stationary condition**s. However, recall that for an arbitrary input signal, the timescale[14] of these weakly stationary states (or “process states”) is unknown. To say the least, the precise number of these “process states” – as well as their duration, distribution, and moments – is anything but unknown. Nonetheless, we show that over time, the approximation produced by the **HPS transform** results in a time series of **ATS segments** that exhibit *small* and *controlled* variability around the (hidden) fundamental frequencies of its underlying random process. That is, without explicit awareness of timescale on the input signal, the time series of these random-length “**ATS segments**” describe a bounded-error controlled variability trajectory over the (true but hidden) “process states” of the underlying random process ***{X}***.[15] It is because of this variability that each such **ATS segment** could be regarded as a “harmonic” overtone/ undertone of the “fundamental frequencies” of ***{X}***. As a result, we refer to the ideas we introduce here as to the theory of the “**Harmonic Process State**” (**HPS) transform**.

Subsection 1.2.2: Error And Stability

More importantly, we show that this decision-making is done optimally in ***O(1)*** worst case computational time (and under modest memory requirements!). In fact, through an approximation equivalency, we reduced a deterministically undecidable online problem of *significant* decision-making complexity into a statistical approximation online problem of *nominal* decision-making complexity – where said approximation is generated under rigorous error management at constant confidence levels.

To do this, the trajectory of each **ATS segment** is constrained by an autonomously adapted error bound proved in this paper to be robust and equivalent (in discrimination power) to interval-based decision-making across said **ATS segment**. To this end, a relaxation factor over the error bound allows control of tradeoffs between the stability of the trajectory[16] exhibited by said approximate process-state tracking segments (i.e., **ATS segments**, if any) and the goodness of fit of said approximation (i.e., the accumulation of quantization error induced by an **ATS segment**-based approximation).

[13] See [GRAY:STATIONARY] for a review of stationary and ergodic random processes.

[14] That is, whether such phenomena manifests across seconds, minutes, tens of minutes, etc.

[15] Giving preliminary insight to this extraordinary result, this is accomplished via continuous speculation of fixed length intervals into a random-length ATS segment.

[16] This random number is referred to as the HPS fractality ‹n› of the HPS approximation, referring to the total number of ATS segments needed to represent the input signal ‹y(i)›.

Our goal is that a significantly small number ***‹n›*** of **ATS segments** characterize the approximations produced by the **HPS transform**, where each meets a bounded property over quantization error. Although *any* transform robustly exhibiting these properties would be highly desirable, until now accomplishing this has been *neither* possible *nor* obvious. We show that the **HPS transform** produces approximations that balance stability of the trajectory while maintaining goodness of fit over the input signal.[17] Moreover, the **HPS transform** is positioned to operate in a stable convex portion of the feasible solution space where *smaller* ***‹n›*** results in same (or increased) accumulated quantization error and conversely, *larger* ***‹n›*** typically result in same (or less) accumulated quantization error.

Subsection 1.2.3: Decision-Making

To address uncertainty in decision-making, the **HPS transform** maps the original decision-making problem into an equivalent decision-making problem where inferences are based on statistics resilient to ill conditioning that may (or may not) be present in the input signal.

Specifically, the "**HPS decision-making**" is based on a class of statistic which is a linear combination of other robust statistics[18] – specifically, windowed operations over sampled moments of the input signal ***‹y(i)›***, which are, themselves, unbiased, consistent, and optimal ***MLE*** random variables.[19] This is done in such a way so that inferences made on this decision-making space have significance when mapped back to decision-making over the original input signal ***‹y(i)›***.

Fig. : Block diagram of the HPS TRANSFORM. The input signal *‹y(i)›* is transformed into three compact signals, the HPS monitor signal *‹mon(i)›*, the HPS error signal *‹ê(i)›* and the HPS outlier signal *‹ô(i)›*.

Subsection 1.3: The HPS Transform

Preliminary, one could abstract the **HPS transform** as a filter that extracts "random-length constant-level components" (if any such is present) from any input signal – even when said segments may be subject to dispersion and/or noise processes. shows the basic block diagram for the **HPS transform**.[20] Note how an input signal ***‹y(i)›*** is transformed into three compact signals as follows:

$$HPS(\langle y(i)\rangle) = \langle\, mon(i)\rangle + \langle \hat{o}(i)\rangle + \langle \hat{e}(i)\rangle. \qquad (1.0)$$

The **HPS monitor signal *‹mon(i)›*** provide a "process state" tracking signal. It contains unearthed random-length **ATS segments** as well as the transitions that connect consecutive pairs of them. depicts how an arbitrary signal (left side) is transformed (when feasible) into a series of **ATS segments** (right side). Each **ATS segment** is represented through a shaded rectangular area, which encapsulates the span and duration as well as the variability of the segment (i.e., the range of values of the input signal contained within said segment). The **HPS transform** uncovers

[17] More precisely, goodness of fit is tightly controlled w.r.t. unbiased tracking estimates of the input signal, which in turn, provide lag-based optimal tracking of the input signal.
[18] See [PICCOLO:ROBUST M-ESTIMATORS] for a review of robust statistics.
[19] Specifically, HPS decision-making operates over linear combinations of robust random variates (e.g., ‹μ[‹y(i)›|m]› and ‹σ[‹y(i)›|m']›) as opposed to w.r.t. the input signal ‹y(i)›.
[20] The HPS transform introduced here is implemented and verified; its reference implementation, referred as to the online HPS monitor, is described later in detail.

these *random-length* **ATS segments** under consistent bounded error and confidence. Yet, this approximate sampling inversion possesses an *optimal* per kernel-iteration worst running time of just ***O(*1*)*** operations (i.e., a constant and small number of elementary operations per kernel-iteration). Moreover, this transformation process is controlled through just a small, intuitive, and simple set of externally visible parameters (related to decision-making and error control), described later on.

The **HPS outlier signal *‹ô(i)›***, robustly generated with low overhead, tracks significant departures in relative magnitude (referred to as outliers) in the original signal. Outliers are tracked *w.r.t.* parameters of average statistical performance. Whereas ***‹mon(i)›*** is useful to *robustly* monitor (finite and stable) variability during observation of a random process (e.g., water level), ***‹ô(i)›*** is useful in *early* detection of significant relative magnitude effects (e.g., flash flood) that unavoidably occur and for which speedily reaction is needed.

By definition, the **HPS error signal *‹ê(i)›*** tracks the *instantaneous* quantization error an stationary-based **approximation** induces *w.r.t.* the input signal. Note that ***‹ê(i)›*** is a dense signal that could be *loosely* approximated as a noise process. In contrast, whereas ***‹mon(i)›*** is a sparse signal with respect to ***‹y(i)›***, ***‹ô(i)›*** is an *extremely* sparse signal. By definition, ***‹ê(i)›*** tracks the *instantaneous* quantization error that the **HPS approximation** incurs *w.r.t.* the input signal. The accuracy of this approximation depends on both the presence of *true* stationary conditions on the input signal as well as on the introspection scheme used by the **HPS transform** to unearth those stationary conditions.

Note that significant amount of the original information content from ***‹y(i)›*** is contained within ***‹mon(i)›*** and ***‹ô(i)›***. More importantly, in an information theoretic sense, ***‹ô(i)›*** has the maximal content of the input signal as it contains events with *extraordinary* low probability of occurance (i.e., heavy tail events). Moreover, ***‹ê(i)›,*** an *approximate* noise process has a *significantly reduced* information content. In contrast, as **ATS segments** in ***‹mon(i)›*** represent *over/under* tones of the fundamental frequencies of the input signal, its information content is somewhere in between these two other signals depending on the likelihood probability of unearthed **ATS segments**.

Next, we proceed with the formulation of the **HPS transform** but first, let us review the structure of the paper. In **Requirements**, we review motivating goals and translate them into requirements. In **HPS Optimization Problem**, we examine the optimization problem that the **HPS problem** seeks to approximate. In **Approach**, we present basic ideas behind the generation of a robust approximation. In **Formulation,** we formally specify the **HPS transform**. Next, we provide illustrations of its use in **Applied Examples**. Then, we examine concerns on **Technical Considerations** and in **Related Work** we consider the relevance of related ideas. Finally, we wrap up in **Conclusion** while theorems are presented in the **Appendix**.

II. Motivating Goals And Requirements

From an end-to-end perspective, a thorough adaptive process control solution must take into consideration important underlying sub-problems, these being shown in .

	Sub-problem	Description
A	sampling collection	Applies a sampling function over the process;
B	sampling analysis	Analyzes samples according to some optimality criteria such as storage, correlation, etc.
C	communication relay	Transfers samples, analyses, and/or decisions;
D	state handling and decision-making	Integrates samples and/or analyses into decision-making about the random process being observed;
E	adaptation and triggering	Implements decisions into the original process and/or triggers other adaptation processes.

Table : END-TO-END perspective of sampling inversion within the context of adaptive process control.

Within the context of adaptive process control, attention has focused on [C, D, E]. That is, [A, B] are typically referred to as pre-conditioning to [C, D, E] while [A, B] are typically handled in *ad-hoc* ways and [C, D, E] simply interface to [A, B] at a *raw* data level. Sampling inversion (which specifically relates to [A, B]) has non-negligible consequences over [C, D, E]; for example, in terms of data volume and robustness of decision-making. This way, note that the above-described decomposition does *not* involve *explicit* constraints over [A, B]. Therefore, our approach is to explore *approximation* tradeoffs in [A, B] that can lead to significant reduction in the complexity experienced on [C, D, E] leading to adaptive process control applications that exploit benefits from sampling inversion.

Section 2.2: Motivating Example

Now, consider the adaptive process control scenario – taken from [**NRM_USPTO99**][21] – depicted in **Fig. 3**. It shows a distributed resource ***R*** at a client ***C*** subject to remote decision-making at a server ***S***. This decomposition helps us understand further our end-to-end approach to sampling inversion – as implied in **Fig. 3**. In general, it is desirable that communication requirements imposed by a solution approach be *small* (*w.r.t.* [C]) as long as decision-making be *robust* (*w.r.t.* [D]) while preserving the overall goal of a *stable* (adaptive) process control (*w.r.t.* [E]).

Fig. 3: Motivating example. The structure of a distributed, loosely coupled, adaptive process control. Client *C* performs monitoring of a local resource *R* with sampling effort ‹*y(i)*› but reports to server *S only* on changes over a state memory ‹*π(k)*›, thus exhibiting a reduction property ‹*k*› ‹‹ ‹*i*›

By cross-referencing **Fig. 3** *w.r.t.* , the description of the adaptive process control model is enhanced with signal characterization as follows.

1. Samples about ***R*** (*w.r.t.* [A]) are collected *w.r.t.* some underlying random phenomena[22] ***{X}***.
2. ***C*** becomes a smart client capable of performing analyses, inferences, etc. (*w.r.t.* [B]) about ***{X}*** rather than being solely a data collection point.
3. ***C*** communicates with server ***S*** (or perhaps other clients) in terms of *inferential* knowledge (*w.r.t.* [C]).

[21] See [NRM_USPTO99] for an extensive description of this scenario and its possibilities in distributed, loosely coupled adaptive resource management.

[22] Examples of relevant random phenomena are network delay, network bandwidth, fluid pressure, weather disturbances, tectonic plate movement, earthquake vibrations, signal stenography, population growth, financial share price, object tracking, etc. where tracking of temporally-stable changes in variability w.r.t. a shifting baseline is highly desirable.

4. ***S*** then incorporates such into its decision-making (*w.r.t.* [D]) – in this case, being adaptive resource management over ***R***.
5. Decision-making may require adjusting compensation measures, which then (*w.r.t.* [E]) could be done at either ***S*** or at ***C*.**

Note that the model described in is generic to any form of distributed management of resources. However, the **HPS transform** is *not* limited to adaptive control applications; its relevance extends to domains that benefit from awareness of a stationary-based approximation to its signal stimuli. Moreover, because the magnitude of the data induced by a stationary-based approximation can be quite significant, applications derive scalability benefits in [C, D, E].

Section 2.3: Requirements For Feasible Approximations

The resulting *stationary-based* approximation produced by the **HPS transform** is referred to as the **HPS approximation**. When true (weakly or not) stationary conditions are present within the input signal, these long-term conditions are referred to as process states of the input signal. As **ATS segments** are unearthed, the collection of **ATS segments** uncovers the underlying process states of the input signal. Conceptually, one could conceive **ATS segments** as sampling the hidden process states of the input signal. Because of this uncertainty, **ATS segments** are referred to as *approximate* process states. The optimal number of **ATS segments** is equal to the number of (true but hidden) process states.

However, for an **HPS approximation** to be *feasible* is just necessary that the number ***‹n›*** of uncovered **ATS segments** be "*sufficiently small*" so that it meets the reduction property ‹***k***›‹‹***i***› enunciated in . This way, an **HPS approximation** is by definition desensitized to noise present within the input signal. However, for an **HPS approximation** to be a solution, this desensitization has to be accurate and robust. Therefore, it is necessary that **ATS segments** also minimize the **accumulation of quantization error** they induce over the input signal. An **HPS approximation** that satisfies the above is a feasible solution. Such **HPS approximation** greatly simplifies and makes more robust pre-conditioning to [C, D, E] (but in particular, decision-making) while increases signal compressibility and maintains suitable signal fidelity.

