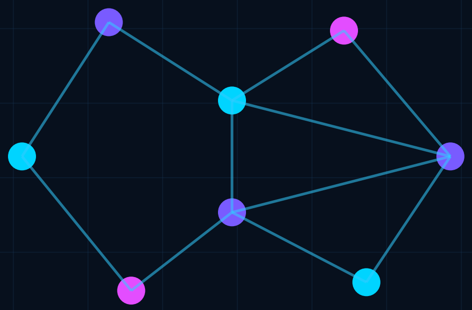
MYTHOS SYSTEMS

TECHNICAL LIBRARY / RESEARCH INTELLIGENCE

CONTINUOUS-TIME AI

Liquid Neural Networks (LNNs)

Current-source review, evidence-bounded adoption decision, explicit limitations, reproducibility controls, and preserved source research note.



PERMANENT ID	EVIDENCE	ADOPTION	FRESHNESS
RES-015	A-	RESEARCH AND TARGETED PILOT	STABLE WATCH

Freshness review: 2026-09-17 / Next scheduled review: 2027-03-16

RESEARCH CONTROL SURFACE

Required Research Fields

Each field remains independently reviewable; evidence strength does not imply production authority.

EVIDENCE GRADE — **A-**

Strength of the retained source set and technical basis.

SOURCE AUTHORITY — **2 registered authorities**

Liquid Time-constant Networks; Closed-form Continuous-time Neural Networks

FRESHNESS — **STABLE WATCH**

Reviewed 2026-09-17; dynamic sources require event-driven revalidation.

CONFLICTS — **VISIBLE / NOT SILENT**

Differences between specification, vendor behavior, research claims, and reproduced evidence must remain explicit.

LIMITATIONS — **EXPLICIT**

No certification, procurement approval, legal conclusion, or unbounded production claim is created by this report.

REPRODUCIBILITY — **REQUIRED**

Material claims require exact versions, representative data, failure cases, artifacts, and repeatable acceptance criteria.

ADOPTION POSTURE — **RESEARCH AND TARGETED PILOT**

Use liquid and continuous-time networks only in bounded temporal, control, sensor, or robotics experiments with stability analysis and conventional baselines.

REVIEW TRIGGER — **2027-03-16**

Scheduled at 180 days; earlier on material source, vulnerability, implementation, or contradictory-evidence change.

SOURCE AUTHORITY / FRESHNESS

Authority Register and Freshness Delta

Primary-source lineage is preserved; current-source changes are separated from unperformed implementation claims.

AUTHORITY 01

Liquid Time-constant Networks

https://arxiv.org/abs/2006.04439

AUTHORITY 02

Closed-form Continuous-time Neural Networks

https://www.nature.com/articles/s42256-022-00556-7

2026-09-17 FRESHNESS DELTA

Primary-source register retained and freshness controls advanced to the current review date. No new product or laboratory performance claim is introduced without reproduced evidence.

Freshness policy: scheduled review + immediate review on source supersession, material release, vulnerability, retraction, or contradictory evidence.

LIMITATIONS / CONFLICTS / REPRODUCIBILITY

Evidence Must Survive Contact With Reality

- Source authority is not implementation evidence. Product behavior, scale, security, interoperability, and operator fit remain environment-specific.

- Conflicting evidence is recorded rather than averaged away. Specification behavior, vendor implementation, preprint results, and production observations are separate evidence classes.

- Quantitative claims are informative until reproduced under documented hardware, software, data, topology, load, and failure conditions.

- Adoption posture is reversible. A stronger source set can coexist with a pilot-only or research-only recommendation when maturity, safety, economics, or operability remain unresolved.

ADOPTION / VALIDATION / REVIEW

Decision Gate and Validation Plan

ADOPTION POSTURE

RESEARCH AND TARGETED PILOT

Use liquid and continuous-time networks only in bounded temporal, control, sensor, or robotics experiments with stability analysis and conventional baselines.

1 / SOURCE	Pin exact versions, publication state, retrieved artifacts, and source hashes.
2 / PROTOTYPE	Demonstrate the core mechanism with representative data and instrumented execution.
3 / FAILURE	Exercise malformed input, dependency loss, stale state, overload, recovery, rollback, and misuse.
4 / BASELINE	Compare against an incumbent, classical, or simpler architecture using the same workload.
5 / ACCEPT	Record acceptance criteria, residual limitations, owner, expiration, and revalidation triggers.

EXPIRATION / REVIEW TRIGGER

Next scheduled review: 2027-03-16 (180-day cadence). Review sooner after material source revision, breaking implementation release, critical vulnerability, retraction, benchmark contradiction, changed workload, or changed legal/safety boundary.

8. Complete Source Research Note

SOURCE-LINEAGE NOTICE

The following material preserves the complete original source note, excluding only the conversational wrapper and outer Markdown fence. Legacy status labels and absolute language are retained for traceability and do not override the candidate status, limitations, or adoption decision in this edition.

