ROBUST RECOVERY OF DETERMINISTIC TIMECODE SIGNALS UNDER ANALOG DEGRADATION

Brady Cruse

Massachusetts Institute of Technology
Cambridge, MA, USA
bcruse@mit.edu

ABSTRACT

Digital vinyl systems recover a causal playback-control trajectory from a deterministic stereo waveform. This paper presents a controlled evaluation framework for this symbolic recovery task and applies it to three lightweight policies with a shared bitstream decoder: a Fixed-Threshold Symbol Decoder (FTSD), an Adaptive-Threshold Symbol Decoder (ATSD), and a State-Constrained Temporal Decoder (SCTD). The experiment uses a 10-s, 44.1-kHz segment of a commercial 20-bit timecode profile with 1000 position events per second. Additive-noise and burst-dropout conditions are repeated over five fixed random seeds, hard clipping is sampled at 0.025-full-scale increments between 0.30 and 0.50, and isolated dropouts are used to measure reacquisition. Accuracy is measured relative to an operational reference obtained by applying FTSD to the clean segment, not to independently instrumented physical ground truth. At a noise-to-signal RMS ratio of 0.06, ATSD raises mean confirmed availability from 0.868 to 0.936, while SCTD raises total availability to 0.981 using marked predictions. For 64-sample repeated dropouts, SCTD raises availability from 0.396 to 0.641 without changing the 0.396 confirmed rate. Fine clipping measurements locate a common acquisition boundary between 0.375 and 0.400 FS rather than a meaningful policy ranking. The contribution is the matched comparison and explicit separation of availability, reference-relative correctness, and temporal behavior; the component methods are adaptations of established detection and tracking ideas.

1. INTRODUCTION

Real-time audio control systems often encode symbolic information inside a continuously varying analog carrier. In these applications, the desired output is not the waveform itself but a stable symbolic state, such as absolute position, direction of travel, or playback rate. The design problem therefore differs from conventional audio restoration: the objective is not necessarily to reconstruct the clean waveform sample by sample, but to recover a correct and low-latency control trajectory from a degraded observation.

The motivating application is a digital vinyl system (DVS). A DVS uses a control record or control CD to manipulate software playback from a turntable or CD deck. A stereo control waveform is digitized, decoded into absolute position, speed, and direction, and mapped to a digital track. This preserves low-latency tactile interaction while enabling digital media and audio processing [1, 2, 3]. A decoder must therefore balance continuity against correctness: an unlock interrupts control, but a plausible wrong position can create an audible jump.

Commercial profiles are not interchangeable. They may differ in carrier frequency, stereo phase convention, modulation, register length and feedback polynomial, state period, and position mapping. Open-source implementations consequently maintain profile-specific definitions [4, 5]. Here, "anonymized" means only that the vendor and product name are omitted. The waveform is not transformed, and the correct profile-specific sequence is used.

This problem sits naturally within real-time music signal processing because the target is a causal control trajectory rather than a restored waveform. It is related to feature tracking and parameter conditioning in interactive audio systems [6, 7, 8], hard versus reliability-conditioned decisions [9, 10], and model-based tracking [11, 12]. The policies below are transparent representative baselines and adaptations. We do not claim that threshold detection, adaptive hysteresis, or first-order prediction is novel in isolation, nor do we claim a comparison with every production decoder.

The contribution of this paper is threefold. First, it formulates the evaluation around three separate questions: whether a state is available, whether it agrees with a clean operational reference, and how the state behaves over time. Second, it compares three causal policies that share one deterministic-state decoder and isolate fixed crossing detection, adaptive crossing detection, and bounded temporal hold. Third, it reports a controlled comparison under noise, repeated burst dropout, and clipping, including all component metrics and a fine clipping sweep. This matched structure is important because it lets changes in behavior be connected to the recovery policy rather than to an unrelated front-end change.

The remainder of the paper defines the deterministic signal and shared front end, describes the three policies, formalizes the evaluation framework, reports the controlled experiment, and discusses the resulting robustness–reliability tradeoff.

2. SIGNAL AND SHARED DECODER

2.1. Observation model and deterministic state

Let $\mathbf{x}[n] = [x_L[n], x_R[n]]^\mathsf{T}$ denote the clean stereo signal and let

$$\mathbf{y}[n] = \mathcal{D}\{\mathbf{x}[n]\} + \mathbf{w}[n] \tag{1}$$

denote a degraded observation. The operator $\mathcal{D}\{\cdot\}$ represents structured corruption such as clipping or dropout, while $\mathbf{w}[n]$ represents additive noise.