Section 2.4: Requirements For The HPS Transform

However, if adaptive process control is to be driven by a stationary-based approximation to an input signal, it is desirable that its generating function (that is, the **HPS transform**) exhibits desirable properties. These desirable properties are summarized as below.

1. Bounded performance[23] requires that **HPS approximations** possess consistent precision and accuracy. Moreover, for *online* algorithms, this also requires that **HPS approximations** be generated within feasible computational time.
2. Stability[24] requires that the operation of the **HPS transform** be in a "stable optimality region" that behaves in such a way that *small* variation on input parameters results in *small* variation in

[23] See [REF:COMPLEXITY] for a review of the algorithmic complexity of algorithms.
[24] See [REF:NUMERICAL STABILITY] for a review of stability in numerical analysis.

output qualities associated with the **HPS approximation**.

3. Robustness[25] requires that ill-behaved input (such as departures from normality) be handled in such a way that the above properties continue to be met.

We show that the **HPS transform** addresses *all* these requirements in a *robust* and *efficient* manner, delivering a class of feasible stationary-based approximations for any input signal **‹*y(i)*›**.

III. Optimization Problem

The **HPS transform** is *not* limited to the realm of stationary signals; it definitively applies to non-stationary signals. The **HPS transform** allows extracting from non-stationary signals, when possible, the presence of **localized stationary condition**s. This is *not* a discrepancy; bursts of **localized stationary condition**s are found across signals said to be non-stationary across long-term horizons. Nonetheless, for clarity, we consider next the case of a signal with **‹*n*›** process states.

The **HPS problem** is an optimization problem *w.r.t.* goodness of fit and stability of the trajectory of **HPS approximations.** First, a metric ‹›, referred to as the **HPS fractality** of the **HPS approximation**, tracks the total number of **ATS segments** needed to represent a stationary-based approximation of the input signal. This way, **HPS fractality** represents a measurement of the stability of the trajectory of the **HPS approximation**. Second, an ***MSE*** metric is used to track a *form*[26] of the total **HPS quantization error** induced by an **HPS approximation**. This way, this ***MSE*** metric represents a measurement of the goodness of fit of the **HPS approximation.**

However, an “interaction effect” exists between goodness of fit and stability of the trajectory. One need only realize that the **HPS transform** generates a stationary-based approximation that consists of **‹› ATS segments** where each **ATS segment** represents an *approximate* process state $\boldsymbol{\{\varphi_{k,j}\ (\mu_{k,j},\ \sigma_{k,j})\}}$ and where along each such, accumulation of **HPS quantization error** is bounded *w.r.t.* an error conditioning goal (referred to as the **HPS error bound**). This way, the *tighter* the **HPS error bound** is made to be (consequently resulting in *higher* goodness of fit), the *more* **ATS segments** are likely to be generated (and consequently, resulting in *lower* stability of the trajectory) for such **HPS approximation**. This results in a convex region containment of *feasible* and *optimal* solutions. As a fact, inside this region, *higher* stability of the trajectory implies *(same or) lower* goodness of fit while *lower* stability of the trajectory implies *(same or) higher* goodness of fit.

The **HPS transform** incurs in an additional source of error. Whilst the above error source relates to the aptness of a stationary-based approximation model *w.r.t.* an arbitrary input signal; a second error source is also present, this being the inherent *targeting* error incurred in estimating a baseline value for each **ATS segment**. Nonetheless, this secondary error source *also* exhibits the aforementioned “interaction effect”. To see this assume a signal has **‹*n*›** *true* process states and the **HPS transform** generates an **HPS**

[25] See [REF:ROBUSTNESS] for a review of robustness in statistical analysis.

[26] A carefully constructed quadratic error form is used to closely track the behavior of error across both tracking signals as formulated elsewhere within the paper.

approximation for it consisting of ‹› **ATS segments** (where ‹› ≥ ‹***n***›**)**. Now, the *greater* the number ‹› of **ATS segment**s unearthed from the signal, then the *smaller* that the induced **HPS quantization error** would be. This is due to the sampling distribution of baseline estimates (for unearthed **ATS segments)** *w.r.t.* the *true* baseline value (of their underlying process state). As a result, some of these will be slightly *off target* yet as more **ATS segments** are unearthed and used to represent a *true* process state, the more likely they would (as a set) *sample* the *true* baseline value of the corresponding process state.[27] This way, the trajectory of such an **HPS approximation** represents an unbiased sampling process exhibiting controlled – referred to as *harmonic* – variability centered on the underlying (if any) **HPS fundamental frequencies** of *any* signal. This (**HPS Bounded Trajectory Theorem**) result is formalized on the **Appendix**. For now, it suffices to recall that the resulting **HPS quantization error** ($\langle \mu_k \rangle - \langle \mu_k(i) \rangle$) is a *well-behaved* error form, a result of being a linear combination of two robust estimators.

Clearly, the *optimal* solution to the **HPS problem** is – of course – just the **‹*n*›** *true* process state**s** $\{\varphi_k\ (\mu_k, \sigma_k)\}$, each having its respective $\langle \mu_k \rangle$ (that is, its corresponding **HPS fundamental frequency**) as its estimated *baseline* value. However, there is a tradeoff; we seek to obtain a *small* enough number ‹› of **ATS segment**s (where ‹› ≥ ‹***n***›) as long as such ‹› also results in *small* enough accumulation (of a certain form) of **HPS quantization error**. The **HPS transform** generates such a representative instance, a particularly well-conditioned **HPS approximation** constructed to represent an *approximate* solution within the convex region spanned by feasible tradeoffs of goodness of fit and stability of the trajectory. The choice of this representative instance is important; to this end, **Fig. 4** provides intuition about its selection. The **HPS transform** focuses into a *small* region (within the feasible convex area) constrained *apriori* by the selection of input parameters. We show that for fixed confidence level ***p*** (used in **HPS decision-making**), the resultant **HPS approximation** is determined solely by the CLT stabilization orders (***m*** and ***m'***) used. Yet, due to the probabilistic nature of **HPS decision-making**, resultant **HPS approximations** produced by the **HPS transform** exhibit the observed *small* range of tradeoffs in goodness of fit and stability of the trajectory.

Fig. 4: Intuition into the operation region of the HPS Transform.

Fig. 4 depicts the classic operating curve for continuous approximations – that is, more degrees of freedom, less error (and conversely) – but this time augmented with intuition about the operating curve of the **HPS transform** by depicting operating contours for various confidence levels. For illustration purposes, **Fig. 4** shows this *apriori* confidence value ***p*** as a (blue) dot inside a (grey) box that represents said *posteriori* operating region and from which orthogonal projections (shown as grey rectangles) to the spanning axis depict a mapping to the resulting range of values in goodness of fit and stability of the trajectory. Specifically, our **MSE Equivalency Theorem** provides with a way to map an *apriori* confidence value ***p*** into a *posteriori* error bound. This way, we are

[27] Intuitively, there are some ‹›/‹n› chances of estimating the true baseline value with each successive pick being more likely to be on target than any previous as the estimation process is preconditioned, as time goes by, by increasing memory of observed stationary conditions.

now able to enforce consistently – during the construction of every **ATS segment** – an expected maximal error bound over **HPS quantization error**. Resultant **HPS approximations** exhibit error behavior that is consistent to assumed error probability.

For comparison purposes, we now consider the *offline* case – recalling that there is *no* deterministic solution for the online case. We show a ***divide-and-conquer*** optimization (as presented in the ***Sidebar "An Offline Solution To The HPS Problem"***) that exhibits ***O(N log N)*** worst time complexity when coupled with an ***O(1)*** merge process.

An *offline* solution to the HPS problem

Given a discrete signal ***g(i)***, we want to identify time segments exhibiting wide sense stationary ***(WSS)*** conditions regardless of timescale. Let ***k*** be the level (where ***0 ≤ k ≤ log N***), ***N*** be the size of the input (padded, if necessary, to a multiple of ***2***), and ***WSS^k*** be a ***WSS*** segment at the ***kth*** level. Let any ***WSS*** segment be represented by a tuple ***(‹l_{low}›, ‹l_{high}›, μ_l)*** composed of its end-points ***(‹l_{low}›, ‹l_{high}›)*** and a representative value (***μ_l***) for its ***WSS*** condition.[28] Finally, let ***{WSS^k}*** (referred to as the *segment-set*) be the set of segments produced by a merge process belonging to the ***kth*** level.[29] For any level ***kth***, its *segment-set* solution is defined to be the set of ***m*** (where ***1 ≤ m ≤ N/2^k***) ***WSS*** segments spanning (without overlap) the totality of the underlying sub-intervals spanned by the *segment-set* solutions from the ***kth-1*** level. For argumentation consistency, when no ***WSS*** conditions are present in a given interval, each discrete point ***i*** of the interval represents a *trivial* ***WSS*** segment ***(i, i, g(i))***.

Fig. 5 illustrates this ***divide-and-conquer*** scheme for a small series of just ***8*** samples and thus, ***k=3*** and ***k+1=4*** levels. It also illustrates ***three*** instances of the merge process (corresponding to ***k=2*** and ***3***) together with time indexes for (relevant endpoints) of merge operands. For example, at the ***k=3*** level, a ***WSS$^{k=2}$*** segment (ending at index ***4***) undergoes a merge process with another ***WSS$^{k=2}$*** segment (starting at index ***5***). Assume there exists a merge function ***WSS^k(low, high)***, which at the ***kth*** level produces a *segment-set* solution ***{WSS^k}*** by merging its two underlying *segment-set* solutions ***{WSS^{k-1}}*** (that is, those from the ***k-1th*** level)**.** To do so, it (*somehow*) tells whether th0se ***WSS*** segments are *similar* – that is, both represent a spanning of the same underlying ***WSS*** condition. Because of transitivity, (it can be shown that) the ***kth*** level merge process need only focus on the inner ***WSS^{k-1}*** segments (that is, from the ***k-1th*** level) being merged. For this purpose, assume a function of ***SIMILAR(WSS$_L$,WSS$_R$)*** exists, which determines whether two ***WSS^{k-1}*** segments referred to as ***WSS$_L$*** and ***WSS$_R$*** are similar. For convenience, we refer to the middle point of any ***kth*** level interval being merged to as the *pivot point* ***p*** of ***{WSS^k}***. This way, the ***WSS^{k-1}*** segment (that is, from the ***k-1th*** level, whether *trivial* or not) to the right of this point is referred to as ***WSS$_R$*** and to the ***WSS^{k-1}*** segment (whether *trivial* or not) on its left to as ***WSS$_L$***. This way, at the ***kth*** level, the merge function ***WSS^k(low, high)*** simply focuses on computing ***SIMILAR (WSS$_L$,WSS$_R$)***. Specifically, if the representative values of those inner ***WSS^{k-1}*** segments are similar, then, at the ***kth*** level, they could be merged into one ***WSS^k*** segment. By induction, the following steps generate the *segment-set* ***{WSS^{logN+1}}***. At the ***1th*** level, ***WSS^1(i-1,i)*** examines the values of two *trivial* ***WSS*** segments, ***g(i)*** and ***g(i-1),*** and (*somehow*) it tells whether these *trivial* ***WSS*** segments are *similar*. At each level, this induction setup is repeated. As stated, the ***kth*** level, ***WSS^k(low, high)*** produces a *segment-set* solution ***{WSS^k}*** by merging its two underlying *segment-set* solutions ***{WSS^{k-1}}*** (that is, those from the ***k-1th*** level)**.** Finally, at the top, ***WSS^{logN+1}(1,N)*** merges together the last two *segment-set* sub-problems (each of size ***N/2***). This algorithm unearths *all* ***WSS*** segments (by transitivity, of *any* length) found within the discrete signal ***g(i)***.

