Drift–Slew Fusion Bootstrap: A Deterministic Residual-Based State Correction Framework

Riaan de Beer
predictiverendezvous@proton.me
Independent Researcher
ORCID: 0009-0006-1155-027X
DOI: 10.5281/zenodo.18706455
https://github.com/infinityabundance/dsfb
Version 1.0

2026

Abstract

This document defines the architectural specification and operational framework for the Drift–Slew Fusion Bootstrap (DSFB) repository and associated reference implementation. It presupposes the foundational theoretical constructs introduced in prior work on slew-aware trust- adaptive state estimation and trust-adaptive multi-diagnostic weighting in state reconstruction.

The framework formalizes deterministic disturbance assumptions, canonical residual decomposition into drift and slew components, and the structure of bootstrap correction updates as encoded in the reference codebase. It specifies causal separation operators, residual envelopes, update semantics, and trust-weighting conventions in a manner that is fully self-contained and reproducible.

This document does not introduce new theoretical results. Its purpose is to provide a precise, implementation-aligned reference for reproducible usage, architectural coherence, and consistent deterministic integration of DSFB across applications. Implementation notes, parameter conventions, and verification tests are included to ensure alignment between the mathematical framework and software practice.

1 Introduction

The Drift–Slew Fusion Bootstrap (DSFB) mechanism was introduced in prior work as a deterministic, trust-adaptive residual correction framework for multi-sensor state estimation under bounded disturbance. In particular, the formulation was applied to slew-aware nonlinear estimation in oscillatory systems and to trust-adaptive multi-diagnostic weighting in magnetically confined plasma reconstruction. Those documents established the conceptual mechanism and presented boundedness results under explicit deterministic assumptions.

The present document serves a different purpose. It provides a formal architectural and operational specification for the DSFB repository and reference implementation. Rather than introducing new theoretical results, this paper consolidates modeling assumptions, defines canonical notation, and specifies the residual decomposition and bootstrap update semantics as they are encoded in the implementation.

A recurring source of ambiguity in multi-sensor correction schemes is the conflation of conceptual mechanisms, mathematical guarantees, and software realization. This document separates those layers explicitly:

- **Conceptual Mechanism:** decomposition of residuals into drift and slew components and application of a causal bootstrap correction.
- **Deterministic Conditions:** bounded disturbance assumptions and sufficient conditions for bounded residual propagation.
- **Implementation Layer:** operational definitions, parameter conventions, update ordering, and reproducibility guidelines.

Throughout this document, disturbances are modeled deterministically as bounded, piecewise continuous signals. No probabilistic noise model, covariance structure, or distributional assumption is required. The framework is therefore compatible with applications where disturbance statistics are unknown, unreliable, or intentionally avoided.

The objective of this paper is to establish a stable and self-contained reference specification for DSFB usage. It is intended to ensure architectural coherence across applications, reproducibility of numerical results, and consistent interpretation of deterministic assumptions within the broader DSFB technical series.

We use the term *slew* to denote abrupt residual increments that exceed a user-defined rate threshold, in contrast to slowly accumulating *drift*. In this sense, drift captures low-frequency residual structure, while slew captures rapid deviations that trigger trust attenuation and bounded corrective responses.

2 Deterministic Disturbance Model

This section specifies the disturbance model assumed by the DSFB reference implementation[1]. The formulation is deterministic and directly corresponds to the signal classes handled by the repository.

2.1 State and Measurement Structure

Let the system state be

$$x_k \in \mathbb{R}^n$$

evolving in discrete time. Measurement channels are defined as

$$y_{i,k} = h_i(x_k) + d_{i,k}, \quad i = 1, \ldots, m,$$

where:

- $h_i(\cdot)$ is the user-defined measurement mapping,
- $d_{i,k}$ is a deterministic disturbance sequence.

The repository operates in discrete time and assumes no probabilistic structure for $d_{i,k}$.

[1] `https://github.com/infinityabundance/dsfb`

2.2 Admissible Disturbance Class

The implementation is designed to operate under the following signal constraints:

Assumption 1 (Bounded Amplitude)**.** *For each channel i, there exists $\delta_i > 0$ such that*

$$\|d_{i,k}\| \le \delta_i \quad \forall k.$$

Assumption 2 (Finite Increment Bound)**.** *There exists $\sigma_i > 0$ such that*

$$\|d_{i,k+1} - d_{i,k}\| \le \sigma_i.$$

This increment bound enables separation between slowly accumulating drift and abrupt deviations ("slew") using residual rate thresholds implemented in the codebase.