The symbolic sequence is generated by a 20-bit linear-feedback shift register (LFSR). If $\mathbf{b}_k$ is the register state at event k and $\mathcal{F}$ is the profile-specific transition, then

$$\mathbf{b}_{k+1} = \mathcal{F}(\mathbf{b}_k). \tag{2}$$

The 950000 valid states are generated once and stored in a lookup table mapping each recovered 20-bit word to absolute position. The profile resolution is 1000 positions/s.

This deterministic structure provides more than a convenient position index. Not every 20-bit word belongs to the profile trajectory, and a locally plausible bit can still be inconsistent with the expected progression. Requiring a sequence of valid transitions therefore acts as an implicit error check. It also makes absolute position identifiable from a sufficiently long local history rather than from a periodic carrier phase alone. The decoder uses these constraints before any temporal prediction is considered.

2.2. Stereo decision events

A *decision event* is the selected-polarity crossing of the secondary channel at which the primary-channel magnitude is sampled and one candidate LFSR bit is emitted. At nominal speed there is one such event per millisecond. At $f_\mathrm{s} = 44100$ Hz, the nominal spacing is

$$N_\mathrm{event} = \frac{f_\mathrm{s}}{1000} = 44.1 \text{ samples/event.} \tag{3}$$

This conversion is useful when interpreting a dropout specified in audio samples: a 64-sample gap spans about 1.45 nominal decision events, whereas 256 samples span about 5.80 events.

Both channels are used. Their crossing polarity determines direction, and the complementary channel supplies the magnitude used for the bit decision. The scalar recursions below are applied per channel; the signal is not downmixed to mono, and no learned stereo fusion is used. The implementation also supports the profile's phase, primary-channel, and polarity conventions when determining forward and reverse progression.

2.3. Baseline and crossing detection

Each channel $c \in \{L, R\}$ has a slowly varying baseline

$$z_c[n] = z_c[n-1] + \alpha_z\big(x_c[n] - z_c[n-1]\big), \qquad \alpha_z = \frac{1/f_\mathrm{s}}{\tau_z + 1/f_\mathrm{s}}, \tag{4}$$

with $\tau_z = 1$ ms. This recursion removes DC offset and slow drift without requiring a block buffer. FTSD declares a polarity transition at $z_c[n] \pm T_0$, where $T_0 = 128$ in 16-bit PCM units.

ATSD and SCTD additionally track a short-term absolute-amplitude envelope

$$a_c[n] = 0.995 a_c[n-1] + 0.005 |x_c[n]|, \tag{5}$$

and use the crossing threshold

$$T_c[n] = \mathrm{clip}\big(0.25 a_c[n],\ T_0/10,\ 3T_0\big). \tag{6}$$

The lower bound prevents the hysteresis from collapsing when the input is weak, and the upper bound prevents a large recent value from making the detector unresponsive. If $|x_c[n] - z_c[n]| > 10 a_c[n]$, the adaptive front end updates $z_c[n]$ but suppresses that sample as an impulse. This is a causal guard against an isolated extreme sample being interpreted as a valid crossing.

2.4. Bit extraction, lookup, and confirmation

At event k, the magnitude m_k is the distance between the sampled primary channel and its running baseline. The symbol reference is

$$\theta_k = \left(1 - \frac{1}{48}\right)\theta_{k-1} + \frac{1}{48} m_k, \tag{7}$$

and every policy uses the same hard bit

$$\hat{b}_k = \mathbb{1}[m_k > \theta_k]. \tag{8}$$

The bit history is advanced forward or backward according to stereo polarity. A mismatch to the expected LFSR progression replaces the expected state by the observed word and resets a validity counter. A lookup position is exposed after more than 24 consecutive progression matches. In other words, a new absolute lock requires 25 valid transitions.

An important camera-ready correction follows from auditing the supplied prototype. Its original ambiguity-band rule selected 1 above the reference, 0 below it, and fell back to the same hard comparison inside the band. It was therefore algebraically identical to Eq. (8). The middle policy is consequently described and named according to the behavior it actually implements: adaptive crossing detection and impulse suppression, not an unimplemented soft-output symbol decoder.

3. RECOVERY POLICIES

Table 1 gives a high-level view before the individual policies are described. The shared bit decision and lookup make the comparison narrow by design.

3.1. Fixed-Threshold Symbol Decoder

The Fixed-Threshold Symbol Decoder (FTSD) uses T_0 and Eq. (8). It assumes that the fixed crossing margin is adequate and that each available event can be reduced immediately to a binary observation. The deterministic progression requirement still supplies memory at the state level, but the crossing detector itself does not adapt to the recent signal magnitude.