Now we focus on the presumed merge function ***WSS^k(low, high).*** To move from any ***k-1th*** level to the ***kth*** level, such merge process must consider ***four*** distinct cases; these being represented as cases ***a*** through ***d*** in **Fig. 6**. At both ***k-1th*** and ***kth*** levels, a *light grey* rectangles depicts a *trivial* ***WSS*** segment whereas a *blue* rectangle depicts a non-*trivial* ***WSS*** segment. Merged ***WSS^{k-1}*** segments at the ***kth*** level are shown in *black*, with the resulting ***WSS^k*** segment shown in dark color (that is, as a combination of *blue* and/or *black*). These cases correspond to whether (or not) the merge operands (***WSS$_L$*** and ***WSS$_R$***) are *trivial* ***WSS^{k-1}*** segments. This way, case ***a*** represents the merging of two *trivial* ***WSS^{k-1}*** segments; cases ***b*** and ***c*** illustrate each the merging of a *trivial* ***WSS^{k-1}*** segment with a (non-trivial) ***WSS^{k-1}*** segment; and finally, case ***d*** illustrates the merging of two non-*trivial* ***WSS^{k-1}*** segments. **Fig. 6** also shows that, for each such case, ***SIMILAR(WSS_L,WSS_R)*** has two possible outcomes; either the merge operands (that is, ***WSS_L*** and ***WSS_R***) are *similar* (depicted atop the right of each case) or they are *not* (depicted atop the left of each case). This way, the left side of each case depicts a (***k^{th}*** level) merge solution that leaves intact the *segment-set* solutions of the ***k-1^{th}*** level. In contrast, the right side of each case depicts a (***k^{th}*** level) merge solution that reworks ***WSS_L*** and ***WSS_R*** segments (from the *segment-set* solutions of the ***k-1^{th}***) into some new ***WSS^k*** segment, this now being optimal at the ***k^{th}*** level. This way, when at the ***k^{th}*** level both ***WSS_L*** and ***WSS_R*** are *similar*, case ***a*** produces a new non-*trivial* ***WSS^k*** segment by merging two *trivial* ones; case ***b*** and ***c*** produce (each) an extended ***WSS^k*** segment by merging a *trivial* ***WSS^{k-1}*** segment with a non-*trivial* one; and case ***d*** produces an extended ***WSS^k*** segment by merging two non-*trivial* ones. The function ***SIMILAR(WSS_L,WSS_R)*** determines whether two ***WSS^{k-1}*** segments are *similar*; its outputs are a ***yes/no*** answer, and when *similar*, a representative value for the resultant ***WSS^k*** segment together with its endpoints. This is done simple by *somehow* comparing corresponding representative values ***μ_L*** and ***μ_R***.. Since at the ***k^{th}*** level, each such value has been pre-computed taken at the ***k-1^{th}*** level, this results in an ***O(1)*** computational complexity at each such merge, which in turn results in an overall ***O(N log N)*** computational complexity for the entire ***divide-and-conquer*** optimization.

The sketched algorithm is correct – it extracts from an arbitrary ***g(i)***, the {***WSS^{NlogN}***} ***WSS*** segments found within, without prior awareness (and regardless) of both the number of such segments present within and the timescale of localized stationary conditions. Nevertheless, its robustness depends on 1) the robustness of ***SIMILAR(WSS_L,WSS_R)*** in determining the similarity of representative values and 2) on the meaningfulness of the value chosen to represent a ***WSS*** segment.

Fig. 5: *Divide-and-conquer* setup for the *offline* HPS problem.

Fig. 6: Possibilities at the *pivot point* of the *k^{th}* level merge process.

This *offline* ***divide-and-conquer*** algorithm generates, in ***O(N log N)*** time, the optimal ***‹n› WSS*** segments – if any exists for ***‹g(i)›***. It generates *true* ***WSS*** segments and because of this, it need not address error; *no* tradeoffs are made. Nevertheless, given inherent variability and statistical outliers, one would prefer (for robustness reasons) to determine whether the two ***WSS*** segments being compared are *statistically similar*. This tradeoff results in an approximation problem, for which error need be managed. The *online* **HPS transform** represents an *online* variant of this. It generates an *approximate* {***WSS***} *segment-set* solution consisting of **‹› ATS segments** (where **‹› ≥ *‹n›***) – in optimal (*offline*) ***O(n)*** time yet under bounded error and with consistent confidence.

IV. HPS Conditioning

Next, we review some basic conditioning tricks needed before we discuss the *online* generation of out **stationary-based approximations.**

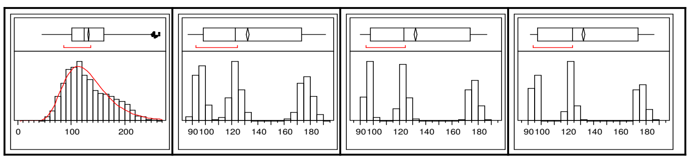
Fig. 7: Detailed block diagram for the basic HPS decision-making element.

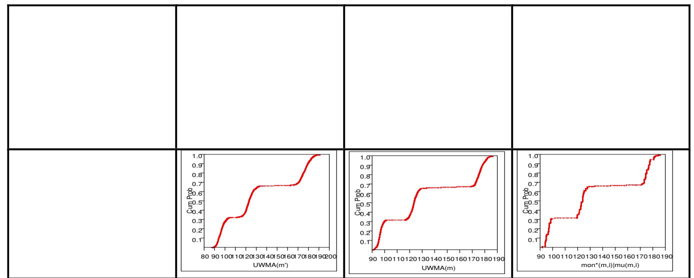
Section 4.1: Super-Heterodyning

Fig. 7 shows a block diagram for the basic decision-making element of the **HPS transform**. The input signal ***‹y(i)›*** is split into *two* signals,

1. the *original* input signal ***‹y(i)›*** and
2. a *time-delayed* version of the input signal, ***‹y(i-τ)›***,

where ***τ*** represents said delay.[30] Regardless of choice of ***τ***, the idea of self-generating a reference signal from an input signal in order to *significantly* increase discrimination power achievable in decision-making is *somewhat* analogous to the revolutionary concept of the super-heterodyne (***SHD***) receiver.[31] We refine this *langsyne* concept to open possibilities in decision-making, which we then apply toward statistical signal processing (i.e., the **HPS transform**). Specifically, filtering of a signal into its stationary-based approximation is accomplished via *sequential* decision-making. In order to establish a sequential decision-making model, the input signal and its time-delayed signal (***‹y(i)›***, ***‹y(i-τ)›***) are both smoothed into (*specially constructed*) robust indicators on which to root inferences about localized stationary conditions on the input signal ***‹y(i)›***. We refer to this decision-making process as to **HPS decision-making**.





(a)	(b)	(c)	(d)

Fig. 8: Histograms (*pdf*, top) and (*cdf*, below) for: (a) input signal, (b) fast signal, (c) slow signal, and (d) HPS approximation. Note the effect of smoothing.

Section 4.2: CLT-Stabilization

Careful selection of smoothing (and smoothing degree) reduces variability inherent within **localized stationary condition**s. For an input signal, careful smoothing may cause multimodal distributions to emerge on the corresponding ***pdf*** plot. For example, **Fig. 8** shows such case through ***pdf*** and ***cdf*** plots for an input signal and three smoothed versions of the signal referred to as the **HPS fast signal**, the **HPS slow signal**, and the **HPS approximation**. Smoothing is achieved by the humble ***UWMA***-class smoothers (*Uniformly Weighted Moving Average*, i.e., -values).

Although their construction is simple and straightforward, ***UWMA***-class smoothers are robust in areas of critical concerns. The robustness of these is rooted on the interaction of three central arguments: the law of large numbers[32], the central limit theorem, and the curtailment of influence and correlation functions. For conciseness and clarity, the elaboration of these arguments is provided later. For now, suffices

[30] It can be argued that, one would prefer to substitute the constant delay τ with an adaptive delay ‹τ(i)› (possibly adapted w.r.t. some constraint) or otherwise randomized delay (i.e., ‹τ› centered around it). These, and similar issues, are investigated later in Parameters.

[31] The WordWeb 3.01 Dictionary [REF:WORDWEB] defines super-heterodyne receiver as "A radio receiver that combines a locally generated frequency with the carrier frequency to produce a supersonic signal that is demodulated and amplified." Note that this (still in use) concept, introduced c. 1917 by [ARMSTRONG:SHD], was intended for enhancing the reception of weak radio signals. This was achieved through non-smoothed differencing of the SHD signals. Moreover, the SHD-generated reference signal represented a known "prior" – something not present here. This is somewhat analogous yet significantly different than our approach.

[32] See [GRAY:LARGE NUMBERS LAW] for a review of the law of large numbers.

to note that thanks to the law of large numbers, depending on the choice of the smoothing function, smoothing can be used to approximate sampling averages that are unbiased, precise, and consistent estimators of the mean of the original input signal ***‹y(i)›***. Then, owing to the central limit theorem[33], hypothesis testing indicators derived from properly scaled ***UWMA*** sampling averages are *approximately* normal[34].

Section 4.3: Robustness of Clt-Stabilization

The *optimal* smoothing degree ***m*** to achieve *robust* CLT-based stabilization of an *arbitrary* signal ***‹y(i)›*** is a time-varying number ***‹m(i)›***, which varies according to both variability and outlier conditions. However, particular threshold values of ***m*** exist (e.g., ***m › 30***) at which most (useful) underlying distributions of ***‹y(i)›***-values robustly converge (or *already* converged) into (*approximately*) normal probability distributions [**GRAY:CLT-ORDER**]. For convenience, we refer to the application of proper smoothing conditioning over a signal ***‹y(i)›*** to as the generation of its "CLT-stabilized signal of order ***m*** for ***‹y(i)›***". Based on our notational conventions, this is represented by the random variable ***‹μ[‹y(i)›|m]›***. Succinctly, ***‹μ[‹y(i)›|m]›*** specifies the application of an ***UWMA(m)*** smoothing filter over ***‹y(i)›***.

signal	*outlook*	*indicator*	*clt*	*sub-interval*
HPS fast signal	***present***	***μ[‹y(i)›|m′]›***	***UWMA (m′)***	***((i-τ)••• i]***
HPS slow signal	***recent past***	***‹μ[‹y(i-τ)›|m]›***	***UWMA(m)***	***(i-(τ+m)••• (i-τ)]***

Table 3: The super-heterodyned signal set: HPS fast signal and HPS slow signal.

It could be argued that one would prefer to substitute an ***UWMA(m)*** smoother for some faster (and more accurate) smoother – such as the often used ***EWMA (m*)***, since (***m* ‹‹ m***) or even an ***ARIMA(p,q,d)*** filter.[35] Unfortunately, such would be counterproductive here; as such techniques would hinder the (CLT-based) induced robustness over **HPS decision-making**. In actual fact, this induced robustness is achieved only through the "less efficient and accurate" ***UWMA(m)*** -based CLT-stabilization of ***‹y(i)›***. Therefore, for robustness, the input signal is converted into a CLT-stabilized signal of some order ***m′*** while its *time-delayed* version is converted into a CLT-stabilized signal of some other order ***m***. The CLT-stabilized signals are referred to as the **HPS fast signal** and the **HPS slow signal**, respectively. These names are derived from the fact that each sample of the **HPS fast signal** provides an ***UWMA(m′)*** outlook to the present of the input signal ***‹y(i)›*** whereas the **HPS slow signal** provides a ***τ***-delayed ***UWMA(m)*** outlook to its recent past (see Table 3).

Section 4.4: Approximate Tao-Invariance

We wish to determine whether localized stationary conditions span an arbitrary interval. To this end, we

[33] See [GRAY:CLT] for a review of the central limit theorem with respect to sampling averages. Strong central limit theorems expand the conditions on which the basic central limit theorem applies to, increasing the robustness of this fundamental result. See [REF:STRONG-CLT] for a review of these strong central limit theorems. Of particular relevance are results related to dependency between the samples [REF:SCLT-CORRELATION], uneven spacing between samples [REF:SCLT-SPACING], distributions of the samples [REF:SCLT-DISTRIBUTIONS], and impact of heavy-tails [REF:SCLT-HEAVYTAILS].

[34] Actually, they follow a student t distribution; provided a sufficiently large smoothing degree was used and resulting variance was bounded (i.e., finite σ2). Note that this "bounded variance across the sampled finite interval" requirement is almost always met. A violation to finite variance across the smoothing interval applied over ‹y(i)› will be an ±∞ discontinuity in ‹y(i)›, such as y(x)=tan(x). However, this case, after an outlier detection delay, would be detected as a supra-ordinary statistically significant outlier.

[35] See [MONTGOMERY:EWMA-SPC] and [REF:APPLIED-ARIMA, ROSS:TIME SERIES CHAPTER] for a review of UWMA and ARIMA filters respectively.

promote the concept of **τ**-invariance [**GRAY:ERG**] and then lead to a surprisingly robust and efficient *approximate* test. Let ***‹g(‹y(i)›)›*** be a property (e.g., ***mean, variance, kurtosis***) associated with signal ***‹y (i)›***. Informally, **τ**-invariance *w.r.t.* a property requires that the property in question remains "unaffected" over time (i.e., *time-invariant*). However, for our purposes, we are only interested on its *approximate* behavior across a finite time interval.[36] To this end, we formalize the concept of **approximate τ-invariance** to be finite and stable mean and variance across a finite interval of ***‹y(i)›***. This implies that for all ***i*** in , for all **τ** as long as ***i–τ*** remains in , there exists small ***ε*** such that:

and . (3.0)

We show that with the use of the **HPS fast signal** and **HPS slow signal**, it is possible to robustly and efficiently test for the *approximate* presence of **localized stationary condition**s across a finite interval.

V. Approach

The **HPS transform** uses stationary-based encoding to generate an **HPS approximation**. Therefore, the goal of the **HPS transform** is to estimate the presence (and representative value) of each **localized stationary condition** found within the input signal. In **Article III: Optimization**, we specified an *offline* algorithm, which optimally accomplished this. Now we focus on an *online* version. The basic idea for the generation of a stationary-based approximation is similar; *online* **HPS decision-making** tries to find and track the presence of **localized stationary condition**s (this time) via **ATS segments** and then, between any consecutive pair of (*non-trivial*) **ATS segments**, it generates fine-grain (sample-by-sample) non-constrained tracking.

We start first by presenting the **random process model.** Then, we review ideas behind our *online* **HPS transform**, which we refer to as the **online HPS monitor**. These relate to decision-making and error-control. However, before going further, the reader ask to keep in mind the **Sidebar *"Definition and Inter-Relation of Key Terms"*** for reference and insight into terms central to the elaboration.