Document ID: RES-015 Version: 1.0.0 (Definitive Registry Edition) Status: Published Research Note Classification: Public Organization: Mythos Systems (MYTHOS AI) Owner: Aaron Stovall Effective Date: 2026-07-16 Applies To: AI researchers, autonomous systems engineers, and robotics architects designing edge AI, adaptive sensor fusion, and continuous-learning pipelines Companion Standards: MYTHOS-AI-0012 (Neuro-Symbolic AI & Continual Learning), MYTHOS-EMT-0003 (Neuromorphic & NPU), MYTHOS-ROB-0001 (Physical AI & Autonomous Fleets), MYTHOS-AI-0011 (MoE & SSM Architecture)

The architecture of deep learning has hit a biological and mathematical wall. Traditional artificial neural networks (Transformers, CNNs, RNNs) are discrete-time, static systems. Once trained, their synaptic weights are frozen. Because they rely on rigid, fixed parameter spaces, they are catastrophically brittle when exposed to distribution shift (e.g., a self-driving car trained in clear weather encountering snow). To adapt, they must be completely retrained, risking catastrophic forgetting.

Liquid Neural Networks (LNNs), based on continuous-time recurrent neural networks (CT-RNNs) and inspired by the nervous system of the C. elegans nematode, shatter this static paradigm. In an LNN, the synaptic weights and time-constants are not fixed; they are fluid functions of time and input. The network's topology literally rewires itself during inference to adapt to changing stimuli.

This research note defines the mathematical dynamics of LNNs. It dissects the transition from Linear Time-Invariant (LTI) systems to Liquid Time-Constant (LTC) networks, the computational bottleneck of numerical ODE solvers, and the Closed-form Continuous-time (CfC) breakthrough that makes them deployable on edge hardware. Understanding these dynamics is a prerequisite for building the ultra-adaptable, continuous-learning autonomous agents and physical robots mandated by the Mythos Architecture.

To understand LNNs, we must move from discrete matrix multiplications to continuous-time calculus.

1.1 The Continuous-Time Foundation

Traditional RNNs update their hidden state in discrete steps: $h_{t+1} = f(h_t, x_t)$. LNNs are governed by Ordinary Differential Equations (ODEs), where the state changes continuously over time:

- The ODE: $\frac{dh(t)}{dt} = -h(t) + f(h(t), x(t), \tau, A)$
- The term $-h(t)$ is a leakage term, pulling the state back to equilibrium.
- τ is the time constant, dictating how fast the state decays.

1.2 Liquid Time-Constants (LTC)

In a standard CT-RNN, τ is a fixed parameter. In an LTC network (the foundation of LNNs), τ and the synaptic weights A are made input-dependent.

- The Fluid Equation: $\tau(x) \frac{dh(t)}{dt} = -h(t) + A(x)f(h(t), x(t))$

- The Impact: The time constant and the effective connectivity are now fluid functions of the incoming data. If the input is noisy or chaotic, the network can dynamically increase τ to smooth out the noise. If the input is sparse and critical, it can decrease τ to react instantly. The network's architecture is no longer a fixed DAG; it is a liquid topology that flows and rewires based on context.

While mathematically elegant, LNNs introduce severe computational constraints that delayed their commercial adoption.

2.1 The ODE Solver Bottleneck

Because LNNs are defined by differential equations, they cannot be computed via simple forward-pass matrix multiplications. They require numerical ODE solvers (e.g., Runge-Kutta) during both training and inference.

- The Flaw: Solving an ODE requires evaluating the function multiple times per time step to approximate the continuous curve. On standard GPUs (optimized for dense matrix math, not iterative solvers), this made LNNs 10x to 100x slower than equivalent discrete RNNs or Transformers.

2.2 Stability and Boundedness

Continuous-time dynamical systems can explode to infinity if not carefully bounded.

- The Constraint: The fluid time-constants $\tau(x)$ must be strictly bounded to positive values, and the activation functions must be carefully chosen to prevent the ODE from diverging during inference. This makes training LNNs mathematically unstable compared to standard backpropagation.

To solve the ODE solver bottleneck, researchers at MIT CSAIL developed Closed-form Continuous-time (CfC) networks.

3.1 The Analytical Solution

Instead of relying on a GPU to numerically approximate the ODE at runtime, CfC networks derive a closed-form mathematical approximation of the ODE solution.

- The Mechanic: The ODE is split into linear and non-linear parts. The linear part is solved exactly, and the non-linear part is approximated using an exponential decay function.
- The Result: The continuous-time dynamics are compressed back into a discrete update step: $h_{t+1} = \sigma(-\Delta t / \tau) h_t + (1 - \sigma(-\Delta t / \tau)) f(x_t)$.
- The Impact: CfC networks possess the liquid, adaptive properties of LNNs (input-dependent time constants) but can be executed via standard matrix multiplications on GPUs. They are as fast as standard RNNs but possess the robustness and continuous-learning capabilities of ODEs.