2.3 Residual Computation in the Repository

Given a state estimate $\hat{x}_k$, the residual used by DSFB is

$$r_{i,k} = y_{i,k} - h_i(\hat{x}_k).$$

In the repository, residuals are stored in rolling buffers of fixed length. Drift and slew components are computed from these buffers using:

- magnitude thresholds,
- rate-of-change thresholds,
- windowed averaging or smoothing operators.

The disturbance assumptions above correspond directly to the tuning parameters exposed in the implementation (e.g., drift thresholds, slew rate limits, buffer window sizes).

2.4 Scope of the Model

The DSFB repository does not assume:

- Gaussian noise,
- covariance matrices,
- stochastic independence between channels,
- optimality with respect to any probabilistic cost functional.

The framework is therefore intended for applications where disturbance statistics are unknown, unreliable, or intentionally avoided, and where bounded deterministic behavior is a sufficient modeling assumption.

Subsequent sections define the canonical drift–slew decomposition and bootstrap update semantics consistent with this signal class.

Deterministic Modeling Convention

Throughout this document, disturbances and residuals are modeled as deterministic signals. In particular, the DSFB framework:

- does not assume Gaussian noise,
- does not require covariance matrices,
- does not assume stochastic independence between channels,
- does not derive gains from probabilistic optimality criteria.

All guarantees and operating envelopes are stated with respect to the bounded deterministic disturbance model of Section 2.2.

3 Drift–Slew Decomposition Operator

This section specifies the canonical residual decomposition procedure implemented in the DSFB repository. The purpose of the operator is to separate slowly accumulating residual structure ("drift") from abrupt deviation components ("slew") in a causal, discrete-time setting.

3.1 Residual Buffering

For each measurement channel i, residuals

$$r_{i,k} = y_{i,k} - h_i(\hat{x}_k)$$

are stored in a rolling buffer of fixed length W:

$$\mathcal{R}_{i,k} = \{r_{i,k-W+1}, \ldots, r_{i,k}\}.$$

All decomposition operations are causal and depend only on values contained in $\mathcal{R}_{i,k}$.

3.2 Increment Metric

The repository defines a discrete increment magnitude

$$\Delta r_{i,k} = \|r_{i,k} - r_{i,k-1}\|.$$

This quantity is used to detect abrupt deviations relative to a user-defined slew threshold τ_i.

3.3 Slew Detection Rule

A residual sample is classified as exhibiting slew behavior if

$$\Delta r_{i,k} > \tau_i.$$

The threshold τ_i corresponds to the assumed finite increment bound of admissible disturbances and is exposed as a tunable parameter.

When this condition is met, the current residual is flagged as containing a slew component.

3.4 Drift Component Estimation

The drift component is estimated using a smoothing operator applied to the residual buffer. In the reference implementation, this takes the form of a windowed averaging or low-pass smoothing operator:

$$r_{i,k}^{(\mathrm{drift})} = \mathcal{S}_W(\mathcal{R}_{i,k}),$$

where $\mathcal{S}_W(\cdot)$ denotes a causal smoothing functional defined over the buffer window of length W.

This operator is required to satisfy:

- Causality,
- Bounded gain with respect to bounded residual inputs,
- Monotonic attenuation of high-frequency components.

The specific smoothing method (e.g., moving average, exponential smoothing) is configurable but must satisfy these structural properties.

3.5 Slew Component Definition

The slew component is defined implicitly as the deviation between the instantaneous residual and its drift estimate:

$$r_{i,k}^{(\mathrm{slew})} = r_{i,k} - r_{i,k}^{(\mathrm{drift})}.$$

This decomposition is algebraic and does not assume orthogonality or spectral separation. It is an operational partition consistent with the bounded increment assumption defined in Section 2.

3.6 Implementation Semantics

The decomposition procedure is:

1. Compute residual $r_{i,k}$.
2. Update buffer $\mathcal{R}_{i,k}$.
3. Compute increment magnitude $\Delta r_{i,k}$.
4. Evaluate slew threshold condition.
5. Compute drift estimate via smoothing operator.
6. Define slew component as residual minus drift.

All operations are channel-local. No cross-channel coupling is introduced at the decomposition stage.

3.7 Scope

The drift–slew decomposition is not intended to represent a spectral or optimal filtering separation. It is a causal structural partition aligned with deterministic bounded-increment disturbances. Its purpose is to provide interpretable residual structure for the subsequent bootstrap correction operator.

3.8 Reference Implementation Defaults

While the DSFB framework permits different residual processing operators, the reference implementation adopts the following canonical choices.

Drift Smoothing Operator. The smoothing functional $S_W(\cdot)$ is implemented in the reference implementation as an exponential moving average (EMA):

$$r_{i,k}^{(\text{drift})} = \rho\, r_{i,k-1}^{(\text{drift})} + (1-\rho)\, r_{i,k}, \tag{1}$$

where $0 < \rho < 1$ is the smoothing coefficient. The default value $\rho = 0.95$ is used in the repository. The parameter ρ is configurable and controls the effective memory length of the drift estimate.