Definition And Inter-Relation Of Key Terms	
STATISTICAL CONCEPT	
localized stationary condition	***wide sense stationariness***[37] *across finite interval*
DISCRETE OPTIMAL	
process state	***localized stationary condition*** *of significant timescale*
WSS segment	*representation of* ***localized stationary condition ‹›***
HPS fundamental frequency	*first moment of a* ***localized stationary condition***
DISCRETE APPROXIMATION	
approximate τ-invariance	*approximate* ***localized stationary condition*** *across finite interval exhibiting finite and stable μ and σ²*
ATS segment	*estimate (based on* ***approximate τ-invariance****) of ‹›*
representative value	*estimate (based on* ***approximate τ-invariance****) of* ***HPS fundamental frequency***

Section 4.1: Random Process Model

A random process ***{X}*** is modeled here as a composite

[36] It is acknowledged that stationary properties are not usually discussed w.r.t. finite intervals, for this reason we refer to this treatment as to "approximate τ-invariance".

[37] See [REF:WSS] for a review of wide-sense stationary.

function[38] of non-overlapping[39] ***On/Off*** random sources $\langle \varphi_k(i) \rangle$ (of unknown moments, distributions and durations) as follows:

$$\{X\}=\Sigma_k \Sigma_i \langle \varphi_k(i) \rangle, \quad (5.0)$$

where each random source $\langle \varphi_k(i) \rangle$ may have unknown duration, moments, and distributions.[40] Let $\langle \mu_k(i) \rangle$ be a sampling mean of any such random source $\langle \varphi_k(i) \rangle$. By subtracting $\langle \mu_k(i) \rangle$ from its respective $\langle \varphi_k(i) \rangle$, we obtain a residual $\langle \eta_k(i) \rangle$:

$$\langle \eta_k(i) \rangle = \langle \varphi_k(i) \rangle - \langle \mu_k(i) \rangle). \quad (5.1)$$

The above let us rewrite (5.0) to describe the random process $\{X\}$ as:

$$\{X\}=\Sigma_k \Sigma_i \; (\langle \mu_k(i) \rangle + \langle \eta_k(i) \rangle). \quad (5.2)$$

Our interest is unearthing the presence of **localized stationary condition**s. This has important consequences; (to see this) consider that under the presence of stationary random sources, (5.0) reduces to

$$\{X\}=\Sigma_k \; (\langle \mu_k \rangle + \Sigma_i \langle \eta_k(i) \rangle). \quad (5.3)$$

However, our interest is unorthodox; we mine for finite bursts of what we refer to as "**approximate τ-invariance**". Each such burst is modeled as a **localized stationary condition**. Let ‹› be one such **localized stationary condition**. By definition, it can be described by a mean $\langle \mu_k \rangle$ modified by a dispersion $\langle \eta_k(i) \rangle$. This way, for convenience, if the **localized stationary condition** ‹› is of significant duration, we refer to such as a **process state** (of $\{X\}$) and denote it by $\{\varphi_k(\mu_k, \sigma_k)\}$. For convenience, we also define a *trivial* process state to be a **localized stationary condition** ‹› of size 1. Similarly, in the domain of $\langle y(i) \rangle$, a process state is modeled by a ***WSS*** segment and hence, a *trivial* process state by a *trivial* ***WSS*** segment.

We define the **HPS fundamental frequency** $\langle f_0(\langle \varphi_k \rangle) \rangle$ of a process state to be the *baseline* (i.e., constant level) of the process state –being equal to the *constant* estimator value that minimizes error across the process state $\{\varphi_k(\mu_k, \sigma_k)\}$.[41] This value is estimated by the ***mean*** of $\{\varphi_k(\mu_k, \sigma_k)\}$ or simply, $\langle \mu_k \rangle$. More generally, $\langle f(\langle\rangle) \rangle$ is the *optimal* representative value of ‹›.

Example: On the realization domain of $\{X\}$, given a random segment ‹› of duration |‹›| spanning the interval $[\langle u_{low} \rangle, \langle u_{high} \rangle)$ of the order ***m*** CLT-stabilized ***r.v.*** $\langle \mu[\langle y(i) \rangle | m] \rangle$ of a realization $\langle y(i) \rangle$, we estimate:

. **(5.4)**

This ***grand mean*** of $\langle y(i) \rangle$ on ‹› is an estimate of the *baseline* of ‹›.

Rewriting (5.3) in terms of **HPS fundamental frequencies,** we get:

$$\{X\} = \Sigma_k \langle f(\langle\rangle) \rangle + \Sigma_k \Sigma_i \langle \eta_k(i) \rangle. \quad (5.5)$$

That is, the **HPS transform** models the random process $\{X\}$ as a (time series) sum of **HPS fundamental frequencies** hidden by a variability backdrop. There are three general cases to consider in advancing this (localized stationary conditioned) model.

1. If a random source $\langle \varphi_k(i) \rangle$ (say of length ***r***) is indeed stationary[42], then $\langle \mu_k(i) \rangle \rightarrow \langle \mu_k \rangle$ and $\langle \eta_k(i) \rangle \rightarrow \langle \sigma_k \rangle$. Then, $\langle \varphi_k(i) \rangle$ is simply modeled as

[38] ***This intuitive look at a random process relates to the "ergodic decomposition theorem" [GRAY:ERGDECOMP]. Informally, the theorem states that "under quite general assumptions, any non-ergodic stationary process is in fact a mixture of stationary and ergodic processes, just that you don't know in advance which one [GRAY:ERGODIC]."***

[39] Without loss of generality; overlapping random sources can be modeled as shown in case (2).

[40] See [GRAY OR REF:MIXTUREPROCESS] for a review of mixture random processes.

[41] Note that this definition requires knowledge of future ‹y(i)›-values.

[42] See [GRAY:ERGODICITY, REF:STATIONARITY] for a review of stationarity.

generating just a single process state $\{\varphi_k(\mu_k,\sigma_k)\}$.

2. However, if a random source $\langle\varphi_k(i)\rangle$ manifests stationarity by parts, then we recursively apply (5.0). Let q be the total number of localized stationarity conditions found within. Then, the random source $\langle\varphi_k(i)\rangle$ is now modeled by a set of process states $\{\varphi_{k,j}(\mu_{k,j},\sigma_{k,j})\}$ for all $j\in q$, being interspersed across the remainder of just (r-∑|$\{\varphi_{k,j}(\ldots)\}$|) *trivial* process states.
3. Finally, if the random source $\langle\varphi_k(i)\rangle$ is non-stationary across *all* finite intervals within, it is now modeled as r *trivial* process states $\{\varphi_{k,j}(\mu_{k,j},\sigma_{k,j})\}$ – for $j\in r$.

The stationary-based encoding generalized above exhibits significant **signal compressibility** potential for arbitrary realizations $\langle y(i)\rangle$ of $\{X\}$. In contrast, **signal fidelity** depends on how accurately it is possible for an *online* algorithm to determine both **(a)** the presence and location of localized stationary conditions and **(b)** the *optimal* representative value for each such.

Section 5.2: HPS Decision-Making

To address the first problem, it is necessary to address two decidability concerns: *first*, a way to detect the presence of a **localized stationary condition** is needed and *second*, given a **localized stationary condition**, it is necessary to determine whether the next observation belongs (or not) to it.

As stated, in **Optimization** we specified an *offline* algorithm, which accomplished both these optimally and therefore resulted in an (**HPS irreducible**) overall error of $\Sigma_k\Sigma_i\langle\eta_k(i)\rangle$ specified on (5.15). In contrast, it can be shown that for any *online* algorithm the "membership problem" is equivalent to the problem of *exactly* identifying the beginning $\langle_{low}\rangle$ and end $\langle_{high}\rangle$ of any **localized stationary condition**s ‹› – a matter that is ***undecidable*** for any *online* algorithm (that is, lacking knowledge of future $\langle y(i)\rangle$ values).[43] However, we show here that it is possible for an *online* algorithm to infer – with statistical confidence (under bounded error and limiting probability) – the *approximate* presence and duration of **localized stationary condition**s ‹› found within a realization $\langle y(i)\rangle$ of $\{X\}$.

An *online* solution to the membership problem is needed. To this end, one may recall the $similar(w_l,w_r)$ function from the *offline* algorithm presented in **Optimization**. However, such function makes use of future $\langle y(i)\rangle$ values; therefore, its use on this *online* problem is not possible. As a result, we are interested in some estimation of $similar(w_l,w_r)$ suitable for *online* use; that is, an efficient solution that relies solely on knowledge of $\langle y(i)\rangle$ values at hand. To this end, the deceitfully simple ideas introduced in **HPS Conditioning** will be shown to add to a whole greater than the parts.

Subsection 5.2.1: The HPS Conjecture

To estimate the presence of a **localized stationary condition** ‹›, we promote the concept of **approximate τ-invariance**. Two conditions are necessary for **approximate τ-invariance** – (1) stationary mean and (2) finite-and-small variability across a finite interval.[44] To achieve this, an *inference* speculates whether (*or not*) "the "present", poised by the outlook of the **HPS fast signal**, is ***sufficiently***

[43] For example, consider points (p) or (e) in Fig. 15.

[44] See [MONTGOMERY:SPC] for the benefits of said statistical properties.

similar to that of the known “recent past”, poised by the outlook of the **HPS slow signal**.

outlook	*sym*	*indicator*	*signal*	*clt-stabiliz.*	*interval*
present	‹›	*μ[‹y(i)› |m’]›*	*HPS fast Signal*	*UWMA(m’)*	*((i-τ), i]*
recent past	‹›	*‹μ[‹y(i-τ)› |m]›*	*HPS slow Signal*	*UWMA(m)*	*(i-(τ+m), (i-τ)]*

Table 4: Components of the inferential approximation *HPS_Conjecture(‹›,‹›)*.

To this end, the choice of present and recent past is made to split an **HPS introspection interval** ‹› into two non-overlapping time sub-intervals ‹› and ‹›, where ‹›≡‹›∪‹›.[45] **Table 4** provides a comparison of these two outlooks. Effectively, this speculates whether sampled population means (and variances) across outlooks ‹› and ‹› – *taken from CLT-stabilized signals* – are different enough[46] to reject (or otherwise, similarly enough to accept) the **HPS conjecture**[47] across their combined span ‹›.

When **approximate τ-invariance** spans both present and recent past, their tracking indicators ought to be ***similar*** under statistical confidence. A straightforward application of the “**approximate τ-invariance**” test (**4.0**) would be neither efficient nor robust as it would be both arbitrary (e.g., which ε criterion to use) as well as computationally expensive (e.g., an instance has to be tested against *all* other instances). Moreover, the test relies on limiting averages, which extend an *infinite* influence function[48] to *heavy tail* outliers as well as to long-term correlation. Fortunately, statistical tests exists which tradeoff aspects of robustness, algorithmic complexity, and sensitivity (see **Sidebar *“Statistical Testing Alternatives”***).

Statistical Testing Alternatives

1. The most obvious test is the paired ***t-test*** [**LAPIN: PAIRED T-TEST**]. However, it is computationally expensive; each sample is paired against all other samples in subtests which are then pooled into the test statistic. Moreover, this test is indeed too sensitive for our needs as it magnifies by [illegible] times the influence of any outlier.
2. An alternative is the ***Wilkinson-Signed Rank*** (WKS) test [**LAPIN: SIGNED RANK TEST**], which is more robust. However, this test is also computationally expensive.
3. However, our robust indicators (see **Table 3**) can cleverly be used to *indirectly* but efficiently test (3.0) through a “***comparison of sampled population means for unequal variances***”.[49] This test is less sensitive than the ***paired t-test*** and less robust than the ***Wilkinson-Signed Rank*** test. The test has an optimal worst-case algorithmic complexity of just ***O(***1***)***. This test is our choice.

As stated on **Sidebar *“Statistical Testing Alternatives”***, choice [3] is our preference.[50] Our atypical use compares populations drawn at *different* time (that is, the present and recent past outlooks) but from *somewhat similar* sources (both are drawn from CLT-stabilized signals of different order but of the same realization ***‹y(i)›***). First, this lays out testing within the robust CLT-stabilized domain of ***‹y(i)›*** as

[45] Although, such outlooks could be of variable duration ‹› – with size |‹›| being adapted w.r.t. some optimality constraint – without loss of generality, we show how testing is robustly conducted in terms of finite duration outlooks across ‹y(i)›. Without loss of generality, we rely on fixed intervals (and) as opposed to random (‹› and ‹›).
[46] Note that the phrase “enough” stands in for “with statistically significance.”
[47] This single decision bit represents just a conjecture (that is, about the presence of approximate τ-invariance) and thus it is referred to as the HPS conjecture.
[48] See [PICCOLO:INFLUEN] for a review of influence functions w.r.t. robust statistics.
[50] That is, a comparison of sampled population means of different size under unknown mean and variance with H0: recent past ≈ present and H1: recent past ≠ present.

opposed to the unqualified domain of ***‹y(i)›***.[51] Second, together with the use of ***UWMA(•••)*** windowed outlooks for ***‹y(i)›***, this limit the influence of outliers and curtails long-term correlation in ***‹y(i)›***.[52]

Fig. 8: An inferential approximation to *Similar()*. Testing setup for the HPS conjecture at some time *i* *w.r.t.* the outlooks of the CLT-stabilized signals.