This policy is deliberately simple. It provides the clean-signal operational reference and a baseline for judging whether additional front-end or temporal structure is useful. Its expected failure mode is abrupt loss of confirmation: noise or level change creates an extra crossing or removes a true one, the observed word no longer follows the expected LFSR transition, and the 25-transition streak must be rebuilt.

3.2. Adaptive-Threshold Symbol Decoder

The Adaptive-Threshold Symbol Decoder (ATSD) replaces the fixed crossing threshold by Eq. (6) and enables the impulse guard. It keeps Eq. (8), the lookup table, and the 25-transition lock condition. It therefore tests a modest and practical question: can a recent amplitude estimate preserve event timing when the fixed hysteresis margin becomes poorly scaled?

ATSD does not add buffering or future context. The envelope and threshold each require one state value per channel, so the update remains causal and constant-cost. It is expected to help when the waveform is present but the crossing margin is corrupted. It cannot reconstruct events removed by a zero-valued dropout, and it should not be interpreted as a full soft-decision or trellis decoder [10].

Table 1: Policy assumptions, strengths, and limitations.

Policy	Assumption	Strength	Limitation
FTSD	A fixed crossing margin is adequate.	Lowest front-end complexity.	Noise and level change can create or erase crossings.
ATSD	Recent magnitude sets a useful hysteresis scale.	Rejects weak and impulsive crossing errors causally.	Cannot replace genuinely missing events.
SCTD	Recent confirmed position and speed remain useful briefly.	Fills short gaps while preserving a confirmation flag.	Prediction is not new evidence and can be wrong until its finite horizon expires.

3.3. State-Constrained Temporal Decoder

The State-Constrained Temporal Decoder (SCTD) uses the ATSD front end and adds a bounded position-level predictor. A lightweight α–β filter estimates signed playback speed $\hat{v}$ from crossing timing [12]. The fixed gains are $\alpha_p = 1/512$ and $\beta_p = \alpha_p/256$. Each detected quarter-cycle contributes a signed displacement of $1/(4 \times 1000)$ s to the filter, while samples without a crossing contribute zero displacement.

A lookup candidate is accepted if it lies within $J = 64$ position units of the previous confirmed state. With h seconds since the last confirmation, a missing or rejected lookup may be replaced by

$$\hat{x}_{\text{pred}} = \hat{x}_{\text{last}} + 1000\hat{v}h, \tag{9}$$

for $0 < h < H$, where $H = 30$ ms. Predicted outputs are marked unconfirmed. They contribute to total availability but not confirmed availability. After H, prediction stops and the ordinary 25-transition rule may establish a fresh absolute lock.

The horizon is measured from the last confirmed lookup, not from intervening raw crossings. A candidate outside the transition gate is not exposed as an output. These details prevent corrupted bit activity from extending the hold indefinitely and prevent a rejected absolute candidate from being mistaken for confirmed evidence. The resulting behavior occupies a middle ground between a purely reactive decoder and a full probabilistic tracker: recent motion can bridge a short gap, but it cannot create evidence or support unbounded integration.

3.4. Causality and computational cost

All three policies process one stereo frame at a time and use no future samples. Baseline, envelope, symbol-reference, and temporal updates require constant work per frame. Lookup is a direct map operation after the profile table has been built. For a profile with M valid states, table construction and storage are $\mathcal{O}(M)$ offline, while streaming updates and lookup are $\mathcal{O}(1)$. No decoder parameter is retuned between degradation conditions. The experiment therefore compares one fixed implementation of each policy rather than a collection of condition-specific optima.

4. EVALUATION FRAMEWORK

The goal is not waveform fidelity but reliable state recovery. The framework asks three questions: (i) *availability*—does the decoder emit a state?; (ii) *reference-relative correctness*—does it agree with the clean operational reference?; and (iii) *temporal behavior*—is the state stable, and how quickly is confirmation recovered? The same groups can be applied to another deterministic profile once a reference, position unit, and error tolerance are specified.

4.1. Operational reference and availability

FTSD is applied to the clean segment to produce the operational reference $x_k^{(r)}$. This is not independently instrumented physical ground truth. It is suitable for a matched corruption study because every policy is compared at the same indices against the same clean decoder trajectory, but it cannot establish absolute accuracy of that trajectory.