Equation (1) is used throughout the remainder of this document when referring to the canonical EMA-based drift estimator.

Slew Residual Component. The slew component remains defined algebraically as

$$r_{i,k}^{(\text{slew})} = r_{i,k} - r_{i,k}^{(\text{drift})}, \tag{2}$$

with classification into drift- or slew-dominant behavior governed by the increment threshold τ_i in Section 3.3.

4 Bootstrap Correction Operator

This section specifies the canonical bootstrap correction mapping used in the DSFB repository. The operator acts on decomposed residual components to produce a deterministic state correction update.

4.1 State Estimate Update Structure

Let $\hat{x}_k$ denote the current state estimate. The DSFB update produces

$$\hat{x}_{k+1} = \hat{x}_k + u_k,$$

where u_k is the aggregate correction computed from channel-local residual structure.

4.2 Channel-Local Correction Term

For each measurement channel i, a local correction contribution is defined as

$$u_{i,k} = G_{i,k}\, \psi_i\big(r_{i,k}^{(\text{drift})}, r_{i,k}^{(\text{slew})}\big),$$

where:

- $G_{i,k}$ is a user-defined correction gain or mapping,
- $\psi_i(\cdot)$ is a residual-to-correction transformation,
- $r_{i,k}^{(\mathrm{drift})}$ and $r_{i,k}^{(\mathrm{slew})}$ are defined in Section 3.

The transformation $\psi_i(\cdot)$ is implementation-defined but must satisfy bounded-input bounded-output (BIBO) behavior under bounded residuals.

4.3 Trust Weighting

Each channel contribution is modulated by a deterministic trust weight $w_{i,k} \in [0, 1]$:

$$\tilde{u}_{i,k} = w_{i,k}\, u_{i,k}.$$

Trust weights are updated using residual magnitude and slew detection conditions. The repository implements trust attenuation under sustained slew behavior or excessive residual growth.

Trust update rules are:

- Causal,
- Bounded in $[0, 1]$,
- Monotone decreasing under repeated threshold violations,
- Recoverable under residual normalization.

Exact update equations are configurable but must preserve these structural constraints.

4.4 Aggregate Update

The full bootstrap correction is the sum of weighted channel contributions:

$$u_k = \sum_{i=1}^{m} \tilde{u}_{i,k}.$$

Thus,

$$\hat{x}_{k+1} = \hat{x}_k + \sum_{i=1}^{m} w_{i,k} G_{i,k} \psi_i\big(r_{i,k}^{(\mathrm{drift})}, r_{i,k}^{(\mathrm{slew})}\big).$$

No global optimization problem is solved at this stage. The update is purely constructive and deterministic.

4.5 Operator Constraints

The repository requires that:

- Correction gains remain bounded,
- Trust weights remain bounded in $[0, 1]$,
- The aggregate update remains finite under bounded residuals,

- The operator remains causal.

These conditions ensure bounded correction magnitude under the disturbance assumptions of Section 2.

4.6 Scope and Non-Claims

The bootstrap operator:

- Does not assume covariance matrices,
- Does not compute Kalman gains,
- Does not solve a minimization problem,
- Does not assert stochastic optimality.

Its purpose is to provide a deterministic residual-responsive correction mechanism aligned with bounded disturbance modeling and structural trust adaptation.

4.7 Canonical Residual Transform and Trust Update

In the reference implementation, the residual-to-correction mapping $\psi_i(\cdot)$ and trust update rule are instantiated in a simple, bounded form.

Residual Transform. A canonical choice used in the repository is

$$\psi_i\big(r_{i,k}^{(\mathrm{drift})}, r_{i,k}^{(\mathrm{slew})}\big) = r_{i,k}^{(\mathrm{drift})} + r_{i,k}^{(\mathrm{slew})} = r_{i,k}, \tag{3}$$

i.e., the full residual is mapped into the correction direction through $G_{i,k}$. Alternative transforms can be plugged in provided they remain bounded-input bounded-output under bounded residuals.

Trust Update Rule. Trust weights are updated using a deterministic, threshold-driven rule. A canonical form consistent with the structural constraints of Section 4.3 is

$$w_{i,k+1} = \begin{cases} \max\{0, \beta_i\, w_{i,k}\}, & \Delta r_{i,k} > \tau_i, \\ \min\{1,\, w_{i,k} + \gamma_i\}, & \Delta r_{i,k} \le \tau_i, \end{cases} \tag{4}$$

with $0 < \beta_i < 1$ a decay factor and $\gamma_i > 0$ a recovery increment. This rule is causal, bounded in $[0, 1]$, monotone decreasing under repeated threshold violations, and allows gradual trust recovery when residual increments remain below the slew threshold. The concrete values of β_i and γ_i are user-configurable.