Fig. 8 illustrates this testing setup at time ***i***. The recent past (shown in light shade) and the present (shown in dark shade) represent the populations being tested.[53] As stated, the **inference** speculates whether (*or not*) the combined interval ‹›≡‹›∪‹› satisfies "**approximate τ-invariance**" conditions. The output of this inference is a decision bit, which represents solely a ***conjecture*** about the (existence of) **approximate τ-invariance** across ‹› given to the knowledge available at time ***i***. Therefore, for emphasis, hypothesis testing is referred to as the **HPS conjecture**.

Fig. 9: Intuition into sequential hypothesis testing. The recent past outlook (of size *m*) is shown in light shade and the present outlook (of size *m'*) in a darker shade at three decision points (*c, f, h*). These also illustrate the sizeable differences in influence function at each decision point for each outlook.

Fig. 9 provides intuition into this sequential hypothesis re-testing achieved through a ***comparison of sampled population means*** test. Note how color-coded sets of paired bounding boxes are used to illustrate the outlook pairs (i.e., recent past, present) being compared at *three* different decision points in time, these being ***c***, ***f,*** and ***h***. In this coding scheme, the recent past outlook (i.e., from the **HPS slow signal)** is shown in *light blue* and the present outlook (i.e., from the **HPS fast signal)** in *red*. Note how bounding boxes also provide a visualization of the influence function associated with each outlook at each decision point.

Sequential Hypothesis Testing Setup

The illustration below shows a simplified view into the setup of sequential hypothesis testing for "**approximate τ-invariance**" at time index ***i***; the setup is repeated at *each* time index. It tests whether two specially constructed sampled population means are *approximately* equal. Specifically, the setup compares the ***UWMA***()-based estimators associated with the current outlooks for the *slow* signal against that of the *fast* signal.

View into the setup of hypothesis testing for "approximate τ-invariance" at time index *i*.

The testing and test space can be *visualized* through an array of possible **(μ_{slow}, μ_{fast})** ranges of value pairs where μ_{slow} represents the test mean for the *slow* signal and μ_{fast} the value for the *fast* signal. The following table is provided for facilitate visualization; it depicts assertion rules for the **HPS conjecture** corresponding to statistically significant departure from **localized stationary condition**s. Table entries are read as follows; an entry on the table being *zero* represents *tolerable* alignment of sampled means (and thus a *transient* assertion of the **HPS conjecture**) whereas a *non-zero* entry represents *strong* misalignment (and thus a *definitive* denial of the **HPS conjecture**).

slow/fast
μ-3σ

51 This is a recurrent theme in the construction of robust statistical tests. However, while being robust to normality departures on ‹y(i)› , other approaches tend to be computationally expensive, as for example, the Wilkison-Signed Rank test discussed in choice [2].
52 These technical issues are explored in detail in Technical Considerations.
53 Note that here τ =m/2=m'.

μ-2σ ***μ-1σ*** ***μ*** ***μ+1σ*** ***μ+2σ*** ***μ+3σ***	***0*** ***0*** ***0*** ***0*** ***0*** ***1***
μ-3σ ***1*** ***1*** ***1*** ***1*** ***1*** ***1*** ***1***	***μ+2σ*** ***1*** ***0*** ***0*** ***0*** ***0*** ***0*** ***1***
μ-2σ ***1*** ***0*** ***0*** ***0*** ***0*** ***0*** ***1***	***μ+3σ*** ***1*** ***1*** ***1*** ***1*** ***1*** ***1*** ***1***
μ-1σ ***1*** ***0*** ***0*** ***0*** ***0*** ***0*** ***1*** ***μ*** ***1*** ***0*** ***0*** *0* ***0*** ***0*** ***1*** ***μ+1σ*** ***1***	**Assertion rules for the HPS conjecture *w.r.t.* statistical significant departures.** Note that the table depicts *rough* statistical regions on *decreasing* confidence (represented by lighter shades of gray), starting at the ideal alignment ***(μ, μ)*** where confidence on $\boldsymbol{\mu_{slow}=\mu_{fast}}$ peaks. When **localized stationary condition**s are present, the **HPS conjecture** may be asserted by *any* of the darker shades of gray; this is partly due to inherent variability. Essentially, a continuously but transiently asserted **HPS conjecture** will *freely* roam within said region of tolerance across time indexes corresponding to said localized stationary regions. However, this *continuous* sequence of transient assertions affirms with confidence the **HPS hypothesis**, that is, the presence of "**approximate τ-invariance**" across a corresponding interval on the input signal. In contrast, when stationary conditions are *not* present, the **testing indicator** forces away from the tolerance region for transient assertions, thus forcing a definitive denial of the **HPS conjecture**.

Subsection 5.2.2: Sequential Hypothesis Testing

This simple testing choice is *deceitfully* clever. It chooses to compare a carefully-constructed pair of sampled population means (that is, recent past and present) – taken from non-overlapping opposite segments of an *introspection* interval ***=f(m,m′,i,τ)*** – in order to speculate a decision bit (*transiently*) tracking the presence of **approximate τ-invariance** across said interval . For this reason, the interval ***=f (m,m′,i,τ)*** is referred to as the **HPS introspection interval**. **Sidebar** "***Sequential Hypothesis Testing Setup***" gives intuition into the formulation of the ***i-th*** **HPS conjecture** and its decision bit outcome. When said decision bit is set, the **HPS conjecture** is *transiently* asserted; however, when the decision bit is *not* set, the **HPS conjecture** is *conclusively* denied.

As stated, the statistical validity of the conjecture is limited to time ***i***. For example, within the interval ***[b, c]*** of **Fig. 9**, it is *impossible* to determine whether the localized stationary condition ***(a, c)*** ended or a transition is taking place – as an *online* algorithm has no awareness of *future* **‹*y(i)*›** values. Due to this uncertainty, the above test setup needs to be re-applied at each subsequent time ***j›i***, to re-validate (or otherwise invalidate) at *each* time ***j*** any affirmative decision bit obtained at time ***j-***1. Therefore, at each subsequent time ***j›i***, outlooks are updated and the corresponding **HPS conjecture** is setup to incorporate *new* knowledge made available by the **‹*y(j)*›** observation.
Corresponding to subsequent time index ***j›i*** (and thus an updated interval ***=f(m,m′,j,τ)***), a new decision bit is speculated. By transitivity, the repeated application of this *approximate* test effectively applies the **approximate τ-invariance** criteria (**4.0**) across all of **‹*y(i)*›**. By virtue of this design, underlying **localized stationary conditions** in **‹*y(i)*›** – having duration *greater* than the **HPS introspection interval** – are unearthed as continuous transient assertions of the **HPS conjecture**. In actual fact, the approximate presence of **localized stationary conditions** in **‹*y(i)*›** is unearthed in terms of *random-length* **ATS segments ‹›** whose length ‹› is determined by the probabilistic likelihood of unbroken strings of "continuously upheld **HPS conjectures"**. Each such string *affirms* the **HPS hypothesis** – that is, the presence of **approximate τ-invariance** – across the corresponding *random-length* time segment ‹›. Astonishingly, this results in the sequential uncovering of all localized stationary conditions in **‹*y(i)*›** *regardless* of their time-scale!

Subsection 5.2.3: HPS Approximation

In summary, a stationary-based approximation provides innovative means to address signal compressibility.[54] Our stationary-based encoding seeks to reduce each qualified input token ‹› (for example, an **localized stationary condition ‹›**) of random length **‹›=‹u_{high}›-‹u_{low}›** into a token (that is, **ATS segments**) of fixed size 1 and *estimated* value of **‹f_0 (‹›)›**. This way, *any* random-length string of continuously upheld **HPS conjectures** is encoded into *two* (segment delimiter) tokens (referred to as **‹u_{low}›** and **‹u_{high}›**). However, within the resultant **HPS approximation**, one (*or more*) **ATS segments** may represent a *true* localized stationary condition. In other words, a *true* localized stationary condition is *likely* to be represented through a one-to-many

54 The HPS transform generates a resultant HPS approximation that projects a signal ‹y(i)› into; (1) its stationary-based approximation, (2) a time series of heavy-tail outliers, and (3) a time series of the residuals resulting from said stationary-based approximation to the signal ‹y(i)›. However, the use of CLT-based confidence limits results in a signal ‹ô(i)› of negligible size w.r.t. input size N. As a result, the total number of outliers ‹ô(i*)› exceeding detection limits is bounded in probability. Moreover, the error ‹ê(i)›, when computed w.r.t. the CLT-stabilized signals of ‹y(i)›, could be approximated as Gaussian noise N().

relationship between the localized stationary condition and its unearthed **ATS segments**. As a result, if **‹*n*›** represents the *true* number of **localized stationary conditions** present within **‹*y(i)*›**, these will be unearthed as an unknown number **‹*n**›** (where **‹*n**› ≥ ‹*n*›**) of random-length **approximate τ-invariance** time segments (**ATS segments**).[55] This way, the **HPS transform** maps an input signal **‹*y(i)*›** of size ***N*=‖‹*y*›‖** into a sequence of **‹*n**› ATS segments (**and the **‹*n**›+1** fine-tracking transitions between consecutive **ATS segments**).

Current Decision Bit	*current* decision	*current* decision
Previous Decision Bit		
***previous* decision bit *is* set**	**Rule 1 ≝ *f*(μ_{slow} *(i*-1*))***	**Rule 3 ≝ *f*(μ_{slow} *(i))***
***previous* decision bit *is not* set**	**Rule 2 ≝ *f*(μ_{slow} *(i))***	**Rule 4 ≝ *f*(μ_{slow} *(i))***

Table 5: Stationary-based encoding.

However, it is *not* possible for any *online* algorithm to compute an *optimal* value for an **ATS segment**. Instead, *a* representative value is *chosen* based on a stationary-based encoding rules contained on **Table 5**. Under these straightforward encoding rules, just two decision bits – the *current* and *previous* ones – are used to generate, by transitivity, a stationary-based encoding of the *current* sample of the underlying signal. The *selected* representative value is referred to as the referred to as the **HPS forecast**. The **HPS forecast** represents a conservative estimate based on an average of values carefully taken from the **HPS slow signal**. The rationale behind these encoding rules is explained on ***Sidebar "Use of the Decision Bit"***.

Use of the Decision Bit

1. If the *previous* decision bit was set, a set bit *now* represents a transient assertion, i.e., a continuance[56] of a possible **ATS segment**. Here, the **HPS forecast** for the *ongoing* **ATS segment** is kept; *no* revision to the forecast is made – even we have more data about the *estimated* mean of said **ATS segment**. When *true* τ-invariance is present (i.e., mean and variance are approximately *constant*), by virtue of said **localized stationary condition**s, the *new* value is *approximately* equal to the *previous* value of the **HPS forecast**. Hence, we take the liberty to speculate this transitivity claim and preserve the *previous* value of the **HPS forecast**. However, when "***approximate τ-invariance***" is present (i.e., **HPS conjectures** continuously upheld *while* the **HPS error bound** is respected), an *intentional* tradeoff in error is induced *w.r.t.* the stability of the stationary-based approximation.
2. If the *previous* decision bit was *not* set, a set bit *now* represents the start of a *new* **ATS segment**. Here, the **HPS forecast** is set to track the *current* value of its underlying signal. However, because the **HPS forecast** is taken from a CLT-stabilized source (i.e., the **HPS slow signal**, a reduced variability source that provides an unbiased, precise, and accurate sampling mean tracker for the original signal), said error component (i.e., the difference between *true* and *estimated* value of the **ATS segment**) is bounded.
3. If the *previous* decision bit was set, an unset bit *now* represents a definitive denial of the **HPS conjecture**, i.e., it represents the end of an **ATS segment**. Once again, the **HPS forecast** is set to track the *current* value of its underlying signal. However, note that because the **HPS forecast** is taken from a CLT-stabilized source which tracks the sampling mean of the original signal.
4. If the *previous* decision bit was *not* set, an unset bit *now* represents an ongoing lack of **localized stationary condition**s. Once again, the **HPS forecast** is set to track the *current* value of its underlying signal.

One way or another, an **HPS forecast** is simply a function of (*previous* or *current*) values of its underlying signal (that is, the **HPS slow signal**). The resultant **HPS approximation** *samples* the **HPS slow signal** conditioning such sampling *w.r.t.* the presence of **localized stationary condition**s in the **HPS slow signal**. Simply put, when **localized stationary condition**s are present, such sampling sticks onto

[55] In contrast, on the offline case, each ‹› was modeled by exactly one WSS segment.

some representative value – this being an average of recent values – from the **HPS slow signal**. Because the **HPS slow signal** is a CLT-stabilized version of the *true* input signal, overall **HPS forecasts** are (asymptotically) *consistent*, *bounded*, and *unbiased* in their tracking behavior of the input signal. This choice of value – by virtue of the reduced variability induced by CLT-stabilization over an *already* stationary condition – is *limited* in range; as it could only be *slightly off* (i.e., an *overtone*, an *undertone*) or even at the *true* baseline of said inferred **localized stationary condition**. Given this insight, we now examine **HPS error control**, introduced to address signal fidelity.

