Zhu et al. (2025) — Liquid Neural Networks: Next-Generation AI for Telecom from First Principles

Full Citation: ZHU Fenghao, WANG Xinquan, ZHU Chen, HUANG Chongwen. "Liquid Neural Networks: Next-Generation AI for Telecom from First Principles." Zhejiang University, 2025.

Paper Focus: This is a survey/overview paper that explains the design, features, and benefits of Liquid Neural Networks (LNNs) from the perspective of telecommunications applications. It covers three types of LNNs: Liquid Time-Constant Networks (LTCs), Closed-form Continuous-time Networks (CfCs), and Neural Circuit Policies (NCPs). The paper does not introduce new equations but provides an accessible description of the existing LNN framework and demonstrates its application to channel prediction and beamforming.

Note for implementation: This paper is less equation-dense than the 2018 and 2021 papers. Its value for implementation lies in: (1) explaining the CfC closed-form approximation concept, (2) describing the NCP sparse architecture, (3) providing telecom-specific application context, and (4) summarizing the practical advantages of LNNs.

---

1. Overview of Traditional Neural Networks (Section 2)

The paper reviews four types of traditional neural networks and their limitations, motivating the need for LNNs:

1.1 Feed Forward Neural Networks (FNNs)

Information moves in one direction — from input nodes, through hidden nodes, to output nodes. No cycles or loops. Suitable for simple pattern recognition but **struggle with tasks requiring memory or temporal dependencies** due to lack of internal hidden states.

1.2 Convolutional Neural Networks (CNNs)

Specialized for processing structured grid data (images). Use convolutional layers with filters to capture spatial hierarchies. Robust to variations in image data (shifts, scales, distortions), but not designed for temporal sequences.

1.3 Recurrent Neural Networks (RNNs)

Designed for sequential data. Have connections forming directed cycles allowing information to persist. However, traditional RNNs suffer from **vanishing and exploding gradients** on long sequences. Variants like LSTMs and GRUs address this partially but have limitations:

- Cannot model continuous-time dynamics
- Reduced robustness in highly dynamic environments

1.4 Ordinary Differential Equation Neural Networks (ODE-NNs)

Designed to model continuous-time dynamics. Includes CT-RNNs and ODE-LSTMs. CT-RNNs use ODEs to capture continuous-time sequences, suitable for irregular time intervals, but are computationally intensive. Despite improvements, they face:

- Increased computational complexity
- Potential training instability
- Limited effectiveness in highly dynamic environments

---

2. Design of LNNs (Section 3)

LNNs are uniquely designed based on **first principles**, fundamentally differing from other models in their neuron operation. First principles involve deriving properties and behaviors directly from fundamental laws of nature. LNNs mimic the information transmission mechanisms observed at synapses in the nematode *Caenorhabditis elegans*.

2.1 Liquid Time-Constant Neural Networks (LTCs)

The paper describes the basic information flow of a liquid neuron (illustrated in the paper's Fig. 2):

1. A **presynaptic neuron** transmits information to a **postsynaptic neuron** via the synapse between them using presynaptic stimuli.
2. The potential of the postsynaptic membrane acts as a dynamic variable, representing the **hidden states** in the corresponding neural networks.
3. This entire process is described by an **ordinary differential equation (ODE)** which captures the dynamic, non-linear interactions between neurons.

The paper references the LTC ODE from Hasani et al. (2021), which in the notation of the original papers is:

$$\frac{d\mathbf{x}(t)}{dt} = -\left[\frac{1}{\tau} + f\big(\mathbf{x}(t), \mathbf{I}(t), t, \theta\big)\right]\mathbf{x}(t) + f\big(\mathbf{x}(t), \mathbf{I}(t), t, \theta\big)A \tag{LTC ODE, from Hasani 2021}$$

Key facts highlighted by this paper:

- LTCs have demonstrated exceptional flexibility and generalizability, particularly in vehicle autopilot and vehicular communications.
- LTCs achieved **high-fidelity autonomy in complex autonomous systems with as few as 19 liquid neurons**.
- LTCs can be extended to V2X (vehicle-to-everything) communications.

