Why Intelligent Systems Waste Energy

Baseline Regulation as a Missing Architectural Primitive

Robotics / Embodied AI

Tyson Jeffreys
Independent Researcher
tyson@staygolden.dev

Version 1.0 — January 14th, 2026

1 Introduction

Energy efficiency has become a central concern across biological, computational, and engineered intelligent systems. In artificial intelligence, this concern is typically addressed through improvements in hardware efficiency, algorithmic optimization, model compression, or task-level performance tradeoffs. In biological systems, energy efficiency is often discussed in terms of metabolic cost, evolutionary pressure, and resource allocation. Despite progress in these domains, a shared assumption remains largely unexamined: that intelligent systems should operate continuously in an active, task-optimized mode whenever they are online.

This paper challenges that assumption.

We propose that a significant source of energy inefficiency arises not from the *cost of intelligence itself*, but from the absence of explicit **baseline regulation**—an internal architectural mechanism that maintains a low-energy reference state and governs transitions into and out of active modes. In biological organisms, such regulation is fundamental: metabolism, attention, and action are continuously modulated relative to a regulated baseline rather than driven at constant high activation. Artificial systems, by contrast, are typically designed to operate in a perpetually "engaged" state, even when task demands are minimal or absent.

The result is chronic internal activity: persistent inference, monitoring, optimization, and state updates that incur energy cost regardless of necessity. This pattern scales poorly—from individual agents to large model deployments—particularly as system complexity increases.

The central claim of this work is not that current intelligent systems are inefficiently implemented, but that they are **incompletely architected**. Baseline regulation is treated implicitly, if at all, and rarely as a first-class design primitive. Without such regulation, systems lack a principled mechanism for quiescence, internal recovery, or load-sensitive modulation, leading to avoidable energy expenditure.

This paper introduces baseline regulation as an architectural lens rather than a prescriptive algorithm. We formalize the concept using a minimal energetic model of intelligent agents, demonstrate how the absence of baseline regulation leads to chronic over-activation, and explore the implications for artificial systems operating at scale. Importantly, we do not argue for reduced capability or slower performance. Instead, we suggest that **capability expressed relative to a regulated baseline** may be both more energy-efficient and more robust.

The contribution is intentionally cross-domain. By drawing parallels between physiological regulation and artificial systems design, we aim to surface a missing layer in current architectures—one that is already well understood in living systems but underutilized in artificial ones. We conclude by outlining concrete experimental pathways for evaluating baseline regulation in existing AI systems and propose directions for future research.

- propose baseline regulation as a missing architectural primitive
- formalize convex energy model + duty-cycle framing
- derive energy efficiency from regulated agents
- propose falsifiable experiments

2 Energy Posture in Large-Scale Inference Systems

2.1 The hidden cost of always-on intelligence

Modern AI inference infrastructure is optimized primarily for latency and throughput, with systems operating near-continuously at elevated utilization. While this maximizes responsiveness, it introduces structural inefficiencies:

- sustained high average power draw
- frequent peak loads requiring overprovisioning
- increased cooling and thermal overhead
- sensitivity to workload variance

Crucially, most inference systems treat activity itself as free, penalizing only task error or latency, not internal computational load.

This creates a posture analogous to operating biological systems under constant high arousal: functional, but energetically costly and brittle.

2.2 Baseline-aware energy model

Let an inference system be characterized by an internal activity state $x(t)$, representing aggregate computational load (e.g., active tokens/sec, attention bandwidth, routing complexity).

Total power draw:

$$\mathcal{P}(t) = P_{\text{idle}} + k\, x(t)^{\alpha} \qquad \text{with } \alpha \geq 1$$

Energy over time horizon T:

$$E = \int_0^T \mathcal{P}(t)\, dt$$

In current deployments:

- $x(t)$ remains persistently elevated
- variability is managed reactively
- idle time is treated as inefficiency

2.3 Introducing a regulated baseline

We define a baseline operating point x_0, representing the lowest internally stable load consistent with readiness.

Rather than minimizing task loss alone, the system optimizes:

$$J = \mathbb{E}\left[\int_0^T \left(C_{\text{task}}(t) + \lambda\,(x(t) - x_0)^2 + \mu\,\ddot{x}(t)^2\right) dt\right]$$

Where:

$(x - x_0)^2$ penalizes sustained deviation from baseline

$\ddot{x}^2$ penalizes internal thrashing and micro-spikes

This reframes inference as:

event-driven activation around a regulated baseline, not continuous maximal readiness.