Fig. 10: Without future *‹y(i)›*, it is not possible to establish at time *t=b* whether a sample, say *[b]*, belongs to current segment *[a, c]*, to transition *[b, d]*, or to a new segment *[b, c]*.

Section 5.3: HPS Error Control

As stated, the stationary-based approximation induces an error along each time ***i*** of every **ATS segment**. We refer to this error as *quantization* error*.* However, there are *three* different kinds of quantization error because two additional signals are derived through CLT-stabilization of the input signal. For this reason, we refer to quantization error computed *w.r.t.* either **HPS fast signal** (or **HPS slow signal)** to as **HPS quantization error***,* and when computed *w.r.t.* the input ***‹y(i)›***, it is referred *instead* to as **HPS absolute error**. That is,

$$\langle \acute{\varepsilon}_{slow}(i)\rangle = \langle mon^*(i)\rangle - \langle \mu[\langle y(i-\tau)\rangle | m]\rangle, \ \& \ \langle \acute{\varepsilon}_{fast}(i)\rangle = \langle mon^*(i)\rangle - \langle \mu[\langle y(i)\rangle | m']\rangle. \quad (5.6)$$

Now, recall that the amount of error introduced by the **HPS transform** depends on whether (or not) there exists **(1)** a way to identify a **localized stationary condition** in ***‹y(i)›*** as well as **(2)** a robust way to accurately estimate its *true* baseline (i.e., **HPS fundamental frequency**). For example, as illustrated in **Fig. 1o**, given an arbitrary **ATS segment *‹›= [‹u_low›, ‹u_high›)***, *exact* determination of these two factors is *undecidable* for *any* online algorithm. However, as just argued, the **HPS transform** addresses these concerns through robust inferential construction (**HPS decision-making**) of our timescale-independent abstraction (i.e., **ATS segments**). This way, undecidability manifests as *unavoidable* ***α***, ***β*** errors in **HPS decision-making**[57] that at time, as stated, cause an unearthed **ATS segment ‹›** to be *slightly* off-target the *true* baseline ***‹f₀(‹›)›*** it tracks. **HPS error control** provides means to reset this occasional targeting inaccuracy. To this end, **Fig. 11** provides intuition into this reset mechanism through an example that depicts how **localized stationary condition**s (**‹›**) are transformed into a trajectory of **ATS segments ‹›**. **Fig. 11** has ***four*** parts, labeled **(a)** through **(d)**, which are explained next.

(a)	**(b)**
(c)	**(d)**

Fig. 11: Intuition into HPS error control. Part (a) shows a signal with three localized stationary conditions (shown in shades). Part (b) shows the *optimal* forecast for such. Part (c) shows the resultant trajectory of a stationary-based approximation. Part (d) shows accumulation of HPS quantization error.

[57] See [LAPINOC] for a review of α and β errors and the design of operating curves.

Part (a) shows a signal with three **localized stationary condition**s (labeled ***a*** through ***c***) shown with their time spans. Note how shaded areas emphasize the inherent variability present – an *irreducible* source of quantization error. **Part (b)** provides intuition about their *baseline* or **HPS fundamental frequency** of each localized stationary condition. Such (***MLE***) value (labeled $\boldsymbol{\mu_a}$ through $\boldsymbol{\mu_c}$) is the *optimal* representative value; it minimizes **HPS quantization error**.[58] **Part (c)** illustrates the resultant **HPS approximation**, that is, the interlacing of unearthed **ATS segments**. Through **HPS decision-making**, (one or more) **ATS segments** are sequentially unearthed for each localized stationary condition. **HPS quantization error** now results from **(1)** delay (in the build up of confidence) during **HPS decision-making** as well as **(2)** inherent variability present within the **localized stationary condition**. The former is minimized by reducing delay on the tracking of localized stationary conditions, whereas the later is minimized by estimating the *optimal* representative values on such tracking. Now, recall that it is *not* possible for any *online* algorithm to compute the *optimal* representative value; therefore, *a* representative value (herein labeled $\boldsymbol{\sim\mu_a}$ through $\boldsymbol{\sim\mu_c}$) is *chosen* based on a stationary-based encoding rules contained on **Table 5**. As long as **HPS conjectures** continue to be upheld, our *initial* choice of representative value is kept "for as long as such choice remains fit" – that is, *until* the buildup of the quantization error (it induces) exceeds that compatible with the presence of **approximate τ-invariance**. Finally, **part (d)** provides intuition into error sources related to tracking accuracy and targeting accuracy. The handling of these is discussed next.

Subsection 5.3.1: Tracking and Targeting Accuracy

On one hand, accumulation of *large* **HPS quantization error** flags a definite departure from **approximate τ-invariance**. Such can be due to: **(1)** outliers and **(2)** the end of the localized stationary condition. On the former, such a large departure (on the mu domain of **‹*y(i)*›**) triggers the failure of the corresponding **HPS conjecture** and thus the end of the *current* **ATS segment** (if any). On the later, such occurs when an **ATS segment** lingers on (on µ(···)-domain of **‹*y(i)*›**) while the underlying localized stationary conditions ended (on the domain of **‹*y(i)*›**). As a result, HPS quantization error steadily accumulates (for example, those shown by the label of $\boldsymbol{MSE_{a\text{-}b}}$ and $\boldsymbol{MSE_{b\text{-}c}}$) and (*within small delay* ***t=m +m'***) thus also triggers the end of an **ATS segment.** For this reason, this error-control measure relates to tracking accuracy.

On the other hand, if **HPS quantization error** is *consistently small*, it is merely permitted to accumulate within a goodness of fit metric– referred to as the **HPS segment MSE**. This goodness of fit metric tracks the accumulation along the entire span of an unearthed **ATS segment** for a specially constructed accumulation of **HPS quantization error**. For example, this error source is due to inherent variability (shown enclosed within shaded areas for localized stationary conditions ***a***, ***b***, and ***c***). This way, a buildup of **HPS segment MSE** can (over time) *also* trigger the end of the *current* **ATS segment**. Such would depend on how accurate is the targeting of the *true* baseline of the underlying **localized stationary condition** – as such ***MLE*** value minimizes error accumulation resulting from *inherent* variability. For this reason, this error-control measure relates to

[58] That is, for any ATS segment unearthed along a localized stationary condition.

targeting accuracy.

As a result, tracking and targeting errors are tightly bounded and as a result, error is kept small and stable.

Fig. 12: HPS transform block diagram with error-control feedback.

Subsection 5.3.2: Autonomous Error Bound

Both tracking and targeting accuracy assume that *somehow* a robust bound could be *autonomously* derived for **HPS quantization error** at *any* time index for *any* arbitrary signal – otherwise any such bound would be arbitrary – and thus, flawed. Fortunately, we introduce next the **MSE Equivalency Theorem**, which proves one such very *general* result. Succinctly, the theorem states that there exists a well-defined relationship between *overall* statistical confidence in the sequential re-testing of the **HPS conjecture** across the interval and a specially constructed form of **HPS quantization error**. The **MSE Equivalency Theorem** provides an adaptive error bound (*‹MSE_{max}(i)›*), referred to as the **HPS error bound**, that makes possible the integration of such as a conditioning prior into **HPS decision-making** (see **Fig. 12**). This adaptive error bound allows to continuously constrain the accumulation of error during speculation of a *set* of incrementally overlapping **HPS introspection interval**s ***{ }*** (each transiently asserting the **HPS conjecture**) into *one* **ATS segment** ‹›. As stated, by transitivity, said **ATS segment** upholds the **HPS hypothesis**; however, now it is *also* constrained along its span *w.r.t.* the accumulation of this error form**.** For completeness, it is pointed that error accumulation is associated with a portion () of the **HPS introspection interval** being speculated – whether or not said interval ends up being part of the current **ATS segment** ‹›. **Fig. 12** shows the *revised* block diagram of the **HPS transform**; a conditioning prior is now feedback to **HPS decision-making**.

Subsection 5.3.4: Mse Equivalency Theorem

Let ***{}*** be a set of successfully speculated, incrementally overlapping, **HPS introspection intervals** spanning the time segment ***[‹μ_{low}›,‹μ_{high}›]***. We denote this **ATS segment** generation process by ***{ }*** →‹›, and refer to such as **HPS interval coalescence**, which we formalize as:

{ } →‹›:   , such that. **(5.7)**

The **MSE Equivalency Theorem**, next, provides the basis for the application of rigorous error control over **HPS interval coalescence**. Let ***‹f(i)›*** be a signal, let ***‹f(i-τ)›*** be its ‹**τ**›-delayed version, and ***α*** be a confidence level. Let ‹› be an arbitrary finite interval of size ***m'***. Then, at an ***α*** confidence level, the maximum error permissible ***‹MSE_{max}(i)|m'›*** along an interval ‹› of signal ***‹f(i)›*** if **approximate τ-invariance** exists across interval ‹› of signal ***‹f(i)›*** is bounded by

(5.8)

where ***‹ζ(i)|(m,m')›*** represents an error correlation given by

, **(5.9)**

‹μ[‹σ_D(i)|(m,m')›]› represents the average of the pooled standard deviations across the interval ‹›, and where ***t_{max} ≡ t(m+m'-2, α/2)***. The proof is presented on **Sidebar "*Proof to MSE Equivalency*"**.

Through the application of the **MSE Equivalency Theorem**, it is possible to constraint **approximate τ-invariance** hypothesis testing *w.r.t.* an autonomously adapted error-control goal – applicable for *any* signal. This results in the uncovering of **ATS segments** under consistent *resulting* error *w.r.t.* *assumed* confidence. Effectively, for *any* signal ***‹y(i)›***, a consistent confidence level in **HPS decision-making** is sustained – across *all* **ATS segments** uncovered in ***‹y(i)›*** – by translating such into an autonomously adapted bound over accumulated **HPS quantization error** across each one of such said **ATS segments**. As a result, an overall "goodness of fit" over the **HPS interval coalescence** process ***{ }*** →‹› is sustained under bounded probability *w.r.t.* the adaptive error bound specified in accordance to the **MSE Equivalency Theorem** – resulting in tracking and targeting accuracy being kept under rigorous error-control. More importantly, as an desirable side-effect, both **HPS decision-making** and **HPS error-control** are controlled solely by the values of ***m***, ***m′***, and ***α*** chosen.

VI. Formulation

Next, we formally specify an *online* implementation of the **HPS transform**, which we refer to as the **online HPS monitor**. For computational efficiency, partial sums (where the ***nth*** partial sum is) are used throughout; for example, as used in computing windowed sampled means and variances shown below:

, ,

, and **(7.1)**

However, a delay is necessary to initialize CLT-stabilization of ***‹y(i)›***. Its minimal value is the size of the **HPS introspection interval** plus ***m′***: ***d = (m+m′) + m′.*** During this warm-up, the **HPS approximation** defaults to tracking of the "***sampling mean***" of ***‹y(i)›***, that is,

‹mon(i+1)(m,m′)› = ‹μ[‹y(i)›|i]› , for 0≤i≤d. **(7.2)**

In accord to practices, the value ***‹mon(i+1)(m,m′)›*** is referred to as the **HPS forecast *‹mon*(i)›***). After this initialization, the **HPS approximation** is based on **HPS decision-making**.

Proof to the MSE Equivalency

1. The t-test ***‹t*(i)|(m,m′)›≤t_{max}*** is defined as .
2. For our super-heterodyned case, .
3. By (**5.6**)›, **(2)** is also equivalent to .
4. Squaring both sides of **(3)** results in .
5. In turn, **(4)** is equivalent to .
6. By collecting terms in **(5)**, .
7. By letting ; **(6)** becomes .
8. Summation of (6) holds the same relationship over any interval, as follows: .
9. Constraining our view of past history in **(8)** to a window of ***m′***, we get .
10. Explain the following approximation (mean, positive, inequality less than, and finite, bound, and constant for ts and large otherwise….:
11. Resulting in .
12. Taking roots,
13. Since by definition,, then .

Thus, .■

As stated, the **HPS conjecture** is implemented as a comparison of sampled population means having unequal and unknown variances[59]:

, (7.3)

where **‹Δμ(i)|(m,m′)›** represents the estimated difference between means (of present and recent past outlooks), ***‹t*(i)|(m,m′)›*** represents the test statistic, and ***t(m+m′–2, α/2)*** the ***t-test*** threshold:

‹Δμ(i)|(m,m′)› = ‹μ[‹y(i-τ)›|m]› – ‹μ[‹y(i)›|m′]›, **(7.4)**

‹t*(i)|(m,m′)› = μ(i)|(m,m′)› / ‹σ_D(i)|(m,m′)›, **(7.5)**

where the sampled pooled deviation ***‹σ_D(i)|(m,m′)›*** is estimated as:

. (7.6)

The **HPS trigger function** ***‹π(i)m′›*** is used to determine when to update the **HPS monitor signal** as follows:

, **(7.7)**

where the terms ***‹ω(i)m′›*** is discussed below. When a new **localized stationary condition** is unearthed, the **HPS forecast** is set to the present value of the **HPS slow signal** as follows:

‹mon*(i)› = ‹μ[‹y(i)›|‹ω(i)m′›]›. **(7.8)**

Note that ***‹ω(i)m′›***, referred to as the **HPS rating-function**, scales the size of the outlook to be used in generating the **HPS forecast**:

‹ω(i)m′› = (‹›) • m′. (7.9)

The **HPS rating-function** binds the "influence" of past observations over the generation of the **HPS forecast** for the newly discovered **localized stationary condition**. This is done by continuously rating the intensity of **approximate τ-invariance** present in ***‹y(i)›***. This rating, ‹›, is referred to as the **HPS relative index** and for succinctness, is formally specified elsewhere.[60] When **approximate τ-invariance** holds, ***‹ω(i)m′›*** opens up the "window" to make the generation of the **HPS forecast** more robust to outliers (i.e., by weighting more samples). On the other hand, when it does not hold, ***‹ω(i)m′›*** trims down the "window" to focus the generation of the **HPS forecast** into the transient region that triggered the update.