Let $\mathcal{R}$ be indices with a reference, $\mathcal{O} \subseteq \mathcal{R}$ those with any decoder output, and $\mathcal{C} \subseteq \mathcal{O}$ those with a confirmed lookup rather than a prediction. Total availability and confirmed-lock rate are

$$L = \frac{|\mathcal{O}|}{|\mathcal{R}|}, \qquad L_c = \frac{|\mathcal{C}|}{|\mathcal{R}|}. \tag{10}$$

The distinction matters only when a policy can emit an explicitly marked prediction. Reporting L without L_c would make predicted continuity appear equivalent to new profile evidence.

4.2. Correctness and false lock

The mean absolute error over emitted states is

$$E = \frac{1}{|\mathcal{O}|} \sum_{k \in \mathcal{O}} |\hat{x}_k - x_k^{(r)}|. \tag{11}$$

At nominal speed, one position unit corresponds to 1 ms. With $\delta = 20$ position units, the false-lock rate is

$$F = \frac{1}{|\mathcal{O}|} \sum_{k \in \mathcal{O}} \mathbb{1}\big[|\hat{x}_k - x_k^{(r)}| > \delta\big]. \tag{12}$$

A decoder that emits nothing is unavailable, not error-free. If $\mathcal{O}$ is empty, E and F are therefore undefined rather than reported as zero. This convention is especially important at severe noise or clipping levels where every policy can fail to expose a state.

4.3. Stability, score, and reacquisition

With $\sigma_{\Delta\hat{x}}$ the standard deviation of successive available position increments, stability is

$$S = \frac{1}{1 + \sigma_{\Delta\hat{x}}}. \tag{13}$$

The statistic rewards a smooth recovered trajectory, but it does not by itself imply correctness. A stable wrong state remains a false lock under Eq. (12).

The normalized error is the complete transformation

$$\widetilde{E} = \mathrm{clip}(E/E_{\mathrm{sat}}, 0, 1), \qquad E_{\mathrm{sat}} = 1000 \text{ positions}. \quad (14)$$

Because E and S are conditional on output, their contributions are gated by availability:

$$Q = L\left[0.5 + 0.3(1 - \widetilde{E}) + 0.2S\right]. \quad (15)$$

The weights are fixed and not optimized. The score favors availability, then reference-relative error, with stability as a secondary preference. It is a compact plot variable rather than a safety metric; F, L_c, and reacquisition remain separate.

Reacquisition time is measured from the end of a known isolated dropout to the first subsequent confirmed output satisfying the same tolerance δ. A temporal prediction does not end this interval because the metric asks when new profile-confirmed evidence has returned.

5. EXPERIMENTAL SETUP

5.1. Signal, fixed parameters, and reference

The first 10.0 s of a 44.1-kHz, 16-bit stereo commercial DVS waveform are used. The segment has a two-channel RMS amplitude of 0.3081 full scale (FS) and peak magnitude of 0.4948 FS. Both channels are processed without downmixing. The same clean segment, profile lookup, and decoder constants are used for every condition.

Table 2 collects the values that determine the reported behavior. They are fixed before the sweeps. In particular, the temporal horizon and transition gate are not increased for longer dropouts, and the adaptive threshold is not retuned for higher noise.

The operational reference is obtained by decoding this segment with clean FTSD. Metrics are accumulated only where that reference is available. This avoids counting startup indices without a clean lock, while retaining the limitation that the reference comes from the same decoder family rather than an instrumented physical position.

5.2. Controlled degradation operators

Three degradations are applied independently. Let R_x denote the clean two-channel RMS and let $\mathbf{u}[n]$ contain independent unit-variance Gaussian samples. The white-noise condition is

$$\mathbf{y}_\eta[n] = \mathbf{x}[n] + \eta R_x \mathbf{u}[n], \quad (16)$$

with $\eta \in \{0, .03, .06, .10, .16, .26\}$. For nonzero η, these ratios correspond to nominal signal-to-noise ratios from about 30.5 to 11.7 dB. Independent noise is generated for the left and right channels.

Repeated dropout sets both channels to zero for B consecutive samples. At each eligible sample, a non-overlapping burst begins with probability 5×10^{-4}; $B \in \{0, 16, 32, 64, 128, 256\}$. The nonzero lengths span 0.36–5.80 ms, or 0.36–5.80 nominal decision events by Eq. (3). This sweep measures steady-state behavior under many interruptions rather than recovery from a single known event.

Hard clipping is applied independently to each sample and channel as

$$y_c[n] = \mathrm{sgn}(x_c[n]) \min\big(|x_c[n]|, a(2^{15} - 1)\big). \quad (17)$$

The thresholds include $a \in \{1, .9, .7, .6, .5\}$, increments of .025 from .475 to .30, and .25 and .20. The fine spacing around the observed transition is included to distinguish a shared acquisition boundary from an apparent overlap caused by coarse sampling.