2.4 Datacenter-level consequences

From this model, several infrastructure-level effects follow mechanically:

Reduced average utilization

Systems spend the majority of time near x_0, lowering mean power draw.

Lower peak-to-average ratio (PAR)

Shorter, rarer compute spikes reduce provisioning and cooling requirements.

Suppressed instability

Penalizing $\ddot{x}^2$ reduces high-frequency load oscillations that amplify energy waste.

Economic implication

Even modest reductions in duty cycle and PAR produce outsized cost savings, because datacenter costs scale with peaks, not means.

These gains arise without reducing model capability—only its operating posture.

3 Regulated Agents as the Mechanism

3.1 Where regulation enters the stack

The baseline-aware energy posture described above cannot be implemented purely at the infrastructure layer. It must emerge from agent-level regulation.

We model an agent's internal activity state $x(t)$ as governed by:

$$\dot{x}(t) = -\beta\,(x(t) - x_0) + u(t) + w(t)$$

Where:

$-\beta\,(x(t) - x_0)$: intrinsic return-to-baseline

$u(t)$: policy-driven activation (reasoning, planning, retrieval)

$w(t)$: environmental or task perturbations

This introduces an explicit **regulatory pull** toward baseline.

3.2 Interpretation in LLM-based agents

For language or tool-using agents, $x(t)$ may correspond to:

- reasoning depth
- number of parallel thought branches

- retrieval or tool-call frequency
- attention bandwidth or memory access

A regulated agent:

- remains near baseline under low uncertainty
- activates computation sharply when needed
- collapses activity immediately after resolution

This differs from current agents, which often:

- reason continuously
- maintain elevated internal activity even after uncertainty resolves

3.3 From agent regulation to energy savings

When agents regulate internal activity:

- inference bursts become shorter and rarer
- average compute per task decreases
- infrastructure sees smoother load profiles

Thus:

Energy efficiency is downstream of agent regulation.

No new hardware is required.

No reduction in capability is assumed.

The gain arises from doing less when less is required.

4 Testable Predictions

This framework predicts that introducing explicit baseline regulation will yield:

- Lower average watts per inference
- Reduced peak utilization
- Improved energy stability under variable workloads
- Comparable or improved task performance
- Better scaling economics at deployment level

These predictions are measurable using existing inference metrics.

5 Baseline Regulation as an Architectural Primitive

Current intelligent systems are largely architected around task execution, optimization, and performance metrics. Internal state is treated as a byproduct of computation rather than as a regulated

variable in its own right. In contrast, biological systems explicitly regulate internal state—metabolic, autonomic, and attentional—relative to a baseline that is neither idle nor maximally active, but dynamically maintained.

We propose that **baseline regulation should be treated as a first-class architectural primitive**, on par with perception, memory, and action. In this framing, an intelligent system is not defined solely by what it can do, but by how it returns—how it settles, recovers, and modulates internal activity in the absence of external demand. Formally, baseline regulation introduces:

- A low-energy reference state
- Explicit transitions into and out of active modes
- Internal cost for sustained deviation from baseline
- Recovery dynamics independent of task completion

This differs from conventional resource management approaches (e.g., throttling, batching, or sleep states) by embedding regulation within the agent's internal dynamics rather than imposing it externally. The result is not reduced capability, but **capability expressed relative to need.**

6 Implications for Energy Efficiency and System Robustness

6.1 Energy Efficiency Beyond Optimization

Energy efficiency in AI is typically pursued through incremental gains: faster hardware, lower-precision arithmetic, pruning, or improved algorithms. Baseline regulation addresses a different layer entirely—the **structural cause of chronic energy expenditure.**

Without baseline regulation:

- Internal processes remain active regardless of relevance
- Monitoring and inference loops persist indefinitely
- Energy use scales with uptime, not task demand

With baseline regulation:

- Energy expenditure becomes episodic rather than continuous
- Idle time is genuinely low-cost
- System-wide energy usage scales with engagement, not presence

This distinction becomes increasingly important as systems scale from single agents to large, always-on deployments.

6.2 Robustness Through Internal Cycling

Biological systems rely on cyclical dynamics—activation followed by recovery. Continuous activation leads to brittleness, depletion, and failure. The same pattern appears in artificial systems: always-on agents exhibit degraded performance under sustained load, increased error rates, and poor recovery from perturbation. Baseline regulation introduces internal cycling:

- Activation is bounded

- Recovery is explicit
- Stability is maintained through oscillation rather than equilibrium

This suggests that energy efficiency and robustness are not competing goals, but **co-emergent properties** of regulated architectures.

