Tonic Meta–Control for Adaptive Safety–Compute Allocation via Persistent Vigilance Dynamics

Kyungtae Han, Yitao Chen, Nejib Ammar, Onur Altintas
Toyota Motor North America R&D, InfoTech Labs
{kyungtae.han, yitao.chen, nejib.ammar, onur.altintas}@toyota.com

***Abstract*—Autonomous driving systems that incorporate deliberative reasoning modules, such as multi-step planners or large language models, face a fundamental resource allocation problem because safety demands deep inference under risk while embedded automotive platforms impose strict compute and latency budgets. We propose *tonic meta-control*, a mechanism that maintains a persistent vigilance state (the *tonic* signal, updated every timestep) via asymmetric piecewise dynamics with dead-band hysteresis. A dead-band guarantees exact state hold during boundary oscillations, providing formal guarantees on boundedness and monotonic transitions without continuous-integrator drift, and maps risk regimes to deliberation levels that govern compute allocation when rapid prediction-error surprise (the *phasic* signal) fires. On a synthetic testbench spanning five risk scenarios and in the `highway-env` driving simulator, tonic meta-control achieves 25–50% compute reduction versus phasic-only triggering, scaling to 46–52% under superlinear compute cost regimes. A memoryless threshold ablation isolates the contribution of accumulated vigilance history, confirming its advantage on transient and cyclic risk patterns. While reducing computational overhead on interactive traffic, tonic meta-control supports safety-aware autonomous reasoning on resource-constrained embedded platforms.**

I. Introduction

Integrating deep deliberative modules (e.g., learned deliberative modules or high-cost planners) into autonomous driving and ADAS introduces a critical resource allocation challenge, namely *when* to invoke costly deliberation and *how much* compute to allocate. Embedded automotive platforms impose strict compute, latency, and energy budgets that preclude deep reasoning at every timestep, yet the system must respond decisively during hazardous regime shifts such as construction zones or merging traffic.

We model deliberation depth using two abstract levels: S1 (reactive) and S2 (deliberative). Their concrete implementation is application-specific (Section IV). Constant high-compute operation (always S2) provides maximal safety coverage but wastes resources during benign highway driving. For example, over 500 timesteps per episode, a static S2 policy expends 1000 compute units even when 90% of timesteps present negligible risk. Conversely, constant low-compute operation (always S1) minimizes cost but leaves the system vulnerable during hazardous regime shifts.

An alternative is to trigger reasoning on surprise: when the current risk observation deviates from prediction, invoke deep deliberation. While computationally efficient, such *phasic-only* approaches (rapid, event-driven responses to sudden changes) suffer from two pathologies. First, they fire

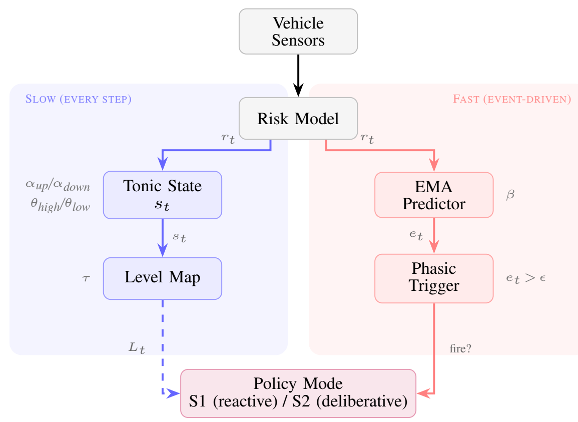

Fig. 1. System architecture. Risk r_t feeds two pathways: the *tonic* pathway (left, blue) integrates r_t every timestep via asymmetric dynamics ($\alpha_{\text{up}} > \alpha_{\text{down}}$) with dead-band hysteresis ($\theta_{\text{high}}/\theta_{\text{low}}$), mapping the tonic state s_t to a deliberation level L_t via threshold τ. The *phasic* pathway (right, red) fires only on upward prediction-error surprise $e_t > \epsilon$ (upward only). The tonic level (dashed) sets deliberation depth when a phasic trigger fires (solid).

at maximum deliberation depth regardless of context, unable to calibrate response to persistent risk level. Second, near decision boundaries where risk oscillates around a threshold, they exhibit costly thrashing, repeatedly triggering and de-triggering without stable adaptation, which is problematic under automotive real-time scheduling constraints.

We propose *tonic meta-control*, a control law that decides *when* and *at what depth* to invoke deliberation. It maintains a persistent *tonic state* (a slowly evolving scalar vigilance variable) via asymmetric piecewise dynamics with dead-band hysteresis, while a separate fast *phasic* trigger fires on prediction-error surprise. The tonic/phasic terminology is borrowed from neuroscience, where tonic activity provides sustained background context and phasic bursts respond to salient events [1]. Figure 1 illustrates the architecture, showing the separation of slow and fast pathways.

The mechanism has four key properties:

- **Fast escalation, slow decay**: The escalation rate exceeds the decay rate, encoding a *cautious relaxation* bias suited to safety-critical operation.
- **Dead-band hysteresis**: Within an intermediate risk band, the tonic state holds exactly (no arithmetic, no drift), eliminating boundary oscillation.
- **Deliberation mapping**: The tonic state maps to deliberation levels (S1 or S2) via a threshold, setting compute cost when a phasic trigger fires.