5.3. Replication and output data

Noise and repeated dropout are each evaluated with seeds 1234–1238. Clipping is deterministic and is evaluated once per threshold. A separate isolated-dropout experiment inserts one burst at five positions evenly distributed through the segment for each value of B. The resulting design contains, per policy, 30 white-noise runs, 30 repeated-dropout runs, 15 clipping runs, and 30 isolated-reacquisition runs. Across three policies, the camera-ready dataset therefore contains 315 decoder-condition runs.

Each run records L, L_c, E, F, S, and Q. Isolated-dropout runs additionally record confirmed reacquisition time. Figures show the mean and one sample standard deviation for replicated conditions. No condition combines noise, dropout, and clipping. Hum, wow/flutter, groove wear, channel imbalance, stylus contamination, and hardware recapture remain outside the experiment.

6. RESULTS

6.1. Clean operation and additive noise

All three policies are essentially coincident on the clean segment. FTSD gives $L = L_c = 1.000$, $E = 0$, $F = 0$, $S = 0.870$, and $Q = 0.974$. ATSD and SCTD give $L = L_c = 0.999998$ and differ from the reference by only 0.0031 position units on average. The clean score is below one because S is a measured increment-stability transform rather than a normalization forced to equal one.

The left column of Fig. 1 shows the component metrics under additive noise. At $\eta = 0.03$, every policy remains continuously available and reference-consistent. At $\eta = 0.06$, FTSD confirmed availability falls to 0.868 ± 0.010, while ATSD reaches 0.936 ± 0.004. The corresponding scores are 0.819 ± 0.011 and 0.893 ± 0.005. This is the range in which a recent amplitude estimate improves crossing timing without requiring missing events to be invented.

SCTD uses the same confirmed evidence as ATSD but emits bounded predictions during gaps. At $\eta = 0.06$, its total availability is 0.981 ± 0.004, confirmed availability remains 0.936 ± 0.004, and $Q = 0.952 \pm 0.006$. The separation between L and L_c is therefore a direct measure of continuity supplied by the hold rather than by new lookup evidence.

At $\eta = 0.10$, the signal structure is largely lost. FTSD gives $L = 0.032 \pm 0.009$, ATSD gives $L = 0.103 \pm 0.015$, and SCTD gives $L = 0.286 \pm 0.013$ with only 0.103 ± 0.015 confirmed. Mean reference-relative errors remain below one position unit for the sparse emitted states, and $F = 0$ at the 20-position tolerance. This does not imply successful decoding: availability and stability have already collapsed. At the two highest noise levels, no policy emits a state, so E and F are undefined and appear blank.

6.2. Repeated dropout

Repeated zero-valued bursts produce a different failure mode because events are removed rather than merely perturbed. The center column of Fig. 1 shows that FTSD and ATSD have similar confirmed behavior once the burst length reaches 64 samples. At $B = 32$, FTSD gives $L = 0.449 \pm 0.015$ and ATSD gives $L = 0.428 \pm 0.019$; the small difference is not evidence of a