The **HPS outlier signal**, ***‹ô(i)›***, flags only 'statistical significant" outliers ***‹ô(i*)›***. These outliers are not ordinary outliers; they rather represent supra-ordinary (heavy tail) events.[61] Let ***‹x(i)›*** represent the original sampled observations, the **HPS outlier signal** flags only those observations that fall outside either the ***(1-π)*** or the ***(π)*** percentiles (where ***π=P(‹x(i)›≥K•‹σ[‹x(i)›|m]›)***). The input signal ***‹y(i)›*** becomes just the ***(1-π)*** plus ***(π)*** percentile conditioned ***‹x(i)›***; that is:

(7.10)

where ***‹δ(i)|m›=‹x(i)›-‹μ[‹x(i)›|m]›***. The **HPS outlier signal** is then:

‹ô(i)› =‹x(i)› – ‹y(i)›. **(7.11)**

Very limited bookkeeping is needed to track **ATS segments**. In fact, we only need track the current (or otherwise, last unearthed) **ATS segment**. This can be done in terms of their duration and endpoints. Let ***‹π(i)m′›***, referred to as the **HPS trigger function**, be a function that determines whether a **localized stationary condition** has been entered or exited; this is done according to two simple rules:

1. If currently in a **localized stationary condition**, let ***‹π(i)m′›→1*** signal its furtherance whereas ***‹π(i)m′›→0*** signals its end.

[59] See [NRINC:T-TEST] for sampled population's mean and variances tests.
[60] In Appendix F, we formally specify the HPS relative index.
[61] See [WILLINGER:NORMALTHAN] for a review of heavy tail event theory.

2. Otherwise, if currently in a transient region, let ***‹π(i)|m′›→1*** signal detection of a new **localized stationary condition** whereas ***‹π(i)|m′›→0*** signals the furtherance of the transient region.

A simple **HPS trigger function** is our **HPS conjecture (7.3)**, a stateless function based on the outlook at time ***i***, and therefore, we have that:

‹π(i)|m′› = ***‹g(i)|Δμ(i)|(m,m′)›***. **(7.12)**

Let the **HPS segment marker** ***‹Ω(i)›*** be a memory that tracks the time indexes corresponding to the endpoints of uncovered **ATS segments**. Through (**7.12**), this is specified by the following recurrence:

. (**7.13**)

This way, at time ***i***, only if current and previous **HPS segment markers** differ (that is, ***‹Ω(i-1)› ≠ ‹Ω(i)›***) then those correspond to endpoints ***‹u_{low}›*** and ***‹u_{high}›*** for an **ATS segment**. This way, at any time ***i***, the duration (referred to as the **HPS segment duration**) of an **ATS segment**, is given by the following equation:

. (**7.14**)

When the **HPS trigger function** fires, the duration of the current **ATS segment** is determined to be the difference between the current and the value of the previous **HPS segment marker**.

The Goodness of fit for **HPS approximations** is ***MSE*** (***Mean Square Error***) based. As stated, because the **HPS transform** is defined *w.r.t.* two tracking signals, the ***MSE*** is also computed *w.r.t* both **HPS quantization errors**. The **HPS windowed MSE** ***‹MSE(i)|τ›*** represents the estimated ***MSE*** across the subinterval (i.e., ***[i- τ , i)*)**. At any time ***i***, it represents the accumulated ***MSE*** (across an outlook of size ***τ***) along the sub-interval and it is given by:

. (**7.15**)

At any time ***i***, the *actual* error contribution to **HPS windowed MSE** is referred to as the **HPS instantaneous MSE** and it is given by:

‹MSE(i)›= √(‹SSE(i)› - ‹SSE(i-1)›). (**7.16**)

At any time ***i***, the accumulated MSE across the span of the current ATS segment is referred to as the **HPS segment MSE,** and given by:

. (**7.17**)

Similarly, the "***squared-sum-of-errors***" ***‹SSE(i)›*** is also defined *w.r.t.* both **HPS quantization errors** as follows:

. . (**7.18**)

Moreover, to bind the influence of outliers, a ***τ=m′*** windowed outlook (corresponding to subinterval) is used and computed as:

‹SSE(i)|τ› = ‹SSE(i)› – ‹SSE(i-τ)›. (**7.19**)

Accumulation of **HPS quantization error** is controlled in terms of **HPS windowed MSE** *w.r.t.* the **HPS error bound**. As stated before, the **MSE Equivalency Theorem** provides the **HPS error bound**:

. (**7.20**)

Note that ***‹ζ(i)|(m,m′)›*** represents an "error correlation" between the **HPS quantization errors** across the subinterval ‹› and it is given by

, (**7.21**)

and ***‹μ[‹ζ(i)|(m,m′)›|m′]›*** represents the average of such "windowed error correlations". Whereas within an **ATS segment**, this term is negligible; otherwise, it may not (see **Fig. 13**). Similarly, ***‹σ_D(i)|(m,m′)›*** represents a "pooled standard deviation" whereas ***‹μ[‹σ_D(i)|(m,m′)›]›*** represents the average of such ***‹σ_D(i)|(m,m′)›*** across sub-interval ‹› of the **HPS introspection interval**.

HPS quantization error is controlled as follows. The **HPS windowed MSE** monitors accumulated error within **HPS introspection interval**, but specifically within the outlook corresponding to the sub-interval (of size m'); therefore, its **HPS error bound** is as follows:

$$\langle MSE(i)|m'\rangle \leq \langle MSE_{max}(i)\rangle \ . \qquad (7.22)$$

Then, since ***‹λ(i) › is ⍰›⍰***, at time ***i***, the **HPS error bound** for the **HPS segment MSE** for any arbitrary **ATS segment** is just:

$$\langle MSE(i)|\langle\lambda(i)\rangle\rangle \leq \langle\lambda^*(i)\rangle \bullet \langle MSE_{max}(i)\rangle. \qquad (7.23)$$

where the value of $\langle\lambda^*(i)\rangle$ is just i-$\langle\Omega(i)\rangle$.

VI. Experiment Setup

A *baseline* experiment was designed for controlled verification of the HPS transform.[62] Next, we describe the baseline experiment.

Section 6.1: Methodology

Recall that a central idea is to detect the presence of process states. Therefore, to baseline its performance, we designed an input signal ***‹y(i)›*** exhibiting "building-block properties" *w.r.t.* process states. To do this, we controlled the *location* of process states, the *transitions* between these, and the underlying *distribution* of the variability around these. This allowed us to analyze the response of the **HPS transform**.

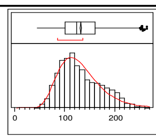
		
(a)	**(b)**	**(c)**

Fig. 18: Signal *‹y(i)›* being studied. Part (a) shows its tune plot. Part (b) shows its histogram and *pdf*. Part (c) shows its composition in terms of four random sources *‹φ$_k$(i)›* representing an underlying process baseline and three additive process states.

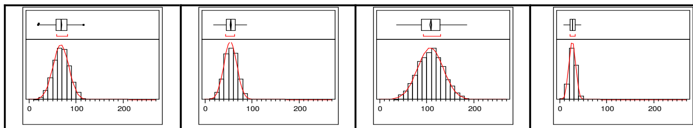
Subsection 6.1.1: Composition Fractal

The particular ***time series ‹y(i)›*** we used is shown in **Fig. 18**. This time series represents a "*composition fractal*" in the detection of process states. This claim follows immediately from noting that *any* sequence of process states can be reduced to sequences of *two*-state sequences ***Pi⎵Pj*** for which there here are only *two* possibilities, either ***Pi>Pj*** or ***Pi<Pj*** (as otherwise, ***Pi=Pj*** represents no transition). This "*composition fractal*" explores both possibilities through the sequence (***P1⎵P2⎵P3***) where ***P1<P2*** and ***P2>P3***. This implies that *w.r.t.* transitions between process states, *all* input signals can be reduced to *sequences* of this composition fractal, and as a result, the performance of the **HPS transform** could be baselined through analysis of this fractal.

Subsection 6.1.2: Mixture Model

The ***random process {X}*** was modeled by the generalized mixture distribution described in (**6.0**). Such model is representative of real-word processes. For example, consider ***{X}*** to be *response delay* of a remote shared resource. This model may associate an underlying random overhead $\langle\varphi_0(i)\rangle$ (e.g., network conditions and/or operational overhead) to the shared resource (e.g., a web server) over which multiple random sources $(\langle\varphi_k(i)\rangle$–$\langle\varphi_0(i)\rangle)$ (e.g., serving sessions) may also be applied at unknown times as such are typically of unknown duration and distribution, thus resulting in a mixture distribution.

[62] For those concerned, the HPS transform is applied to "real-world" data in Examples.



(a) **(b)** **(c)** **(d)**

Fig. 19: Histogram (and approximate *pdf* fit) for each underlying random source of *{X}*. Part (a) shows the baseline whereas parts (b-d) show the three On/Off additive sources. These random sources generate the input signal *‹y(i)›*.

Subsection 6.1.3: Lognormal Distributed

We wanted the ***input signal ‹y(i)›*** to incorporate departures from normality assumptions to assess the response of the **HPS transform** to such.[63] To this end, it was designed to follow a lognormal distribution.[64] This input signal ***‹y(i)›*** is shown in **Fig. 18**, which shows the resulting time series, its histogram (and approximate ***pdf*** fit), and its underlying random components. To construct this lognormal distribution, we applied the *generalized mixture distribution model of multiple random sources* (**6.0**) as explained below.

Subsection 6.1.4: Mixture Components

The mixture was composed of an approximately normally distributed random baseline ***‹φ_0(i)›*** (of duration ***N***) mixed with three approximately normally distributed ***random sources ‹φ_1(i)›, ‹φ_2(i)›*** and ***‹φ_3(i)›*** (each non-overlapping of duration ***N/3***). Histograms for these random sources are shown in **Fig. 19**. This way, the resulting mix[65] was a lognormal distribution: ***{X}≈LN(4.8,0.3)***.

Subsection 6.1.5: Random Variates

Random sources were generated by a ***random variate ‹U_0(•••)›*** which produced[66] an approximately normally distributed ***r.v.***[67] as follows:

$$\langle U_0(\varphi)\rangle = \varphi \bullet |\, N(rand(t)) + N(rand(t')) - N(rand(t''))|. \quad (7.0)$$

N(rand(•••)) represents the standardized normal distribution; ***t***, ***t'***, and ***t"*** tell apart three samplings from it; and ***φ*** is a scaling constant.

Subsection 6.2: Construction

These random sources generated the *"hidden"* **process states** to be unearthed by the **HPS transform**. They generated the following (*true but hidden*) normally distributed *contributions* to process state:

$\langle\varphi_0(i)\rangle$ =	$\langle U_0(100)\rangle$	for all i	≈ N(68, 17)	(7.1)
$\langle\varphi_1(i)\rangle$ =	$\langle U_0(80)\rangle$	i ϵ (1,1200)	≈ N(54, 14)	
$\langle\varphi_2(i)\rangle$ =	$\langle U_0(160)\rangle$	i ϵ (1201,2400)	≈ N(111, 27)	
$\langle\varphi_3(i)\rangle$ =	$\langle U_0(40)\rangle$	i ϵ (2401,3600)	≈ N(27, 7)	

As a result, the mixture distribution of ***{X}*** became visible[68] as:

$$\{X\} = \langle\varphi_0(i)\rangle + \langle\varphi_1(i)\rangle + \langle\varphi_2(i)\rangle + \langle\varphi_3(i)\rangle. \quad (7.2)$$

[63] The impact of such particular distribution is ameliorated because: (1) the robustness of HPS decision-making to normality departures; (2) the use of windowed operators to bound the influence function of outliers, and (3) the robust identification of outliers.
[64] Specifically, the hidden random process {X} was described as {X}≈LN(4.8, 0.3).
[65] That is, the sum of four normally distributed random sources (i.e., ‹φk(i)›) having different means, variances, and durations. See [REF:LOGNORMCLT; WILLINGER:NORMAL] for a review of relevant strong central limit theorems.
[66] This is a consequence [REF:STRONGCLT; REF:RANDOMVARIATES] of being generated by a linear combination of three identically distributed (normal) random variables.
[67] Moreover, its first moments were known: E[‹U0(φ)›]=(0.68)φ and S[‹U0(φ)›]=(0.17) φ.
[68] This fact is particularly important in order to assess the targeting quality (e.g., accuracy and precision) of the HPS transform w.r.t. the statistical filtering of such components.