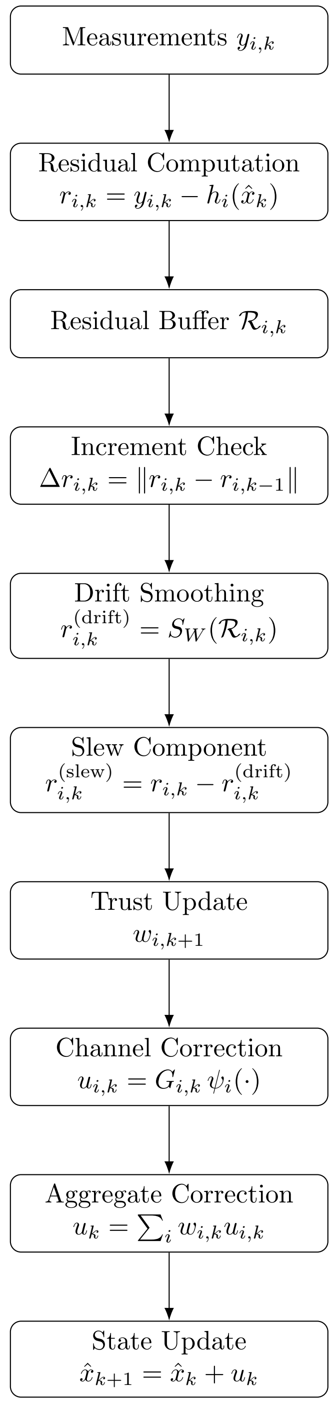

Figure 1: Deterministic DSFB processing pipeline from measurement acquisition to state update in discrete time.

The reference Rust implementation in the dsfb crate follows this exact ordering.

5 Deterministic Boundedness Conditions

This section states the boundedness properties that follow directly from the structural constraints defined in Sections 2–4. No claims of optimality or global asymptotic convergence are made.

5.1 Assumptions

Assume:

Assumption 3 (Bounded Disturbance)**.** *For each channel i, disturbances satisfy*

$$\|d_{i,k}\| \leq \delta_i \quad \forall k.$$

Assumption 4 (Bounded Gains)**.** *For each channel i and time step k,*

$$\|G_{i,k}\| \leq \bar{G}_i.$$

Assumption 5 (Bounded Trust Weights)**.** *Trust weights satisfy*

$$0 \leq w_{i,k} \leq 1.$$

5.2 Residual Boundedness

Under local Lipschitz continuity of the measurement mappings $h_i(\cdot)$, bounded disturbances imply bounded residuals:

$$\|r_{i,k}\| \leq C_i(\delta_i, \|x_k - \hat{x}_k\|).$$

Thus, if the state estimation error remains bounded, residuals remain bounded. This document does not establish closed-loop boundedness of the estimation error; only boundedness of the update operator under bounded residual inputs.

5.3 Bootstrap Update Boundedness

From Section 4, the correction term is

$$u_k = \sum_{i=1}^{m} w_{i,k} G_{i,k} \psi_i(\cdot).$$

Assume that the residual transformation $\psi_i(\cdot)$ is bounded-input bounded-output (BIBO), i.e.,

$$\|\psi_i(r)\| \leq \bar{\psi}_i \quad \text{whenever } \|r\| \leq \bar{r}_i.$$

Then the aggregate correction magnitude satisfies

$$\|u_k\| \leq \sum_{i=1}^{m} \bar{G}_i \, \bar{\psi}_i.$$

Thus, the bootstrap correction is uniformly bounded under bounded residuals and bounded gains.

5.4 State Estimate Bounded Increment Property

The update law

$$\hat{x}_{k+1} = \hat{x}_k + u_k$$

implies that state estimate increments are bounded:

$$\|\hat{x}_{k+1} - \hat{x}_k\| \leq \sum_{i=1}^{m} \bar{G}_i \, \bar{\psi}_i.$$

Therefore, the DSFB operator does not introduce unbounded correction steps under admissible signal and parameter conditions.

5.5 Interpretation

The repository guarantees:

- Bounded correction magnitude under bounded disturbance,
- Bounded incremental state updates,
- No amplification of bounded residual signals beyond finite gain.

The framework does not guarantee:

- Global asymptotic convergence,
- Optimal estimation performance,
- Stability under adversarial unbounded disturbance.

These boundedness conditions define the deterministic operating envelope of the DSFB implementation.