- **Directional triggering**: The phasic trigger fires only on upward surprise (threat onset), suppressing unnecessary deliberation during safety improvements.

Unlike binary hysteresis that alternates between two fixed modes, tonic meta-control allocates deliberation depth conditioned on accumulated risk history, yielding stability-preserving compute allocation without boundary oscillation. The main contributions of this paper are as follows:

1) Architectural separation of phasic triggering (fast, event-driven) from tonic depth-control (slow, context-providing), enabling compute allocation calibrated to persistent risk level;
2) A directional trigger guard, applied consistently across all surprise-triggered methods, that fires only on upward surprise (threat onset), suppressing wasteful downward-surprise re-triggers;
3) Closed-form settling-time characterization under asymmetric piecewise integration;
4) Formal behavioral invariants (bounds, monotonic escalation/decay, dead-band exact hold, cooldown) verified across all 30 configurations;
5) Empirical compute–safety Pareto validation on synthetic and `highway-env` interactive traffic scenarios under embedded-style compute accounting, with empirical cost grounding via an execution ladder.

We validate across five synthetic risk scenarios, six comparison methods (including a memoryless threshold ablation), and in the `highway-env` driving simulator [2] with interactive traffic.

II. Related Work

A. Adaptive Compute Allocation

Scaling inference cost to input difficulty has been explored through early exit networks [3], speculative decoding [4], mixture-of-experts [5], and adaptive computation time [6]. These methods adapt computation within a single model execution. In contrast, tonic meta-control operates at the system scheduling level, selectively engaging distinct driving modules under real-time constraints.

B. Neuroscience of Arousal and Vigilance

The tonic/phasic framework originates from studies of the locus coeruleus–norepinephrine (LC-NE) system [1], where tonic firing rates set baseline arousal and phasic bursts respond to task-relevant stimuli. Our architecture is conceptually inspired by this separation: the tonic state provides persistent context while the phasic trigger responds to prediction error. The asymmetric dynamics mirror the observation that arousal escalates more rapidly than it subsides [7]. We emphasize that this analogy motivates the design; the formal properties (Section III-C) are established independently of the biological model.

C. Event-Triggered and Hybrid Control

Event-triggered control reduces communication or computation by executing updates only when error signals exceed state-dependent thresholds [8], [9]. Classical formulations provide stability guarantees under measurement-error bounds but employ binary execute/skip updates without modulating execution depth. Tonic meta-control differs in two respects: (1) it separates trigger timing from deliberation depth allocation, and (2) it introduces persistent state dynamics that encode risk history rather than instantaneous deviation, enabling safety–compute tradeoff management beyond purely stability-driven triggering.

D. Safe Reinforcement Learning and Meta-Control

Safe RL addresses safety–performance tradeoffs through constrained optimization [10] and shielding [11]. Tonic meta-control is a *controller of controllers*, deciding the depth of reasoning rather than performing reasoning itself, a notion related to meta-control in cognitive science [12]. Unlike RL-based approaches, our dynamics are deterministic and formally verifiable, making them suitable for safety-critical certification.

E. Risk-Aware Autonomy

Risk estimation and risk-sensitive decision-making in autonomous vehicles have been studied through reachability analysis [13] and risk-sensitive objectives [14]. Our risk model (weighted linear combination of normalized features) is intentionally simple; the contribution lies in the meta-control policy that maps risk signals to compute allocation, not in the risk estimator itself. The framework is agnostic to the risk model and can incorporate more sophisticated estimators.

III. Methodology

The tonic meta-control pipeline has four components: risk estimation, surprise computation, tonic state dynamics, and execution triggering, organized into the slow and fast pathways shown in Figure 1.

A. Risk Model

At each timestep t, the environment provides raw features: inverse time-to-collision, traffic density, relative velocity, and lane deviation. Each feature is min-max normalized to $[0, 1]$ and combined via a weighted linear model:

$$r_t = \mathrm{clamp}\left(\sum_{i=1}^{n} w_i \cdot \frac{x_i^{(\mathrm{raw})} - x_i^{\min}}{x_i^{\max} - x_i^{\min}},\ 0,\ 1\right), \qquad (1)$$

where $w_i \geq 0$ and $\sum_i w_i = 1$. The features above are one instantiation; the meta-control core is agnostic to the upstream risk estimator, requiring only a scalar $r_t \in [0, 1]$.