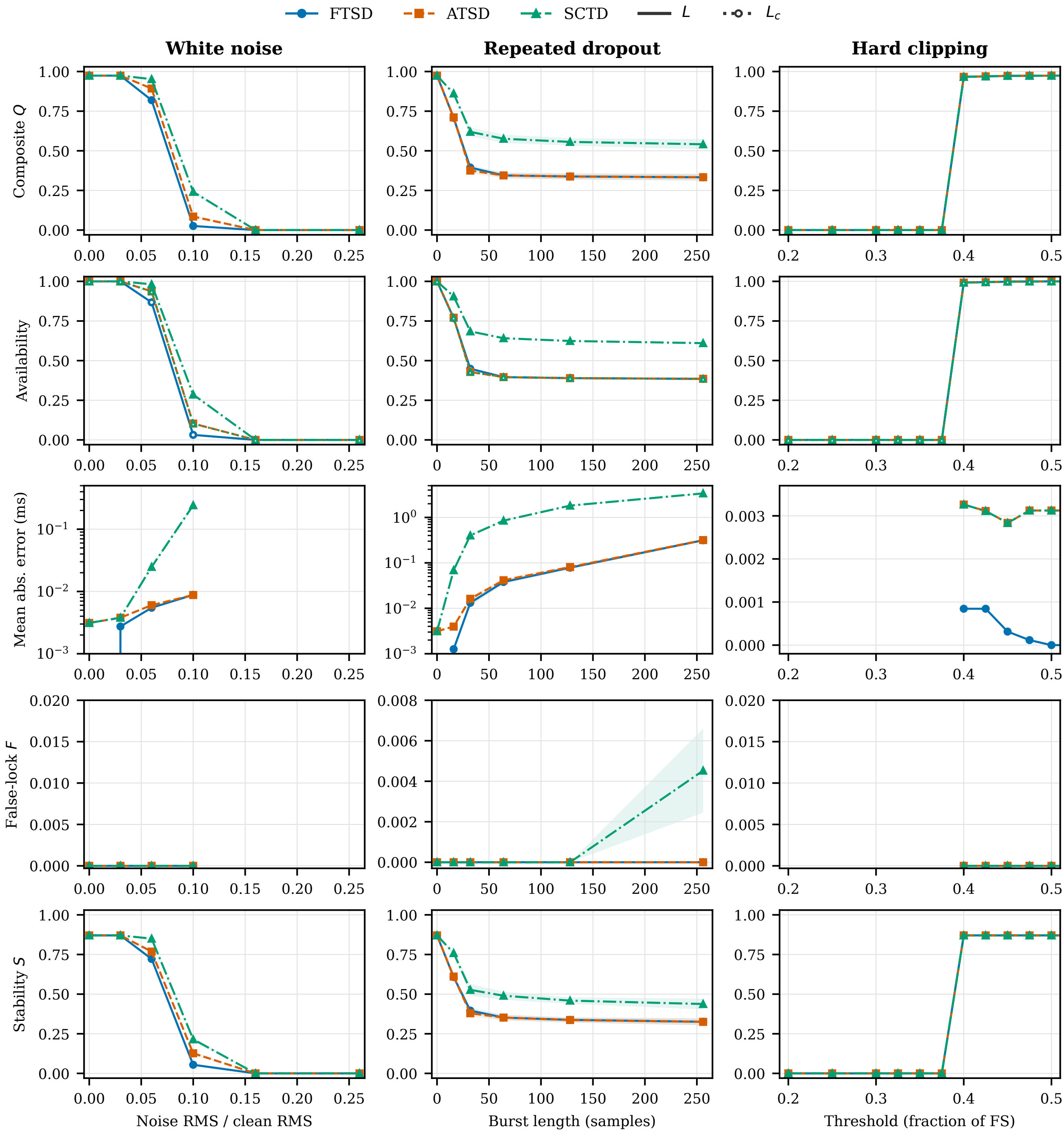

Figure 1: Component results for white noise (left), repeated dropout (center), and hard clipping (right). Rows show Q, availability, mean absolute error, false-lock rate, and stability. In the availability row, filled curves are total availability L and open dotted curves are confirmed availability L_c. Noise and dropout curves show five-run means with one-standard-deviation shading. Error uses a logarithmic scale in the first two columns. Clipping is zoomed to 0.20–0.50 FS; values above 0.50 remain on the clean plateau. Undefined E and F are blank when no output exists.

Table 2: Fixed signal, decoder, and evaluation parameters used in every experiment.

Quantity	Value	Role
Sampling and profile rate	44100 samples/s; 1000 positions/s	Establishes 44.1 samples per nominal decision event.
Register and valid trajectory	20 bits; 950000 states	Defines the precomputed absolute-position lookup.
Baseline and fixed crossing	$\tau_z = 1$ ms; $T_0 = 128$ PCM units	Removes slow offset and sets FTSD hysteresis.
Adaptive crossing envelope	$0.995/0.005$ recursion; $0.25a_c$ clipped to $[T_0/10, 3T_0]$	Sets ATSD/SCTD hysteresis from recent magnitude.
Impulse guard and symbol reference	$10a_c$; 48-event exponential average	Rejects isolated extremes and sets the common bit threshold.
Absolute lock requirement	25 valid LFSR transitions	Prevents a single lookup hit from declaring lock.
Temporal filter and gate	$\alpha_p = 1/512$, $\beta_p = \alpha_p/256$; $J = 64$ positions	Estimates signed speed and rejects distant candidates during a hold.
Temporal horizon	$H = 30$ ms	Limits how long unconfirmed prediction may be emitted.
False-lock and score scales	$\delta = 20$ positions; $E_{\text{sat}} = 1000$ positions	Defines reference inconsistency and normalized error.
Random replication	Seeds 1234–1238	Supplies five fixed replicates for stochastic sweeps.

general fixed-threshold advantage, but it shows that adaptive hysteresis is not automatically useful when the observation is absent.