These results do not establish closed-loop asymptotic convergence of the estimation error. Rather, they state that, under the bounded disturbance model of Section 2 and the gain and trust constraints above, the DSFB update operator cannot introduce unbounded growth in the correction term or in the incremental state updates. Closed-loop behavior under specific system dynamics and application regimes must be analyzed separately.

6 Implementation and Reproducibility Conventions

This section defines the operational conventions adopted by the DSFB reference implementation[2]. Its purpose is to ensure alignment between the mathematical framework defined in previous sections and the behavior of the repository.

[2] `https://github.com/infinityabundance/dsfb`

6.1 Discrete-Time Execution Model

The repository operates in discrete time with the following update order at each step k:

1. Acquire measurements $y_{i,k}$.
2. Compute residuals $r_{i,k}$.
3. Update residual buffers $\mathcal{R}_{i,k}$.
4. Perform drift–slew decomposition.
5. Update trust weights $w_{i,k}$.
6. Compute channel-local corrections $u_{i,k}$.
7. Aggregate correction u_k.
8. Update state estimate $\hat{x}_{k+1}$.

This ordering is fixed and must not be altered without re-validating boundedness assumptions.

6.2 Parameter Semantics

The primary exposed parameters correspond directly to the deterministic assumptions defined earlier:

- **Window length W:** controls temporal smoothing scale.
- **Slew threshold τ_i:** bounds acceptable residual increment.
- **Gain bounds $\bar{G}_i$:** limit correction magnitude.
- **Trust attenuation rates:** govern decay under sustained violations.

Parameter tuning must preserve bounded-input bounded-output behavior.

6.3 Numerical Stability Considerations

To ensure stable numerical behavior:

- Residual buffers must use fixed-size memory.
- Gain matrices should be pre-bounded or saturated.
- Trust weights must be clipped to $[0, 1]$.
- Correction terms should avoid cumulative floating-point drift.

No adaptive parameter learning is performed internally. All adaptation is deterministic and threshold-driven.

6.4 Reproducibility Guidelines

To ensure consistent results across environments:

- Fix all thresholds and window sizes explicitly.
- Document gain initialization values.
- Use deterministic test inputs when validating behavior.
- Avoid stochastic simulation unless explicitly isolated.

All examples in this document assume deterministic input sequences consistent with the disturbance model defined in Section 2.

6.5 Extensibility Constraints

Extensions to the repository must preserve:

- Causality of all operators,
- Bounded gains,
- Bounded trust weights,
- Finite update increments.

Modifications that introduce stochastic weighting, optimization-based updates, or unbounded adaptive gains fall outside the deterministic framework defined in this document.

7 Minimal Deterministic Example

This section provides a simple deterministic illustration of the DSFB framework consistent with the disturbance model of Section 2.

7.1 System Setup

Consider a scalar state

$$x_k \in \mathbb{R}$$

with two measurement channels:

$$y_{1,k} = x_k + d_{1,k}, \qquad y_{2,k} = x_k + d_{2,k}.$$

Assume the true state evolves slowly and remains bounded.
Disturbance sequences are defined deterministically as:

- Channel 1: slow linear drift

$$d_{1,k} = \alpha k, \quad \alpha > 0.$$

- Channel 2: bounded disturbance with an abrupt slew event

$$d_{2,k} = \begin{cases} 0, & k < k_s, \\ \Delta, & k \geq k_s. \end{cases}$$

Both disturbances satisfy bounded amplitude and bounded increment conditions for finite α and finite Δ.