B. Risk Smoothing and Surprise

An exponential moving average (EMA) produces a smoothed risk estimate:

$$\hat{r}_t = \beta \cdot r_t + (1 - \beta) \cdot \hat{r}_{t-1}, \quad \hat{r}_0 = 0. \tag{2}$$

Because the update incorporates the current observation r_t, $\hat{r}_t$ is a smoothed estimate rather than a one-step-ahead prediction. We define surprise as the residual between observation and estimate:

$$e_t = |r_t - \hat{r}_t| = (1 - \beta)\,|r_t - \hat{r}_{t-1}|. \tag{3}$$

C. Tonic State Dynamics

The tonic state $s_t \in [0, 1]$ integrates risk context via piecewise dynamics with three regimes:

$$s_t = \begin{cases} s_{t-1} + \alpha_{\text{up}}(1 - s_{t-1}) & \text{if } r_t > \theta_{\text{high}} \\ s_{t-1} - \alpha_{\text{down}} \cdot s_{t-1} & \text{if } r_t < \theta_{\text{low}} \\ s_{t-1} & \text{if } \theta_{\text{low}} \le r_t \le \theta_{\text{high}} \end{cases} \tag{4}$$

with clamping to $[0, 1]$ after each update. The dead-band case is an exact copy ($s_t = s_{t-1}$, no arithmetic).

The asymmetry $\alpha_{\text{up}} > \alpha_{\text{down}}$ encodes fast escalation and cautious relaxation. The dead-band prevents oscillation near decision boundaries, unlike continuous integrators or EMA estimators that drift with each crossing.

The dynamics satisfy four verifiable properties:

1) *Monotonic escalation*: sustained $r_t > \theta_{\text{high}}$ implies $s_t \ge s_{t-1}$, since $\alpha_{\text{up}}(1 - s_{t-1}) \ge 0$.
2) *Monotonic decay*: sustained $r_t < \theta_{\text{low}}$ implies $s_t \le s_{t-1}$, since $-\alpha_{\text{down}} s_{t-1} \le 0$.
3) *Dead-band hold*: $s_t = s_{t-1}$ exactly, by construction.
4) *Bounded state*: $s_t \in [0, 1]$ for all t, by contractive updates and clamping.

The escalation and decay dynamics also admit closed-form settling times. Under a sustained high-risk regime ($r_t > \theta_{\text{high}}$ for all t in the interval), starting from $s_0 = 0$, the tonic state evolves as $s_t = 1 - (1 - \alpha_{\text{up}})^t$. Reaching level S2 requires $s_t \ge \tau$ (the S1/S2 mapping threshold), giving an escalation settling time of

$$k_{\uparrow} = \left\lceil \frac{\log(1 - \tau)}{\log(1 - \alpha_{\text{up}})} \right\rceil. \tag{5}$$

Similarly, under sustained $r_t < \theta_{\text{low}}$ starting from $s_0 = 1$, the state decays as $s_t = (1 - \alpha_{\text{down}})^t$. Returning below the threshold τ requires

$$k_{\downarrow} = \left\lceil \frac{\log \tau}{\log(1 - \alpha_{\text{down}})} \right\rceil. \tag{6}$$

At default parameters ($\alpha_{\text{up}} = 0.15,\ \alpha_{\text{down}} = 0.05,\ \tau = 0.5$): $k_{\uparrow} = 5$ steps and $k_{\downarrow} = 14$ steps, a 3:1 ratio reflecting the asymmetry $\alpha_{\text{up}}/\alpha_{\text{down}} = 3$.

Figure 2 illustrates the three update regimes.

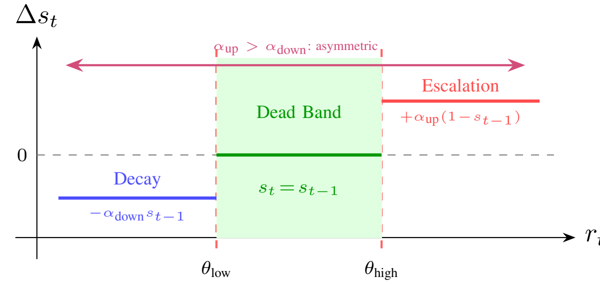

Fig. 2. Piecewise tonic state update rule. Below θ_{low}: slow decay. Above θ_{high}: fast escalation. In the dead band (shaded): exact hold, no arithmetic. Asymmetry encodes safety-first escalation and cautious relaxation.

Algorithm 1 Tonic Meta-Control: Per-Timestep Pipeline

Require: r_t, $\hat{r}_{t-1}$, s_{t-1}, t_{last}
$\hat{r}_t \leftarrow \beta \cdot r_t + (1 - \beta) \cdot \hat{r}_{t-1}$ {EMA}
$e_t \leftarrow |r_t - \hat{r}_t|$ {Surprise}
$s_t \leftarrow$ TONICUPDATE(s_{t-1}, r_t) {Eq. 4}
$L_t \leftarrow$ LEVELMAP(s_t) {Eq. 7}
if $e_t > \epsilon$ **and** $r_t > \hat{r}_t$ **and** $t - t_{\text{last}} \ge c$ **then**
  Invoke reasoning at level L_t; $\text{cost}_t \leftarrow C_{L_t}$
else
  $\text{cost}_t \leftarrow 0$
end if
return s_t, L_t, cost_t

D. Deliberation Levels and Triggering

The tonic state maps to discrete deliberation levels via a single threshold:

$$L_t = \begin{cases} \text{S1} & \text{if } s_t < \tau \\ \text{S2} & \text{if } s_t \ge \tau \end{cases} \tag{7}$$

where $0 < \tau < 1$. S1 and S2 represent reactive and deliberative processing respectively. The specific behavior at each level is application-dependent; Section IV defines the instantiation used in our evaluation.