At $B = 64$, ATSD gives $L = L_c = 0.396 \pm 0.019$, whereas SCTD gives $L = 0.641 \pm 0.025$ with the same L_c. Its mean error rises from 0.041 ± 0.004 to 0.851 ± 0.024 position units because predicted samples are being compared with the operational reference, but the value remains far below the 20-position false-lock tolerance. The gain is therefore short-gap continuity with an explicit confirmation flag, not a change in the evidence needed to re-establish absolute state.

As B increases from 128 to 256 samples, ATSD confirmed availability stays near 0.39. SCTD total availability decreases from 0.624 ± 0.024 to 0.610 ± 0.032 because the 30-ms horizon is fixed. At $B = 256$, its mean error is 3.381 ± 0.126 position units and its reference-relative false-lock rate is 0.0045 ± 0.0021. The nonzero false-lock rate is small, but it is the clearest measured cost of continuing to emit through the longest repeated gaps.

6.3. Isolated dropout and reacquisition

The repeated-burst sweep mixes loss, prediction, and repeated lock rebuilding throughout a run. The isolated experiment asks a narrower question: after one known dropout ends, how long does the decoder take to return to a confirmed, reference-consistent state? Figure 2 reports the result over five insertion positions.

Mean confirmed reacquisition spans approximately 17–36 ms with standard deviations of roughly 20–25 ms. Insertion phase contributes more variation than a monotonic dependence on these short burst lengths. ATSD and SCTD have the same confirmed reacquisition for a given run because SCTD prediction is deliberately not counted as confirmation. For example, at $B = 64$ both give 17.5 ± 24.0 ms; at $B = 128$ both give 35.2 ± 19.7 ms. SCTD still supplies higher total availability during the interval, but it does not claim that prediction has reacquired the absolute code.

6.4. Hard clipping

The right column of Fig. 1 resolves the previously unsampled interval between 0.30 and 0.50 FS. Every policy has $L = 0$ through

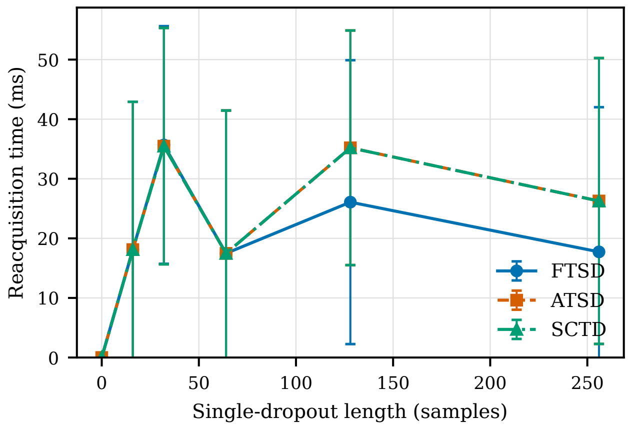

Figure 2: Confirmed reacquisition time after one isolated dropout. Points show the mean over five insertion positions and bars show one sample standard deviation. Prediction does not count as confirmation.

0.375 FS and $L = 0.9922$ at 0.400 FS. At the latter threshold, FTSD gives $E = 0.00084$ position units and ATSD/SCTD give $E = 0.00326$; all three have $F = 0$, $S = 0.870$, and $Q = 0.967$ to the shown precision.

The transition is therefore shared by the three policies. Once crossings and the 20-bit progression are recoverable, they remain nearly coincident and return to the clean plateau at and above 0.50 FS. This corrects the interpretation suggested by the earlier coarse sweep: the data do not support a meaningful temporal or adaptive advantage under severe clipping. In this segment, clipping removes the amplitude relationship used by the common bit extractor, so a later state policy cannot recover an absolute word that was never acquired.

7. DISCUSSION

The results support three main conclusions. First, a fixed crossing margin is sufficient when the waveform is clean and the decision events are clearly separated. The additional policies do not improve a condition that is already continuously available and reference-consistent. This matters in a real-time system because a more structured policy should be justified by a specific failure mode rather than by complexity alone.

Second, adaptive crossing hysteresis helps mainly when the waveform is present but the event margin has become unreliable. This is the additive-noise transition near $\eta = 0.06$: ATSD retains more confirmed output than FTSD without changing the common bit rule or lock requirement. Once an event is genuinely absent, however, the adaptive threshold has nothing to classify. Its confirmed-dropout curves therefore converge with the fixed baseline.

Third, bounded temporal hold is the most effective way to preserve short-gap availability, but it changes the meaning of an output. SCTD trusts the recent trajectory as well as the current lookup evidence. This improves continuity after an established lock, while L_c and the confirmation flag preserve the distinction between an estimate and a verified profile state. The finite horizon is essential: without it, plausible motion could persist indefinitely through corruption. Systems using this approach should expose the confirmed/predicted flag to downstream audio control and should decide explicitly whether an unconfirmed state is preferable to silence.