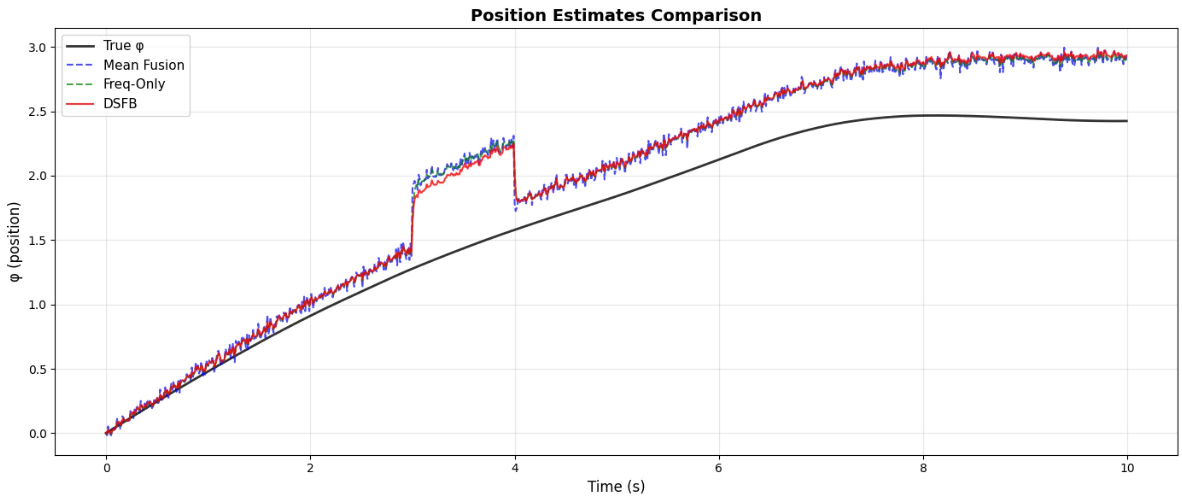

Figure 2: Position estimates from `dsfb_simulation.ipynb`: true state ϕ, mean fusion, frequency-only estimate, and DSFB. The impulse disturbance produces a transient deviation in all methods.

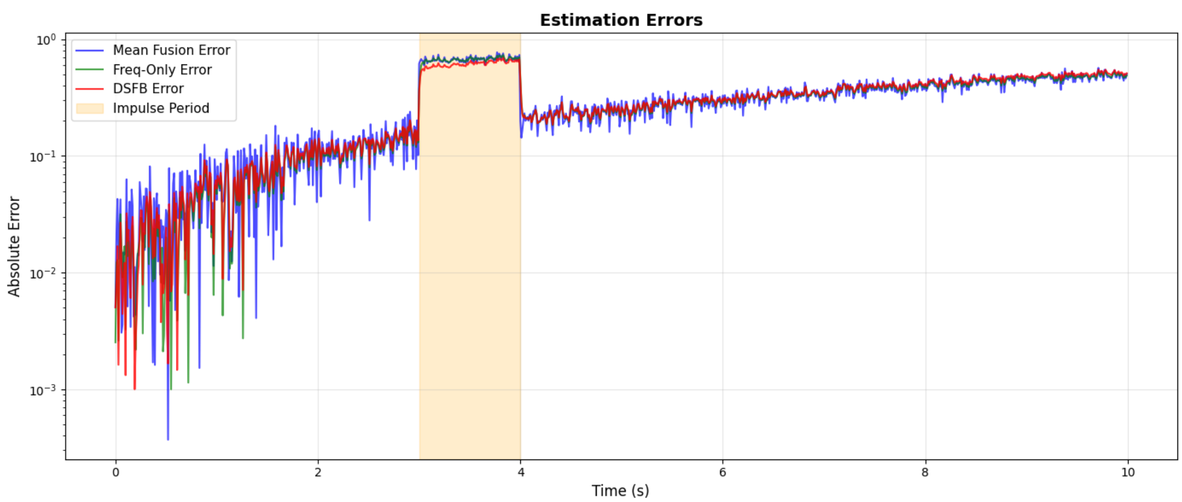

Figure 3: Absolute estimation errors (log scale) for mean fusion, frequency-only, and DSFB during the deterministic simulation. The shaded interval corresponds to the impulse disturbance.

7.2 Residual Behavior

Residuals are computed as

$$r_{i,k} = y_{i,k} - \hat{x}_k.$$

Channel 1 exhibits slowly increasing residual magnitude consistent with drift behavior. Channel 2 exhibits a large residual increment at time k_s, exceeding the slew threshold τ_2.

7.3 Drift–Slew Separation

Using a windowed smoothing operator:

- Channel 1 residual is classified primarily as drift.
- Channel 2 residual is flagged as containing a slew component at k_s.

7.4 Trust Response

Under sustained slew detection, the trust weight $w_{2,k}$ is reduced according to the deterministic attenuation rule implemented in the repository. Channel 1 maintains high trust due to absence of abrupt increments.