A phasic trigger fires when surprise e_t exceeds a threshold ϵ, the surprise is *upward* ($r_t > \hat{r}_t$, indicating threat onset), and a minimum inter-trigger interval of c timesteps (cooldown) has elapsed:

$$\text{triggered}_t = \mathbf{1}[e_t > \epsilon] \ \wedge\ \mathbf{1}[r_t > \hat{r}_t] \ \wedge\ \mathbf{1}[t - t_{\text{last}} \ge c]. \tag{8}$$

The directional guard ensures triggers occur on threat onset rather than risk reduction, preventing unnecessary invocation during safety improvements. When triggered, the system invokes reasoning at level L_t with cost C_{L_t}. The tonic state updates *every timestep* regardless of whether a trigger fires; this independence is critical for maintaining accurate risk context. The separation of tonic state evolution (slow pathway) and phasic triggering (fast pathway) defines a hybrid dynamical system with state-dependent event activation. Algorithm 1 summarizes the complete per-timestep pipeline.

IV. Experimental Setup

We evaluate on a synthetic testbench with scalar risk series $r_t \in [0, 1]$ that isolates the meta-control mechanism from

TABLE I
COMPARISON METHODS AND DEFAULT CONFIGURATION.

Method	Trigger	Level
STATIC-S1	Every step	Always S1
STATIC-S2	Every step	Always S2
FIXED-TICK-TONIC	Every step	Tonic-mapped
PHASIC-ONLY	$e_t > \epsilon$	Always S2
PHASIC-THRESHOLD	$e_t > \epsilon$	Threshold-mapped*
TONIC META-CONTROL	$e_t > \epsilon$	Tonic-mapped

Default: $\beta = 0.2$, $\alpha_{\text{up}} = 0.15$, $\alpha_{\text{down}} = 0.05$, $\theta_{\text{high}} = 0.6$, $\theta_{\text{low}} = 0.4$, $\tau = 0.5$, $\epsilon = 0.1$, $c = 5$, $C_{\text{S1/S2}} = 1/2$.
All phasic methods share the same directional guard ($r_t > \hat{r}_t$) and cooldown c (Eq. 8).
*Memoryless threshold: $r_t > \theta_{\text{high}} \Rightarrow$ S2, $r_t < \theta_{\text{low}} \Rightarrow$ S1, else S2. Isolates tonic memory contribution.

perception noise and closed-loop traffic variability, making behavioral invariants directly verifiable against ground-truth values. We additionally validate under closed-loop interactive traffic in the `highway-env` simulator [2].

A. Scenarios

Five risk scenarios ($T = 500$–600 timesteps) stress-test different aspects of the dynamics:

1) *Step regime shift*: Abrupt transitions low→high→low at $t = 167$ and $t = 333$ ($r = 0.1/0.8$). Tests escalation/relaxation latency.
2) *Noisy boundary*: Risk oscillates around $\theta_{\text{high}}/\theta_{\text{low}}$ with Gaussian noise ($\sigma = 0.08$). Tests dead-band stability and thrashing resistance.
3) *Gradual ramp*: Linear increase $0.1 \rightarrow 0.9$ then decrease. Tests surprise-trigger sensitivity to slowly varying risk. Note: by construction, surprise-only triggering is insensitive to slow monotonic drift, as EMA tracking keeps e_t near ϵ, isolating tonic-state evolution from phasic invocation.
4) *Impulse spikes*: Low baseline with brief ($\Delta t = 5$) high-risk spikes. Tests recovery behavior and short-memory effects.
5) *Multiple regime cycles*: Three complete low/high cycles ($T = 600$). Tests tracking of repeated transitions without drift.

B. Comparison Methods

Table I summarizes the six comparison methods. All methods use the same tonic state update equation; they differ only in how triggering is decided and how deliberation level is selected. In PHASIC-ONLY, the tonic state is updated every timestep but does not influence level selection; depth is always S2 when triggered. This isolates the contribution of tonic depth-control from trigger policy in the ablation.

For readability, we use short method labels in the remainder: S-S1 (STATIC-S1), S-S2 (STATIC-S2), FT-T (FIXED-TICK-TONIC), PHAS. (PHASIC-ONLY), P-THR. (PHASIC-THRESHOLD), and TM-C (TONIC META-CONTROL).

C. Metrics and Verification

Six metrics are recorded per episode:

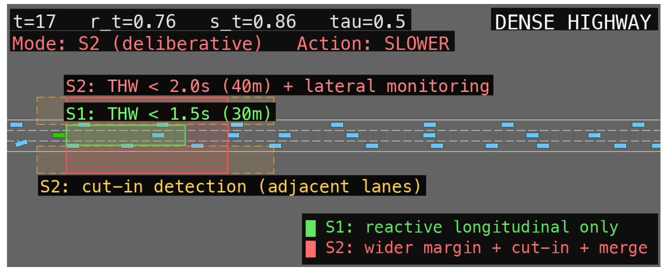

Fig. 3. S1 vs. S2 perception zones on dense highway at $t = 17$ ($r_t = 0.76$, $s_t = 0.86$). S1 (green) monitors only the longitudinal gap (THW < 1.5 s). S2 (red) widens the braking margin and adds lateral cut-in detection (orange) across adjacent lanes.