7 Experimental Directions and Testable Implications

Baseline regulation is not proposed here as a philosophical analogy, but as a falsifiable architectural hypothesis: explicit internal regulation should reduce energy use and improve stability/robustness at comparable task performance, particularly under bursty or adversarial conditions. This section outlines experiments designed to be feasible with existing systems and minimal architectural changes.

7.1 Core prediction

For comparable task success, a regulated system should demonstrate:

Lower integrated energy proxy

$E = \int_0^T \mathcal{P}(t)\,dt$ decreases, primarily by reducing time spent at high $x(t)$ (convex region of the power curve).

- **Reduced internal volatility**

  Lower $\int_0^T \|\ddot{x}(t)\|^2\,dt$, fewer spikes, fewer oscillatory "thrash" patterns.

- **Graceful degradation under constraint**

  When capped (compute, battery, thermal, latency), performance should degrade smoothly rather than catastrophically.

These predictions can be evaluated whether $\mathcal{P}(t)$ is measured directly (hardware power draw) or approximated via a proxy (tokens/sec, GPU utilization, CPU package power, actuator current, duty cycle).

7.2 Experiment A: LLM inference as a regulated dynamical system

Goal: Test whether adding a baseline regulator reduces energy per useful output without meaningfully degrading answer quality.

Setup:

- Choose a fixed model + fixed hardware.
- Define internal load $x(t)$ as a measurable proxy:
  - GPU utilization, GPU power draw, or tokens/sec (or a weighted combination).

- Implement a simple regulator:
  - Add a controller that enforces a "baseline band" for $x(t)$ outside explicit high-demand spans.
  - Practical implementations could include:
    - micro-batching policies
    - adaptive throttling
    - enforced quiescent windows between bursts
    - compute-aware decoding constraints

Metrics:

- Energy per 1k tokens (or per completed request)
- Latency distribution (mean + tail)
- Output quality proxy (human eval or rubric score)
- Variance/peaks of $x(t)$, and $\int_0^T \|\ddot{x}(t)\|^2 \, dt$ (as a thrash proxy)

Prediction: A regulated inference pipeline will reduce energy and thermal spikes, and reduce tail-latency variance, especially under mixed workloads.

7.3 Experiment B: Agent frameworks and "internal load penalties"

Goal: Test whether internal load regularization improves long-horizon efficiency and reduces instability in agentic loops. Setup:

- Choose an agent benchmark (tool use / multi-step tasks / environment navigation).
- Define $x(t)$ as:
  - number of concurrent tool calls
  - planner depth / branching factor
  - memory read/write intensity
  - frequency of replanning events
- Add a penalty term:

$$\lambda \|x(t) - x_0\|^2 + \mu \|\dot{x}(t)\|^2$$

Metrics:

- Task success rate
- Total calls / tokens / tool invocations (as energy proxies)
- Replanning frequency, oscillation rate
- Time-to-completion

Prediction: The regulated agent will show fewer runaway loops, fewer "panic replans," and a lower integrated compute footprint at similar success rates.

7.4 Experiment C: Robotics control with a baseline regulator layer

Goal: Test whether a baseline regulator reduces actuator energy and improves robustness to perturbation. Setup:

- A standard locomotion or manipulation controller (MPC/RL/policy).
- Define internal load $x(t)$ as a proxy for "control strain," e.g.:
  - total torque squared $\sum_i \tau_i^2$
  - jerk / acceleration penalties
  - contact slip rate
  - estimator covariance / prediction error
- Add a slow baseline regulator (1–5 Hz) that:
  - detects persistent elevation of $x(t)$
  - adjusts gait parameters / controller gains / planning horizon to return toward $\mathcal{X}_0$

baseline.

Metrics:

- Joules per meter (or per task)
- Slip / fall rate under perturbation
- Recovery time after disturbances
- Peak torque / thermal stress proxies

Prediction: The regulated system will show reduced energy usage and fewer catastrophic failures under perturbation, with comparable nominal performance.

7.5 Negative tests (what would falsify the claim)

This hypothesis would be weakened if:

- energy use does not decrease despite reduced time at high $x(t)$
- regulation improves energy but consistently causes unacceptable task degradation
- regulation introduces instability (e.g., oscillatory duty cycling) unless heavily tuned

Including falsifiers makes the paper read as serious and scientific, not rhetorical.

