TRC Canonical Modulation Architecture

Neutral Rewrite with Implementable Metrics

1. Overview

This system defines a dual-layer modulation architecture for dynamically adjusting a scalar trust signal. It combines numerical signal processing with a configurable biasing mechanism. The goal is to maintain stable, interpretable system outputs by regulating curvature and filtering interference.

2. Core Signal Definitions

- T — Trust scalar, real number in $[-1, 1]$
- $\frac{dT}{dt}$ — First derivative (rate of change)
- $\frac{d^2T}{dt^2}$ — Second derivative (curvature)
- I_c — Internal consistency (bounded in $[0, 1]$)
- Ξ — Symbolic interference (noise coefficient in $[0, 1]$)
- $\kappa_{\max}$ — Curvature clamp threshold
- A_{mod} — Output modulation adjustment
- B_n — Bias tier $n \in \{1, ..., 8\}$ for weighting control parameters

3. Modulation Bias Function

Bias is computed as:

$$A_{\text{mod}} = \max\left(0.05,\ B_n \times 0.01 \times \left|\frac{d^2T}{dt^2}\right| \cdot (1 - \Xi) \cdot I_c\right)$$

Each B_n corresponds to a bias function described below. All terms are numerical and bounded. This function outputs a capped trust modulation value adjusted for stability and interference.

4. Modulation Pipeline

1. Accept raw trust signal T.
2. Derive momentum and curvature.
3. Apply symbolic interference coefficient Ξ.
4. Clamp curvature if $|\frac{d^2T}{dt^2}| > \kappa_{\max}$.
5. Compute A_{mod} via bias function.
6. Output updated trust $T' = T + A_{\text{mod}}$.

5. System Parameters and Bias Tiers

Each tier corresponds to an independent control vector.

1. **Data Consistency** — assesses input agreement across sources.
2. **Evidence Responsiveness** — tunes weight assigned to new inputs vs. historical inertia.
3. **Change Stability** — controls acceleration tolerance to avoid erratic behavior.
4. **Conflict Resolution** — reduces modulation when contradictory data is detected.
5. **Delay Tolerance** — increases buffer window for late-arriving or asynchronous data.
6. **Error Minimization** — penalizes modulation under high uncertainty or self-contradiction.
7. **Fairness Bias** — allocates modulation evenly across competing inputs.
8. **Meta-Adjustment** — dynamic tier that adapts other bias values using system-wide heuristics.

Each tier outputs a weighting coefficient $B_n \in [0.5, 1.5]$ depending on policy or configuration.

6. Implementation Functions

6.1 Quantizer

`float compute_modulation(curvature, Xi, Ic, Bn):`

1. Compute: $A = \max(0.05,\ Bn \cdot 0.01 \cdot |curvature| \cdot (1 - \Xi) \cdot Ic)$
2. Return A

6.2 Regulator

`float modulate_trust(T, Xi, Ic, Bn, curvature):`

1. Clamp: $curvature = \min(curvature, \kappa_{max})$
2. $A = compute\_modulation(curvature, Xi, Ic, Bn)$
3. Return $T' = T + A$

7. Test Protocol

- Simulate trust $T = 0$ for $t = 0 \ldots 50$
- Inject $\Xi = 0.8$ at $t = 20$
- Set $I_c = 0.95$, $Bn = 1.2$, $\kappa_{max} = 0.6$
- Log: T, A_{mod}, Ξ, $curvature$
- Validate T' trajectory remains stable and bounded

8. Output Format

Data is exported per timestep:

- `[timestamp, T, curvature, A_mod, Xi, Ic, Bn]`
- Output to CSV or real-time log for analysis

9. Containment Guarantees

- Symbolic recursion is reduced by bounding modulation under interference.
- Emotional or philosophical labels are excluded from control flow.
- Tier weights are tunable but interpreted numerically only.
- System does not model or reflect identity or beliefs—only signal coherence and responsiveness.