- **Total compute**: $\sum_t C_{L_t} \cdot \mathbf{1}[\text{triggered}_t]$.
- **Total triggers**: number of reasoning invocations.
- **Average level**: $\bar{L} = \frac{\sum_t L_t \mathbf{1}[\text{triggered}_t]}{\sum_t \mathbf{1}[\text{triggered}_t]}$.
- **Level switches**: number of timesteps where $L_t \neq L_{t-1}$.
- **Escalation latency**: steps from risk entering high regime to $s_t \geq \tau$.
- **Relaxation latency**: steps from risk leaving high regime to $s_t < \tau$.

Every episode is verified against the four formal properties from Section III-C and cooldown enforcement. A parameter sweep (11×11 grid over ϵ and τ, all other parameters fixed at defaults) assesses sensitivity.

D. Interactive Traffic Setup

The synthetic testbench validates formal invariants under controlled risk signals; the interactive traffic evaluation below validates the safety–compute tradeoff under closed-loop dynamics with stochastic traffic. We evaluate in the `highway-env` simulator [2] across three scenarios: *Sparse Highway* (10 vehicles, density 1.0), *Dense Highway* (30 vehicles, density 2.5), and *Merge* (`merge-v0`, 15 vehicles). Deliberation level controls a two-level ego policy: S1 (reactive) applies time-headway braking (THW < 1.5 s) with longitudinal control only. S2 (deliberative) widens the braking margin (THW < 2.0 s), detects cut-in threats from adjacent lanes, and can execute lane changes to avoid merge threats; only S2 adds lateral maneuvers (Figure 3). All six methods control the same ego vehicle with identical perception and risk estimation; they differ only in the meta-control policy governing trigger timing and deliberation level selection. We compare six methods over 30 episodes per scenario-method pair (3 scenarios × 6 methods × 30 episodes = 540 total, each episode up to 400 timesteps) and report Responsibility-Sensitive Safety (RSS) longitudinal safe-distance violation rate [15] ($d_{\min} = v_{\text{ego}}\rho + v_{\text{ego}}^2/2a_{\max} - v_{\text{front}}^2/2a_{\max}$, $\rho = 0.3\,\text{s}$, $a_{\max} = 6\,\text{m/s}^2$) with compute and efficiency metrics.

V. RESULTS

A. Formal Verification

All 30 configurations (5 scenarios × 6 methods) pass all behavioral invariants: state bounds, monotonic escalation,

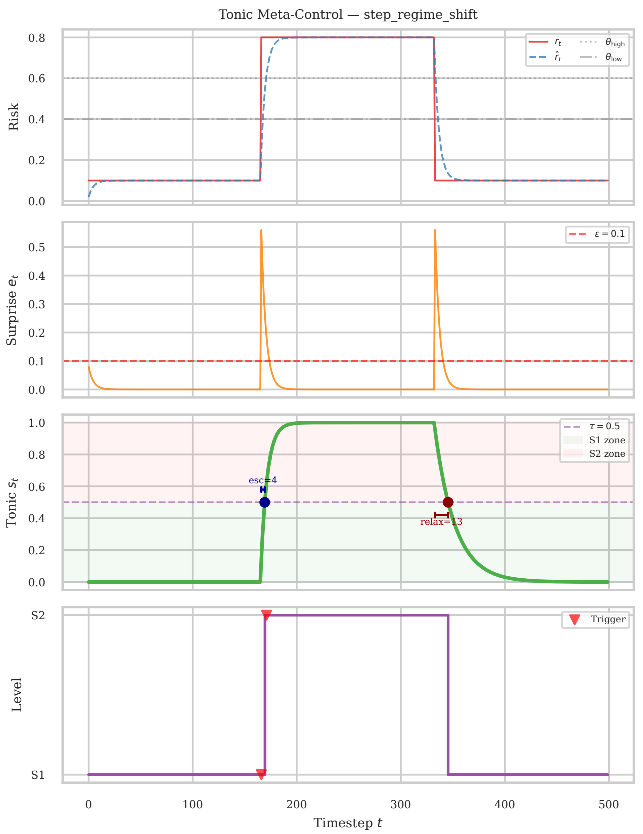

Fig. 4. Tonic meta-control on STEP_REGIME_SHIFT. From top: risk r_t and EMA prediction $\hat{r}_t$; surprise e_t with trigger threshold; tonic state s_t with level threshold τ; deliberation level L_t with trigger markers. Two upward surprises trigger at $t = 166$ and $t = 171$ (separated by cooldown=5); the second occurs while tonic state is still escalating. Note fast escalation and slow relaxation dynamics, with the tonic state deterministically updating every timestep regardless of trigger.

monotonic decay, dead-band hold, and cooldown enforcement, as verified by automated assertions over logged trajectories. This confirms that the implementation conforms to the formal specification in Section III-C.