The fundamental shift of LNNs is the decoupling of model size from adaptability. LNNs can achieve superhuman performance on dynamic tasks with incredibly small parameter counts (e.g., <20 neurons for a drone hovering task).

4.1 Resistance to Distribution Shift

A Transformer trained to drive a car in summer will crash in winter because the pixel distributions change. An LNN trained to drive in summer can dynamically alter its time-constants and effective connectivity to process the winter snow, treating it as a continuous state-shift rather than an out-of-distribution failure. This is a mandatory property for the Physical AI and autonomous robotics mandated by `MYTHOS-ROB-0001`.

4.2 Dynamic Rewiring During Inference

Because the weights are fluid, the network does not have a fixed "path" for data. If a sensor fails, the fluid time-constants of the neurons connected to that sensor naturally decay to zero, effectively pruning the dead pathway and rerouting the computation through healthy sensors. The network self-heals its topology in real-time.

4.3 Continual Learning without Forgetting

Traditional ANNs overwrite their static weights to learn new tasks (Catastrophic Forgetting). LNNs learn by adjusting their fluid time-constants. When a new task is introduced, the network expands its dynamic range, preserving the time-constants used for older tasks while carving out new fluid states for the new task. This mathematically aligns with the anti-forgetting mandates of `MYTHOS-AI-0012`.

To deploy LNNs in production, the Mythos Architecture (specifically the Nona reasoning loop and Morta execution layer) enforces strict operational boundaries.

5.1 Hybrid CfC-Transformer Routing

LNNs excel at processing continuous time-series data (telemetry, audio, sensor fusion) but lack the dense, in-context reasoning capabilities of Transformers.

- The Mitigation: The architecture MUST route continuous streaming data (e.g., drone IMU, network traffic packets) through a CfC module to extract temporal features and compress the state. The resulting compressed state is then injected into a Transformer or MoE model for high-level logical reasoning. This provides the adaptability of LNNs with the intelligence of LLMs.

5.2 Neuromorphic Hardware Mapping

Standard GPUs are inefficient at simulating continuous-time dynamics.

- The Mitigation: For edge deployments (`MYTHOS-EMT-0003`), CfC models MUST be compiled and mapped to Neuromorphic silicon (e.g., Intel Loihi 2, SpiNNaker 2). Neuromorphic chips natively operate in continuous time via Spiking Neural Networks (SNNs), allowing LNN dynamics to be executed in hardware at sub-milliwatt power envelopes.

5.3 The Deterministic Safety Boundary

Because LNNs dynamically rewire themselves, they can exhibit emergent, non-deterministic behavior when exposed to adversarial inputs.

- The Mitigation: Even though the cognitive layer (Nona) is fluid, the execution layer (Morta) MUST remain strictly deterministic. If the LNN proposes a physical action that violates the safety bounds (e.g., a drone pitching 90 degrees due to a noisy sensor reading), the deterministic control layer (Aegis) MUST override the action. The fluidity of the LNN is bounded by the absolute physics of the safety controller.

Static, discrete-time neural networks are fundamentally flawed for autonomous, physical, and continuous-learning applications. They are brittle, easily confused by changing environments, and forget their training. Liquid Neural Networks, particularly the Closed-form Continuous-time (CfC) variants, offer a mathematically sound path to fluid, adaptive, and robust intelligence. By treating synaptic weights and time-constants as dynamic functions of reality, we transform AI from a frozen snapshot into a continuous, flowing river of intelligence.

```
A static weight is a snapshot; a fluid time-constant is a flow.
A frozen network shatters under reality; a liquid network absorbs it.
A discrete step is a guess; a continuous ODE is a truth.

The parameter count does not dictate intelligence; the adaptability does.

> Environments may be chaotic.
> Continuous adaptation must be absolute.
```

End of Research Note RES-015

9. Source Register

Source	Authority and Status	Freshness Control
Liquid Time-constant Networks https://arxiv.org/abs/2006.04439	Primary research Published research Published: 2020	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.
Closed-form Continuous-time Neural Networks https://www.nature.com/articles/s42256-022-00556-7	Peer-reviewed primary research Published Published: 2022	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.

10. Lineage, Review, and Publication Record

Record	Value
Original source file	RES-015_Liquid_Neural_Networks_LNNs.md
Original source words	1428
Upgrade type	Enterprise research report; source and freshness validation; limitations; adoption decision; reproducibility plan
Generated on	2026-07-22
Current status	Candidate Research Report
Publication authority	Formal Mythos approval not yet recorded
Next review	180 days or event-driven trigger, whichever occurs first