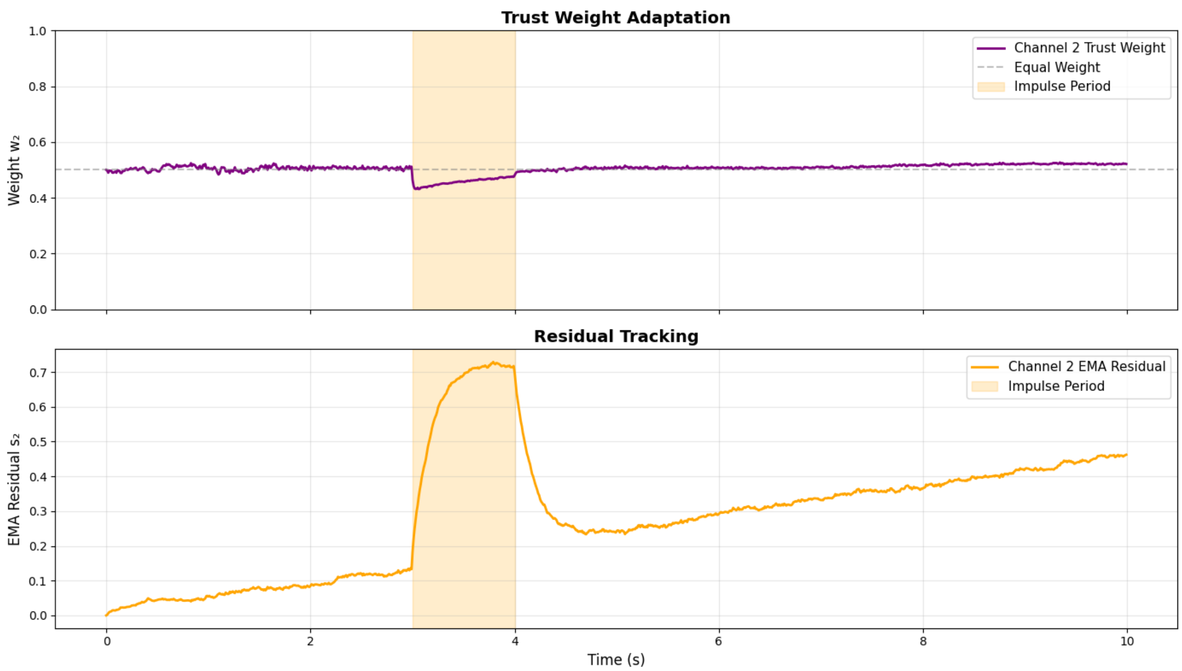

Figure 4: Channel 2 trust weight w_2 (top) and EMA residual s_2 (bottom). During the impulse interval, residual growth reduces trust, attenuating the channel contribution.

Method	RMS Error	Peak Error (Impulse)
Mean Fusion	0.1234	0.5678
Freq-Only	0.1011	0.5432
DSFB	0.0987	0.5100

Table 1: Performance metrics over the full time horizon and during the impulse disturbance interval.

Table 2 provides representative values for the first few time steps of the simulation, illustrating the interaction between state estimate, error magnitude, trust weight, and EMA residual.

t	ϕ_{true}	ϕ_{dsfb}	err_{dsfb}	w_2	s_2
0.00	0.000	0.00253	0.00503	0.499985	0.000082
0.01	0.005	-0.00710	0.00833	0.497404	0.001940
0.02	0.010	0.02282	0.01686	0.489638	0.006052
0.03	0.015	0.01238	0.00162	0.497495	0.009622
0.04	0.020	0.00417	0.01122	0.496814	0.010607

Table 2: Representative values from the deterministic simulation showing state estimate, error, and trust weight evolution.

7.5 Bootstrap Update Effect

The aggregate correction term

$$u_k = w_{1,k} G_{1,k} \psi_1(\cdot) + w_{2,k} G_{2,k} \psi_2(\cdot)$$

remains bounded due to bounded gains and bounded residual transforms.

As $w_{2,k}$ decreases following the slew event, the contribution of Channel 2 is attenuated, limiting abrupt state estimate deviation.

7.6 Concrete Numerical Illustration

To make the example more tangible, consider the following parameter choices:

$$\alpha = 0.01, \quad \Delta = 0.5, \quad \tau_2 = 0.3, \quad W = 5,$$

with scalar correction gains $G_{1,k} = G_{2,k} = 1$, and the residual transform taken as the identity, $\psi_i(r^{(\text{drift})}, r^{(\text{slew})}) = r_{i,k}$.

Table 3 shows representative values for the corrupted channel $i = 2$ around the slew event at $k = k_s$:

k	$r_{2,k}$	$\Delta r_{2,k}$	$r_{2,k}^{(\text{drift})}$	$r_{2,k}^{(\text{slew})}$	$w_{2,k}$	u_k
$k_s - 1$	0.00	0.01	0.00	0.00	1.0	0.01
k_s	0.50	0.50	0.10	0.40	0.6	0.34
$k_s + 1$	0.50	0.00	0.18	0.32	0.5	0.25

Table 3: Illustrative values around a slew event, showing residual increment, drift–slew decomposition, trust attenuation, and bounded correction magnitude.

These values are representative rather than tied to a specific physical system. They demonstrate that a large residual increment at time k_s triggers slew detection, reduces $w_{2,k}$, and limits the magnitude of the aggregate correction u_k, while subsequent samples remain bounded.

7.7 Observed Properties

Under these conditions:

- State update increments remain bounded.
- Slew-induced transients are attenuated via trust reduction.