8 Limitations and Scope

This work does not claim that baseline regulation is sufficient to solve energy inefficiency across all intelligent systems, nor that biological analogies should be applied uncritically to artificial agents. Key limitations include:

- The absence of a single canonical implementation of baseline regulation
- Potential tradeoffs between responsiveness and quiescence

- Domain-specific constraints that may limit applicability

Additionally, baseline regulation should not be conflated with inactivity. Poorly designed regulation mechanisms could introduce latency or underreaction in safety-critical systems. As in biological systems, effective regulation requires adaptive cycling, not static suppression.

This paper focuses on architectural framing rather than implementation details. Specific mechanisms—whether learned, rule-based, or hybrid—are intentionally left open.

9 Relation to Prior Work

Elements of baseline regulation appear implicitly across multiple research areas:

- **Homeostatic reinforcement learning**, which incorporates internal variables into reward structures.
- **Active inference and predictive processing**, which emphasize minimizing internal surprise or free energy.
- **Embodied AI**, which treats agents as energetically situated systems
- **Energy-based models**, which formalize dynamics relative to energy landscapes

What is missing across these approaches is an explicit unifying concept of **baseline regulation as an architectural layer** rather than a task-specific optimization. This work aims to surface that layer and articulate its implications independently of any single framework.

10 Conclusion

This paper argues that a significant source of energy inefficiency in intelligent systems arises not from poor optimization, but from missing architectural structure. Specifically, the absence of explicit baseline regulation leads to chronic internal activity, unnecessary energy expenditure, and reduced robustness at scale.

By introducing baseline regulation as a first-class architectural primitive, we offer a lens through which energy efficiency, stability, and scalability can be jointly addressed. The proposal is intentionally conservative: it does not demand new objectives, new learning paradigms, or new hardware. It suggests only that intelligent systems, like biological ones, benefit from knowing how to rest.

If validated experimentally, baseline regulation represents a low-level design principle with wide-ranging implications—from individual agents to global AI infrastructure. At a time when energy constraints are becoming central to the future of intelligent systems, such structural considerations may prove as important as algorithmic advances.

A Appendix A — Mathematical Details

This appendix expands the formal treatment of baseline regulation introduced in the main text. The goal is not to prescribe a single mathematical formulation, but to show that the concept is well-posed, extensible, and compatible with existing models.

A.1 Expanded Dynamics

Let the internal state of an intelligent system be represented by:

$$x(t) \in \mathbb{R}^n$$

where $x(t)$ includes latent variables such as internal activation, attentional intensity, inference depth, or metabolic/compute load.

We define a baseline state:

$$x_0$$

representing a low-energy, regulated internal configuration.

System dynamics are modeled as:

$$\dot{x}(t) = f(x(t), u(t), e(t)) - g(x(t), x_0)$$

Where:

$u(t)$: task-driven inputs

$e(t)$: environmental perturbations

$g(x(t), x_0)$: a restoring (regulatory) term pulling the system toward baseline

The key distinction from standard dynamical systems is that $g(x(t), x_0)$ the regulatory term is always active, not only during explicit recovery phases.

A.2 Internal Cost Function

We define internal load as deviation from baseline:

$$L(t) := \|x(t) - x_0\|^2$$

Total system cost over time becomes:

$$J = \mathbb{E}\left[\int_0^T \left(C_{\text{task}}(t) + \lambda\, P(t)\right) dt\right]$$

Where:

$C_{\text{task}}(t)$ captures task performance or error

λ controls sensitivity to internal load

This formulation explicitly penalizes sustained internal activation, even when task error is low.

A.3 Alternative Cost Terms

Other formulations may be appropriate depending on system design:

- Energy-weighted load:

$$L(x) = (x - x_0)^\top W (x - x_0)$$

- Temporal load (duty cycle penalty):

$$L(t_0, t_1) := \int_{t_0}^{t_1} \mathbf{1}\{\|x(t) - x_0\| > \theta\}\, dt$$

- Rate-of-change penalty:

$$L_{\dot{x}} := \|\dot{x}(t)\|^2$$

These are alternative definitions of internal load; in all cases, the overall objective remains $J = \mathbb{E}\left[\int_0^T \left(C_{\text{task}}(t) + \lambda\, P(t)\right) dt\right]$ the same, with P chosen per system. These variants emphasize that baseline regulation is not a single equation, but a design constraint.