These three policies can be viewed as different answers to the question "what should be trusted?" FTSD trusts a fixed event margin, ATSD trusts a margin scaled by recent magnitude, and SCTD additionally trusts a recent trajectory for a limited time. None is universally strongest. The useful policy depends on whether the dominant corruption is stochastic, missing-data, or a structured deformation of the shared waveform features.

For the motivating DVS application, white noise approximates loss of decision margin, repeated zeros approximate intermittent contact loss, and clipping approximates excessive input gain. These mechanisms are useful for diagnosis because they isolate different parts of the decoder. They are not substitutes for a cartridge, phono stage, worn control medium, realistic forward/reverse gesture, or a combination of faults. In particular, clipping and dropout results should not be read as predictions for stylus wear or wow/flutter, which change timing and channel relationships in different ways.

7.1. Validity limits

Four limitations constrain the claims. First, the clean FTSD trajectory is an operational reference rather than independent physical ground truth. Absolute position accuracy is therefore not established. A stronger experiment would compare decoder output with an encoder log or instrumented platter motion.

Second, one 10-s segment and one profile cannot establish generalization across records, decks, speeds, directions, or gestures. The present segment is useful for a matched study, but a deployment benchmark should contain several profiles and repeated sections at different speeds and directions.

Third, the synthetic conditions are tested in isolation and omit important hardware effects and interactions. Hardware recaptures should include channel imbalance, cartridge and phono-stage response, stylus contamination, groove wear, hum, impulsive debris, wow/flutter, and combined faults. These additions are more important than extending the current synthetic curves to more decimal places.

Fourth, the score weights were not subjected to sensitivity analysis. Equation (15) prevents a rare smooth output from receiving a high score, but a different application may value false-lock avoidance more heavily than continuity. The component curves should therefore carry more interpretive weight than small differences in Q, and F should remain visible rather than being hidden inside a single aggregate.

8. CONCLUSION

A structured comparison of three real-time policies for deterministic DVS timecode recovery was performed. The Fixed-Threshold Symbol Decoder (FTSD) provides the simplest clean-signal baseline. The Adaptive-Threshold Symbol Decoder (ATSD) improves confirmed availability near the additive-noise transition. The State-Constrained Temporal Decoder (SCTD) uses the same confirmed evidence but increases short-gap availability through explicitly marked, finite-horizon prediction. Fine clipping measurements reveal a shared acquisition boundary rather than a material policy ranking. The central recommendation is to evaluate availability, confirmation, reference-relative error, false lock, stability, and reacquisition separately, and to validate future systems against independent physical-position evidence.

9. REFERENCES

[1] F. Heller and J. Borchers, "Visualizing song structure on timecode vinyls," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, London, UK, 2014, pp. 66–69.

[2] J. Holthuis, "How does timecode vinyl actually work? (pt. 1)," Mixxx Development Team, Nov. 2021, accessed: 2026-07-14. [Online]. Available: https://mixxx.org/news/2021-11-21-dvs-internals-pt1/

[3] ——, "How does timecode vinyl actually work? (pt. 2)," Mixxx Development Team, Dec. 2021, accessed: 2026-07-14. [Online]. Available: https://mixxx.org/news/2021-12-22-dvs-internals-pt2/

[4] M. Hills and xwax contributors, "xwax: Digital vinyl system for linux," Open-source software repository, 2026, accessed: 2026-07-14. [Online]. Available: https://github.com/xwax/xwax

[5] J. Claußen, "How does timecode vinyl actually work? (pt. 3)," Mixxx Development Team, Aug. 2025, accessed: 2026-07-14. [Online]. Available: https://mixxx.org/news/2025-08-27-dvs-internals-pt3/

[6] U. Zölzer, Ed., *DAFX: Digital Audio Effects*, 2nd ed. Chichester, UK: John Wiley & Sons, 2011.

[7] P. Meier, S. Schwär, and M. Müller, "A real-time approach for estimating pulse tracking parameters for beat-synchronous audio effects," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Guildford, Surrey, UK, 2024, pp. 314–321.

[8] J. O. Smith III, *Physical Audio Signal Processing: For Virtual Musical Instruments and Audio Effects*. W3K Publishing, 2010.

[9] J. G. Proakis, *Digital Communications*, 4th ed. New York, NY, USA: McGraw-Hill, 2001.

[10] G. D. Forney, Jr., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[12] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: John Wiley & Sons, 2006.