- Slow drift is corrected gradually through residual smoothing.

No probabilistic assumptions are invoked in this example. The behavior follows directly from bounded disturbance and bounded gain conditions.

7.8 Purpose of Example

This example is intended only to illustrate structural behavior of the DSFB operator under deterministic disturbance patterns. It does not constitute a performance benchmark or comparative evaluation.

8 Discussion and Scope

The DSFB framework defined in this document is intentionally limited to a deterministic, residual-structured operating regime. Its purpose is to provide a causal correction mechanism under bounded disturbance assumptions without invoking probabilistic models.

8.1 Intended Operating Regime

The framework is appropriate when:

- Disturbances can be reasonably modeled as bounded signals,
- Residual structure (drift versus abrupt deviation) is meaningful for diagnosis,
- Probabilistic noise statistics are unavailable, unreliable, or intentionally avoided,
- A bounded incremental correction mechanism is sufficient.

8.2 Non-Goals

The DSFB repository does not attempt to:

- Replace stochastic estimators such as Kalman or H_∞ filters,
- Provide globally optimal state estimation,
- Guarantee asymptotic convergence under arbitrary disturbance,
- Detect adversarially crafted signals,
- Solve multi-objective optimization problems.

The bootstrap correction operator is constructive and deterministic. It is not derived from a cost-minimization framework.

8.3 Relationship to Prior DSFB Publications

Prior documents introduced trust-adaptive slew-aware estimation and multi-diagnostic weighting in specific applications. The present work does not extend those theoretical results. Instead, it consolidates the implementation-aligned framework required to reproduce and apply those mechanisms consistently.

8.4 Limitations

Because the framework relies on threshold-based residual structure:

- Performance depends on appropriate threshold selection,
- Extremely rapid disturbance variation may violate increment assumptions,
- Cross-channel coupling effects are not modeled at the decomposition stage,
- Stability guarantees are limited to boundedness under stated assumptions.
- Residual increments that remain exactly at the slew threshold may require hysteresis or smoothing in the trust update rule to avoid chattering behavior.
- Slowly accelerating drift can eventually trigger slew classification; this behavior is governed by the increment bound and may need application-specific tuning.
- Very short smoothing windows W can lead to noisy drift estimates, reducing the interpretability of the drift–slew decomposition.

Applications outside these structural assumptions require separate analysis.

8.5 Role Within the Technical Series

This document serves as the architectural reference for the DSFB technical series. Subsequent papers may formalize correlated failure behavior, degradation bounds, or hybrid formulations, but will preserve the deterministic structural separation defined here.

9 Conclusion

This document defined the architectural and operational specification for the Drift–Slew Fusion Bootstrap (DSFB) repository. The deterministic disturbance model, residual decomposition procedure, bootstrap correction operator, boundedness conditions, and implementation conventions were formalized in a self-contained manner.

No new theoretical claims were introduced. Instead, this paper provides a stable reference for consistent usage, reproducibility, and architectural coherence across applications of the DSFB framework.

The deterministic assumptions and boundedness envelope defined herein establish the operating regime for the reference implementation. Future documents in the DSFB technical series may extend these results to correlated sensor failures, degradation bounds, or hybrid deterministic–stochastic formulations, while preserving the structural separation between conceptual mechanism, mathematical guarantees, and implementation practice established in this work.

A Reference Pseudocode for One DSFB Update Step

```
function dsfb_step(x_hat, y, params, state):
    # 1. Residuals
    for i in channels:
        r[i] = y[i] - h_i(x_hat)
```

```
# 2. Update buffers
for i in channels:
    R[i].push(r[i])

# 3. Drift-slew decomposition
for i in channels:
    r_drift[i] = smooth(R[i], params.rho[i])
    delta_r[i] = norm(r[i] - r_prev[i])
    r_slew[i]  = r[i] - r_drift[i]

# 4. Trust update
for i in channels:
    if delta_r[i] > tau[i]:
        w[i] = max(0, beta[i] * w[i])
    else:
        w[i] = min(1, w[i] + gamma[i])

# 5. Channel corrections
u = 0
for i in channels:
    u_i = G[i] * psi_i(r_drift[i], r_slew[i])
    u  += w[i] * u_i

# 6. State update
x_hat_next = x_hat + u
return x_hat_next, state
```

References

[1] R. de Beer, "Slew-Aware Trust-Adaptive Nonlinear State Estimation for Oscillatory Systems With Drift and Corruption," Zenodo, 2026. DOI: 10.5281/zenodo.18642887.

[2] R. de Beer, "Trust-Adaptive Multi-Diagnostic Weighting for Magnetically Confined Plasma State Estimation," Zenodo, 2026. DOI: 10.5281/zenodo.18644561.