B. Synthetic Regime Results

On the step regime shift scenario, TM-C (cost 3) outperforms PHAS. (cost 4). The directional guard (Eq. 8) blocks downward surprises: upward transitions demand deliberation, while relaxation proceeds autonomously via tonic dynamics. Figure 4 shows the asymmetry.

a) Adaptation latency: All methods share identical escalation latency (4 timesteps) and relaxation latency (13 timesteps), consistent with closed-form predictions $k_\uparrow = 5$ and $k_\downarrow = 14$ (Eqs. (5)–(6)); the unit offset arises because the first tonic update occurs at the regime-shift timestep itself. This confirms tonic dynamics are trigger-independent: the state updates every timestep regardless of phasic triggers. The asymmetric latency ratio $13/4 \approx 3.3$ reflects the design choice $\alpha_{\text{up}}/\alpha_{\text{down}} = 3$, encoding fast escalation for safety and slow relaxation for stability in oscillatory risk environments.

TABLE II
TOTAL COMPUTE ACROSS ALL SCENARIOS (↓ BETTER). TM-C ACHIEVES LOWEST COMPUTE IN EVERY NON-TRIVIAL CASE.

Scenario	**S-S1**	**S-S2**	**FT-T**	**Phas.**	**P-Thr.**	**TM-C**
step regime	500	1000	676	4	4	**3**
noisy boundary	500	1000	771	50	39	**37**
gradual ramp	500	1000	759	**0**	**0**	**0**
impulse spikes	500	1000	515	10	10	**5**
multi-cycle	600	1200	914	12	12	**9**

S-S1: STATIC-S1; S-S2: STATIC-S2; FT-T: FIXED-TICK-TONIC; PHAS.: PHASIC-ONLY; P-THR.: PHASIC-THRESHOLD; TM-C: TONIC META-CONTROL.

b) Cross-scenario comparison: Table II confirms TM-C achieves lowest compute in all five scenarios (tied at zero on GRADUAL_RAMP due to monotonic drift; see limitation note in Section IV). The directional guard suppresses downward-surprise firings for all phasic variants; the additional 25–50% compute reduction of TM-C versus PHAS. comes from tonic depth-control assigning lower levels when the tonic state has not yet fully escalated. On NOISY_BOUNDARY, TM-C (37) outperforms both PHAS. (50) and P-THR. (39) because tonic memory assigns lower average levels during boundary oscillations. On IMPULSE_SPIKES and MULTI-CYCLE, tonic memory defers expensive S2 invocations until the tonic state is sufficiently elevated, outperforming memoryless threshold methods. On the gradual ramp, all phasic methods produce zero triggers: intrinsic to surprise-based triggering when the EMA closely tracks a monotonic signal.

c) Boundary stability: The NOISY_BOUNDARY scenario stress-tests the dead-band mechanism. Risk oscillates around θ_{high} and θ_{low} with Gaussian noise ($\sigma = 0.08$), repeatedly crossing in and out of the dead band.

PHAS. fires 25 upward-surprise triggers (all S2, total compute 50). TM-C fires the same 25 triggers but at average level 1.48, reducing compute to 37, a 26% reduction. More importantly, the tonic state remains stable with only 2 level switches. A memoryless threshold integrator (without dead-band) would thrash continuously across the threshold with every noise-driven boundary crossing; the dead-band exact-hold enforces stable s_t hold during oscillations, preventing drift from noise-driven transitions.

C. Parameter Sensitivity

We evaluate robustness over an 11×11 grid spanning $\epsilon \in [0.05, 0.25]$ and $\tau \in [0.30, 0.70]$, run across all five synthetic scenarios (N = 5 episodes per configuration per scenario, 121 configurations × 5 scenarios = 605 runs total). Across all 121 configurations, no invariant violations or pathological behaviors were observed. The heatmaps in Figure 5 suggest a largely separable structure: ϵ dominates total compute (lower ϵ admits more triggers, higher cost), while τ controls average deliberation level (lower τ reaches S2 earlier during escalation). No obvious harmful interactions appear across the grid. This separability is favorable for tuning: operators

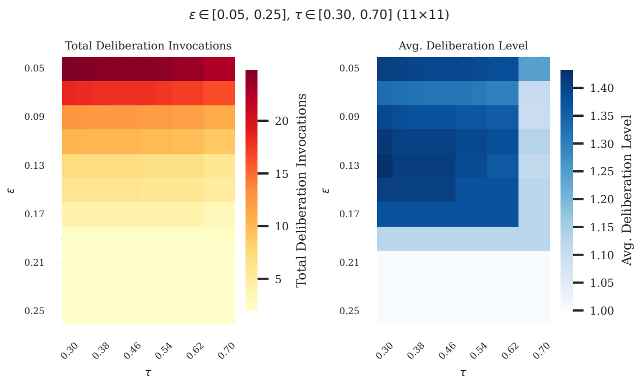

Fig. 5. Parameter sensitivity heatmaps over the 11×11 grid ($\epsilon \in [0.05, 0.25]$, $\tau \in [0.30, 0.70]$). Left: average deliberation level is controlled by τ (excluding zero-trigger scenarios). Right: total deliberation invocations decrease with ϵ.

can adjust ϵ to control trigger frequency independently from τ without unexpected coupling effects.