A.4 Stability Considerations

A regulated baseline introduces a Lyapunov-like structure:

$$V(x) = \|x - x_0\|^2$$

If:

$$\dot{V}(x) \leq 0 \quad \text{when } u(t) = 0,\ w(t) = 0,$$

then the system is provably stable around its baseline state.

Importantly, stability here does not imply inactivity—it implies bounded, recoverable activation.

B Appendix B — Mapping internal state to real systems

This appendix grounds the abstract formulation in existing intelligent systems.

B.1 Large Language Model Inference

For LLMs, $x(t)$ may represent:

- attention activation magnitude
- depth of token processing
- number of active layers
- internal KV-cache utilization

Baseline regulation would imply:

- adaptive depth or width during inference
- enforced low-activation states between prompts
- explicit decay of internal activation rather than persistent readiness

This reframes “idle” inference as a regulated state, not a passive one.

B.2 Agent Frameworks

In agent-based systems:

- $x(t)$ may include planning horizon
- memory retrieval intensity
- simulation rollouts
- policy entropy

Baseline regulation enables:

- agents that disengage when no action is required
- bounded planning loops
- recovery periods independent of reward signals

This directly addresses runaway deliberation and tool overuse.

B.3 Robotics and Embodied Systems

For robots:

- $x(t)$ maps to actuator readiness
- sensor polling rate
- control loop stiffness
- internal state estimation intensity

A regulated baseline allows:

- posture-level rest states
- adaptive sensor throttling
- smooth transitions between action and quiescence

This aligns closely with biological motor regulation and offers clear energy savings.

B.4 Mapping internal state and baseline to real systems

This table is intentionally practical: each row names an internal state $x(t)$, proposes a baseline $\mathcal{X}_0$, and points to a measurement path for $\mathcal{P}(t)$.

Table: Concept → implementation mapping

System	What is $x(t)$ (internal state)	What is $\mathcal{X}_0$ (baseline manifold / setpoint)	How to measure $\mathcal{P}(t)$ (power / cost)
LLM inference (single request)	KV-cache size + activation statistics; token rate; entropy/uncertainty proxy; "compute intensity" counters	Low-token-rate / low-activation "idle" mode between bursts; target entropy band; target token/sec budget	GPU power (NVML), GPU utilization, SM occupancy; joules/token
LLM serving (production)	Queue depth, batch size, token throughput, cache hit rate, context length distribution	Stable low-queue regime; target batch size band; target context window policy	Cluster power draw; energy per request; P95 latency vs joules
Agent loop (tool-using)	Tool-call rate, action frequency, planner depth, memory writes, context	Low-action "monitoring" baseline with bounded deliberation; target duty cycle	CPU/GPU power + tool-call cost + wall-clock
Robotics control	Joint states + $\dot{x}$; control effort $u(t)$; slip/error metrics	Stable gait/stance manifold; minimal corrective torque regime	Electrical power to motors; $\int \tau^2$; thermal load
Online learning / RL	Policy update magnitude; exploration rate; gradient norms; variance	"Quiescent" consolidation windows; bounded update norm; scheduled rest	GPU/TPU power during training; joules per improvement
Distributed systems (general)	CPU load, IO contention, retry rate, backpressure, latency variance	Low-retry, low-contention steady-state	Power + cloud cost + failure/retry energy

Baseline regulation is implementable whenever (a) internal load can be sensed and (b) quiescence/throttling can be enforced...

Measure $\mathcal{P}(t)$ with NVML / motor current / cluster meters. Choose an internal state proxy $x(t)$. Define a baseline $\mathcal{X}_0$ as a low-load setpoint band. Add a regulator term to penalize deviation + thrash. Enforce duty-cycle limits.

C Appendix C — Glossary

Baseline

A low-energy, dynamically stable internal state from which activation occurs and to which the system naturally returns in the absence of demand.

Regulation

The continuous process by which internal state is constrained relative to baseline, independent of external task success.

Duty Cycle

The proportion of time a system spends above its baseline activation threshold.

Internal Load

The cumulative energetic, computational, or structural cost associated with deviation from baseline state.

Activation

Any internal process that increases distance from baseline, including computation, planning, sensing, or actuation.

Recovery

The process of returning internal state toward baseline following activation; distinct from task completion.

Note on authorship and tools

This work was developed through iterative reasoning, modeling, and synthesis. Large language models were used as a collaborative tool to assist with drafting, clarification, and cross-domain translation. All conceptual framing, structure, and final judgments remain the responsibility of the author.