Which, as a result, resulted in the following process states with underlying ***duration*** in time :

P1: $\langle\varphi^*_1(i)\rangle$ =	$\langle\varphi_0(i)\rangle + \langle\varphi_1(i)\rangle$	***for*** $1\leq i\leq 1200$	***(7.3)***
P2: $\langle\varphi^*_2(i)\rangle$ =	$\langle\varphi_0(i)\rangle + \langle\varphi_2(i)\rangle$	***for*** $1201\leq i\leq 2400$	
P3: $\langle\varphi^*_3(i)\rangle$ =	$\langle\varphi_0(i)\rangle + \langle\varphi_3(i)\rangle$	***for*** $2401\leq i\leq 3600$	

Without loss of generality, the duration of a process state was therefore *fixed* (and *known*) to be ***N/3=1200***. This was important in order to analyze ***lag*** on response *w.r.t.* inputs.

Magnitude differences between process states ***P1–P2*** and ***P2–P3*** were such so that they were statistically significant (*w.r.t.* past variability) but of different ***magnitude*** (***P1–P2 ≠ P2–P3***) and ***direction*** (i.e. ***P1–P2*** *increasing*, then ***P2–P3*** *decreasing* transitions). ***Inherent variability*** in ***P1***, ***P2*** and ***P3*** was made to be significant (*w.r.t.* ***P1–P2*** and ***P2–P3***). Moreover, inherent variability at each ***P1***, ***P2***, and ***P3*** was made to be different as in (7.1). Last, ***transitions*** between process states were made to be *instantaneous*.[69] This was of benefit to the analysis of the behavior of ***lag*** *w.r.t.* to shifts in process state. The above incurs in no loss of generality as follows. If the shift was modeled as gradual (or as a random walk) then the response of the **HPS transform** would have converged into tracking of the sampling mean until **approximate τ-invariance** was detected once again. Note this applies regardless of duration of the shift.

VIII. Parameters

The **HPS transform** is controlled via input, system, and control parameters (see **Fig. 2**). The first two types are described next.

Section 8.1: Basic Parameters

Input parameters provide user-specified tradeoff between lag in the response of the **online HPS monitor** and the statistical strength of CLT-stabilization – and thus, that of the resultant **HPS approximation**.[70] Input parameters are just the size (***m*** and ***m′***) of the outlooks used to generate the CLT-stabilized signals.

System parameters frame the performance of the **online HPS monitor** within a feasible operating region. At their default values, they are very stable.[71] The HPS transform exposes two functions to such fine-tuning control: **(1) HPS decision-making** and **(2) HPS outlier detection**. Note both these are *inferential* functions. The operating region of **HPS decision-making** is partly controlled by the specification of ***α*** – a statistical confidence level. The operating region of **HPS outlier detection** is chiefly controlled by the specification of ***K*** – a number of sigma levels. These parameters derive their recommended values from ***Gaussian*** properties under which they deliver high confidence to their respective *inferential* functions.

Section 8.2: Measurements Artifact

A concern is raised *w.r.t.* the determination of how

[69] The duration of a process shift is the time interval between the end of the current process state and the beginning of the next process state, if such exists.

[70] For example, if the underlying distribution of {X} is known, the specific CLT-stabilization order to achieve approximate Gaussian distribution would also be known and therefore, a tradeoff over statistical robustness of the approximation vs. induced lag is evident.

[71] That is, small changes on them yield small or no change in the response.

long into the past should the **HPS hypothesis** be applied in the search for **approximate τ-invariance**. Note that we defined such outlook[72] (i.e., in terms of the duration of the **HPS introspection interval** *)*, whose duration (i.e., **+**) is determined solely by the values of ***m***, ***m'***, and ***τ***. It turns out that through the application of incrementally overlapping sequential **HPS conjectures**; the choice of interval duration is inconsequential.[73] Bit by bit, each successfully held **HPS conjecture** *further* re-affirms the presence of an **ATS segment** (given the knowledge available at time ***i***) whereas the corresponding **HPS error bound** (**4.11**) maintains goodness of fit along the overall **ATS segment** *w.r.t.* a consistent ***α***. The **HPS introspection interval** stands as a measurement artifact; it defines the resolution (that is, a time span) at which introspection (that is, sampling and testing) takes place in the search for **approximate τ-invariance**. As a result, its duration is relevant only for detection of relatively short bursts of **approximate τ-invariance** (i.e., those shorter the **HPS introspection interval**).

Moreover, it is desirable that the subintervals and spanning the **HPS introspection interval** be contiguous; this way, the **HPS hypothesis** is validly asserted throughout all of . Clearly, and could overlap, but such would create correlation within the **HPS introspection interval**, so it is best if be continuous and and do not overlap. Given ***m***, the value of ***τ*** for such is just ***τ=m'=m/2***. This layout minimizes correlation between (recent past and present) outlooks and tests the **HPS conjecture** throughout the maximum continuous span of the **HPS** **introspection interval** (recent past plus the present). As a result, this construction is chosen.

Finally, the last question becomes whether (*or not*) the size of the outlooks ought to be adapted based on some constraint (such as the observed presence of **approximate τ-invariance**). However, such variable size outlooks can be counterproductive *w.r.t.* **HPS decision-making**. The **HPS introspection interval** is the *measurements artifact* and as such it need be consistent across *all* **HPS conjectures** in order to achieve robust sequential decision-making.

Section 8.3 Operating Curve

Fig. 22 illustrates the operating curve of the **online HPS transform**.[74] Three axes span the space: **(1)** the CLT stabilization order of the **HPS slow signal** (***m***), **(2)** the CLT stabilization order of the **HPS fast signal** (***m'***), and **(3)** the ratio (***K/k***) where ***K*** represents the number of sigma levels used to recognize **HPS outliers** and ***k*** represents the number of sigma levels used to speculate **HPS conjectures**.[75] Data points in this space are plotted in terms of cubes, for which *color-coding* is used to rank *accumulated* **HPS quantization error** and *volume-coding* is used to rank **HPS fractality** of resultant **HPS approximations**. Specifically, *blue* cubes correspond to *low* **HPS quantization error** and *red* ones to *high* while *small* cubes correspond to *low* **HPS fractality** and *large* ones to *high*. These metrics relate to the feasibility of resultant **HPS approximations** (i.e., goodness of fit and HPS fractality). The plot contains data points corresponding to some *60* **HPS approximations** produced by the **online HPS**

[72] That is, how long into the past of the fast and slow signals.

[73] That is, as long as CLT-stabilization is achieved for both signals and as long as approximate temporal stability spans a time interval longer than the HPS introspection interval.

[74] This analysis used an implementation without error-control parameters. Moreover, its HPS decision function used a fixed dispersion test ‹h(•••)› not the ‹g(•••)› test.

[75] Note that k is related to α, for example, k=3 corresponds to approximately α =0.01.

transform at different values of ***m***, ***m'***, and ***K/k*** for the same test data (see **Fig. 1?**).

Fig. 22: Example operating curve for the online HPS transform (without error-control).

Analysis shows a **stable bounded region** of optimal performance.[76] This region is shown (in *shaded outline*) on the lower-front part, comprising a broad range of values on ***m'*** and ***m***. The response of the **online HPS transform** was stable across values of ***m*** (ranging from ***10-240***) each evaluated at different ***m/m'*** ratios while for ***K=3***, lower values of ***k*** bounded this region. Together, these parameters frame the feasible operating region of **HPS decision-making**.

Section 8.4: Control Parameters

Control parameters frame the tolerance to error, as we need a way to achieve a tradeoff between **HPS fractality** (i.e., number of uncovered **ATS segments**) and accumulated **HPS quantization error**. Specifically, we expose two functions to such fine-tuning: **(1)** relaxation of the maximum accumulation of **HPS quantization error** along an **ATS segment** and **(2)** enforcement of maximum segment duration for any **ATS segment**. The former tradeoffs *increased* **HPS quantization error** for *decreased* **HPS fractality** while the later tradeoffs *increased* **HPS fractality** for *decreased* **HPS quantization error**.

Let us refer to an **online HPS transform** without these enhancements to by ***A1***, and to the one with these enhancements to as ***A2+A3***. Recall that the current **ATS segment** ‹› is tracked by the markers ‹***Ω(i-1)***›, ‹***Ω(i)***›, which correspond (*when different*) to ‹u_{low}› and ‹u_{high}›. To enforce a maximum duration Ω_{max}, is equivalent to force **HPS decision-making** to fail the **HPS conjecture** when such condition (‹$\Omega(i)$›-‹$\Omega(i-1)$›> Ω_{max}) is met. In turn, this forces the termination of the current **ATS segment** and the re-evaluation of the **HPS forecast**.[77] A similar argument applies to ***A3*** but this time *w.r.t.* the constraint ‹$MSE(i)|\lambda(i)$› ≥ $\lambda^*(i)$›•‹$MSE_{max}(i)$› is introduced. This bound can be relaxed by introducing a scaling factor κ_0, as in κ_0•$\lambda^*(i)$›•‹$MSE_{max}(i)$›, which results in a more lenient conceptualization of how tightly fit an **ATS segment** ought to be. *Increased* ***κ*** *loosens* the **HPS error bound** (and thus the goodness of fit) on generated **ATS segments**, whereas a *decrease* in ***κ*** *tightens* it.

To enforce this fine-tuning, the following equations are updated.

(8.?)

(8.2)

(8.3)

Note that the values ***1***, ***3***, and ***4*** just encode HPS conjecture, segmentation, or bounded error related failure conditions, respectively. The value of zero encodes (*a continuance of*) the presence of **approximate τ-invariance** at time ***i***. These changes effectively implements the ***A2+A3*** HPS monitor.

Section 8.5: Sufficiency Of Parameters

Next, we examine the response of ***A1*** to the baseline experiment in order to determine whether more parameter control is needed. Input parameters were

[76] Per our requirements, we are looking for an area of consistently small and blue cubes.

[77] For completeness, it is stated that whether (or not) a new ATS segment is then created depends on the nature of the stationary present (true vs. approximate) across the (forcibly ended) previous ATS segment and the subsequent one (if any).

$m=60$, $m'=m/2=30$; system parameters were $\alpha=0.001$, $K=3$, and $\tau=m'=m/2$; and control parameters were $\Omega_{max}=m+m=90$ and $\kappa_0=2$.

Fig. 23: Baseline performance of the A1 online HPS transform.

Fig. 23 shows the resultant **HPS monitor signal**. The figure has two parts. On the top part (*using the scale at the left*), the signal $\langle y(i)\rangle= \langle f(\langle x(i)\rangle)\rangle$ provides background to the ***UWMA(m)***-based sampling mean $\langle \mu[\langle y(i)\rangle|m] \rangle$, which in turns provides background to frame the **HPS monitor signal**. The resulting **HPS fractality** is extremely low[78] while the **HPS monitor signal** is insensitive to inherent variability *within* a **HPS process state** but as desired, sensitive to shifts between process states, which are recognized and handled – after a lag[79] – for which the **HPS monitor signal** tracks the sampling mean. Finally, note that the **HPS monitor signal** is started only after a **warm-up delay**; under which it builds both stability and confidence and during which, it again tracks the sampling mean. The bottom part (*using scale at the right*) shows the accumulation build up of **HPS segment MSE** across **ATS segments**.

In accordance to our goals, we look for low **HPS fractality** and low **HPS quantization error** but only when associated with unbiased, precise, consistent estimation of the mean. However, note that the resultant **HPS approximation** sometimes *overestimates* and *underestimates* the sampling mean. As a result, error behavior is inconsistent, large, and more importantly, not bounded.

Fig. 24: Baseline performance of the *A2+A3* online HPS transform.

Fig. 24 shows the performance of ***A2+A3***. As with ***A1***, instantaneous process shifts are recognized and handled after a lag and during the absence of **approximate τ-invariance**, the **HPS monitor signal** tracks the sampling mean. However, ***A2+A3*** produces tracking in terms of a significantly *larger* number of **ATS segments** of much *smaller* duration. Yet still, **HPS fractality** remains low. Moreover, goodness of fit is high as tracking and targeting error is now constrained on each **ATS segment**. More importantly, ***A2+A3*** is an unbiased and consistent tracker of the sampling mean. To this end, note that the build-up of **HPS segment MSE** along each **ATS segment** is now consistent, small, and bounded.

These results verify the control parameters; goodness of fit control parameters tradeoff **HPS fractality** and goodness of fit. Specifically, **HPS segment MSE** – that is, goodness of fit – was kept approximately constant (proportionally to the magnitude of inherent variability present) across **ATS segments**, and the number of **ATS segments** – that is, **HPS fractality** – was significantly small.

Let us now examine the performance of the ***A2+A3*** **online HPS transform** in detail.

[78] In the baseline experiment, HPS compressibility was $1-\langle n\rangle/N\approx 1-160/3600$ (i.e., 98%).
[79] Defined by the lag function $G(m, m', \tau)$ in (4.7), a function of the input parameters.