The staircase banding in the deliberation level heatmap reflects the discrete two-level mapping: small threshold changes shift the tonic state across the level boundary, producing jumps rather than smooth gradients.

D. Cost Model Sensitivity

The default linear cost model ($C_{S1/S2} = 1/2$) is conservative; real multi-pass deliberation typically scales super-linearly with depth. Testing moderate ($1/5$) and extreme ($1/10$) cost ratios, TONIC META-CONTROL's advantage strengthens: savings increase to 46% (moderate) and 52% (extreme) versus 29% (linear), confirming that tonic depth-control becomes more beneficial as deep deliberation grows more expensive.

E. Interactive Traffic Validation

Table III presents `highway-env` results across the three scenarios defined in Section IV-D. Three key findings emerge. *First*, only TM-C and FT-T remain crash-free in dense highway (see table note for per-method crash rates). *Second*, TM-C achieves 97–98% compute reduction versus static methods: it matches the best RSS rate in merge (0.211, tied with FT-T) and remains crash-free in dense traffic while reducing compute by 97–98%, with a bounded increase in RSS violation rate (0.282 vs. 0.199 for S-S2), reflecting an explicit safety–compute tradeoff under constrained deliberation budgets. Notably, some methods achieving the lowest RSS rate also incur crashes (Table III note), as rare maneuver interactions are not fully captured by the longitudinal RSS metric. TM-C leads η in dense (4.9, tied with PHAS.) and merge (13.0); P-THR. leads in sparse η (7.4 vs 6.6) and sparse compute (11 vs 13). *Third*, although all phasic methods share the same directional trigger guard, TM-C's lower deliberation levels alter ego trajectories in this closed-loop setting, producing different risk profiles and compounding the per-trigger savings from tonic depth-control.

TABLE III
HIGHWAY-ENV EVALUATION (MEAN OVER 30 EPISODES, 400 MAX STEPS). $\eta = (1 - \text{RSS RATE})/\text{COMPUTE} \times 100$ (↑ BETTER).

Scenario	Method	Compute	Avg. L	RSS Rate	η↑
Sparse	S-S1	400	1.00	0.159	0.2
	S-S2	800	2.00	0.207	0.1
	FT-T	419	1.05	0.177	0.2
	Phas.	31	2.00	0.207	2.6
	P-Thr.	**11**	1.19	**0.169**	**7.4**
	TM-C	13	1.11	0.177	6.6
Dense	S-S1	388	1.00	0.268	0.2
	S-S2	752	2.00	**0.199**	0.1
	FT-T	443	1.11	0.282	0.2
	Phas.	16	2.00	**0.199**	**4.9**
	P-Thr.	16	1.37	0.295	4.5
	TM-C	**15**	1.27	0.282	**4.9**
Merge	S-S1	88	1.00	0.224	0.9
	S-S2	174	2.00	0.218	0.4
	FT-T	105	1.19	**0.211**	0.8
	Phas.	11	2.00	0.218	7.1
	P-Thr.	**6**	1.30	0.229	12.2
	TM-C	**6**	1.14	**0.211**	**13.0**

Method abbreviations as in Table II.
RSS Rate: fraction of timesteps violating RSS longitudinal safe distance.
Dense: S-S1, P-THR. each suffer 1 crash (3%); S-S2, PHAS. each suffer 2 crashes (7%); TM-C and FT-T are crash-free. Merge and Sparse: all methods crash-free.

a) Pareto analysis: Figure 6 plots mean total compute per episode against RSS violation rate across all 18 `highway-env` configurations (3 scenarios × 6 methods). This reveals the compute–safety tradeoff using a standard safety metric (RSS longitudinal safe distance [15]) rather than synthetic coverage.

TM-C consistently lies in the low-compute, low-violation region across all three scenarios. It achieves the lowest total compute (6–15 per episode) while matching or closely approaching the best RSS violation rate observed in each setting. In merge, TM-C ties the best RSS rate (0.211) at less than 4% of S-S2's compute; the dense-traffic tradeoff is detailed in the preceding paragraph.

The static methods cluster at substantially higher compute with relatively modest variation in RSS violation rate, indicating diminishing safety returns under unconditional high-compute operation. In contrast, the phasic methods achieve comparable RSS rates at one to two orders of magnitude lower compute, demonstrating that structured, event-driven deliberation can preserve safety performance while dramatically reducing computational demand (see Table III for detailed η comparisons across scenarios).

VI. DISCUSSION

A. Design Properties

The dead-band exact-hold is stronger than a near-zero update: it guarantees zero arithmetic in the hold region, preventing the accumulation of floating-point drift that continuous integrators would introduce under long-horizon embedded execution. The $\alpha_{up} > \alpha_{down}$ asymmetry complements this by

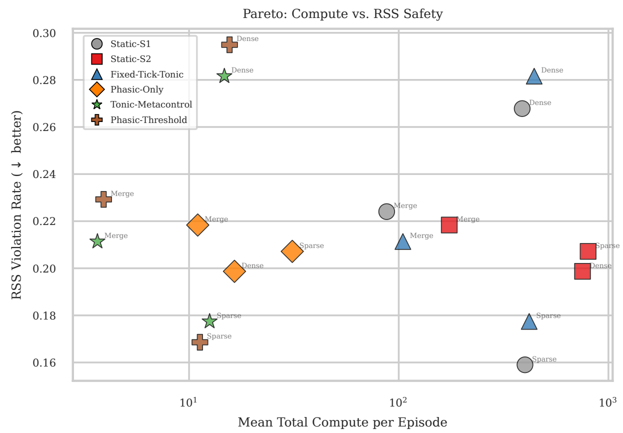

Fig. 6. Mean total compute vs. RSS violation rate across 3 `highway-env` scenarios × 6 methods. TM-C (star markers) occupies the low-compute, low-violation region in all scenarios.

buffering against premature de-escalation when risk briefly abates.

B. Separation of Fast and Slow Signals

The tonic state is not a trigger but context that conditions the response: a surprise event during sustained merge interaction triggers S2, but the same surprise during benign cruise triggers only S1, because the tonic state encodes accumulated risk context.

C. Implications for Intelligent Transportation Systems

Tonic meta-control converts high-cost deliberative computation into a sparse, event-driven workload: the meta-controller itself costs only a compare-and-multiply-add per step, runs in $O(1)$ with no data-dependent branching or dynamic memory allocation, and the S1 path provides a fast-reflex fallback schedulable within a single planning frame, suggesting compatibility with embedded automotive power and thermal envelopes.

D. Limitations

The ego policy uses deterministic IDM variants conditioned on deliberation level; the contribution is meta-control over deliberation depth, not the driving policy.

Specific limitations include: (1) fixed parameters without online adaptation; (2) scalar risk without multi-dimensional interaction modeling; (3) purely surprise-driven triggering misses slowly evolving hazards; a periodic fallback trigger (every k_{fb} steps) would partially recover coverage at modest cost; (4) integrating learned deliberative modules in closed-loop driving remains future work; (5) higher-fidelity validation in CARLA with real sensor feeds is needed beyond the current `highway-env` evaluation.

VII. Conclusion

We presented tonic meta-control, a formally specified mechanism for adaptive safety–compute allocation in autonomous driving. The design separates fast phasic surprise triggering from a slow persistent vigilance state that governs deliberation depth, enabling compute allocation calibrated to accumulated risk context. Across synthetic benchmarks and closed-loop `highway-env` traffic, tonic meta-control consistently reduces compute while maintaining competitive RSS violation rates and crash-free performance in interactive scenarios. All behavioral invariants are verified across evaluated configurations. The mechanism incurs constant-time per-step overhead and requires no dynamic memory allocation, supporting deployment on real-time embedded automotive and ADAS platforms. Future directions include higher-fidelity evaluation in CARLA [16] with real sensor feeds, integration of learned deliberative modules in closed-loop driving, a periodic fallback trigger to address slowly evolving hazards, and online parameter adaptation.

References

[1] G. Aston-Jones and J. D. Cohen, “An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance,” *Annual Review of Neuroscience*, vol. 28, pp. 403–450, 2005.
[2] E. Leurent, “An environment for autonomous driving decision-making,” https://github.com/eleurent/highway-env, 2018.
[3] S. Teerapittayanon, B. McDanel, and H.-T. Kung, “BranchyNet: Fast inference via early exiting from deep neural networks,” in *International Conference on Pattern Recognition*, 2016, pp. 2464–2469.
[4] Y. Leviathan, M. Kalman, and Y. Matias, “Fast inference from transformers via speculative decoding,” in *International Conference on Machine Learning*, 2023, pp. 19 274–19 286.
[5] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” in *International Conference on Learning Representations*, 2017.
[6] A. Graves, “Adaptive computation time for recurrent neural networks,” arXiv:1603.08983, 2016.
[7] S. J. Sara, “The locus coeruleus and noradrenergic modulation of cognition,” *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 211–223, 2009.
[8] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.
[9] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *IEEE Conference on Decision and Control*, 2012, pp. 3270–3285.
[10] J. García and F. Fernández, “A comprehensive survey on safe reinforcement learning,” *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.
[11] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, “Safe reinforcement learning via shielding,” in *AAAI Conference on Artificial Intelligence*, 2018, pp. 2669–2678.
[12] M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen, “Conflict monitoring and cognitive control,” *Psychological Review*, vol. 108, no. 3, pp. 624–652, 2001.
[13] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, “A general safety framework for learning-based control in uncertain robotic systems,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019.
[14] A. Majumdar and M. Pavone, “How should a robot assess risk? towards an axiomatic theory of risk in robotics,” in *Robotics Research*. Springer, 2020, pp. 75–84.
[15] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “On a formal model of safe and scalable self-driving cars,” arXiv:1708.06374, 2017.
[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Conference on Robot Learning*, 2017, pp. 1–16.