2.2 Closed-Form Continuous-Time Neural Networks (CfCs)

While LTCs can adapt to changing environments, their **lack of a closed-form solution requires computationally intensive iterative solvers** for forward propagation and backpropagation. The CfC approach (Hasani et al., 2022) addresses this:

A **closed-form solution** was proposed to approximate the true solution of the LTC ODE. The closed-form expression:

- Successfully circumvents the high overhead of traditional ODE solvers
- Approximates the solution with a few parameters
- Is represented by a specially designed deep neural network structure

Conceptual equation (not explicitly numbered in this paper but described):

Instead of numerically solving the LTC ODE at each step, CfCs learn a direct mapping:

$$\mathbf{x}(t + \Delta t) \approx g_{CfC}(\mathbf{x}(t), \mathbf{I}(t), \theta_{CfC}) \tag{CfC concept}$$

where g_{CfC} is a closed-form neural network that approximates the ODE solution. This eliminates the need for L unfolding steps of the fused solver.

Implementation significance: CfCs are represented by a specially designed deep neural network structure (depicted in the paper's Fig. 3) that significantly reduces computational complexity while maintaining the adaptability and robustness characteristic of liquid neural networks.

2.3 Neural Circuit Policies (NCPs)

NCPs combine multiple CfC or LTC neurons into several layers with a specific sparse structure. The paper describes a typical NCP with four distinct layers:

Layer	Name	Role
1	**Sensory neuron layer**	Receives raw input data
2	**Inter neurons layer**	Processes and integrates information internally
3	**Command neurons layer**	Forms higher-level representations and decisions
4	**Motor neurons layer**	Produces final outputs

Key architectural principles:

- Layers feature **sparse connections** both within and between them
- Mimics the sparse connectivity observed in biological neural networks
- Reduces computational complexity
- Accelerates information exchange and fusion

NCP connectivity structure (described in text, for implementation):

- Sensory -> Inter: sparse feed-forward
- Inter <-> Inter: sparse recurrent connections
- Inter -> Command: sparse feed-forward
- Command <-> Command: sparse recurrent connections
- Command -> Motor: sparse feed-forward

This can be implemented via **binary masks** on weight matrices.

Key result: NCPs have demonstrated robust flight navigation capabilities when presented with **out-of-distribution data**, generalizing effectively to scenarios not encountered during training. This makes NCPs especially valuable for real-world deployment.

---

3. Features and Benefits of LNNs (Section 4)

3.1 Superior Generalizability and Robustness

LNNs exhibit superior generalizability and robustness compared to traditional neural networks due to their biologically inspired design that allows **continuous adaptation** to new and varying inputs. They maintain high performance even when faced with data that deviates significantly from the training set.

3.2 Enhanced Expressivity

LNNs produce significantly more detailed and longer latent-space trajectories than Neural ODEs and CT-RNNs, indicating a higher capacity for nuanced temporal representation. The paper references the trajectory length analysis from Hasani et al. (2021).

3.3 Improved Interpretability

LNNs can disentangle complex neural dynamics into comprehensible and distinct behaviors. By leveraging techniques such as decision trees to analyze neural policies, LNNs provide clear explanations for their decision-making processes.

3.4 Lower Complexity

Three factors contribute to lower complexity:

1. **Sparse connectivity** in NCPs reduces computational overhead.
2. **Closed-form solutions** (CfCs) eliminate the need for iterative ODE solvers.
3. **Strong expressive power** enables complex tasks with fewer neurons (e.g., 19 neurons for autonomous driving).

3.5 Continuous-Time Modelling

LNNs leverage ODEs to model dynamic interactions between neurons, capturing the continuous and fluid nature of real-world processes more accurately than discrete-time networks. This enables more precise channel estimation, interference management, and adaptive modulation schemes.

---

4. LNNs for Wireless (Section 5)

4.1 Integrated Sensing and Communication (ISAC)

ISAC merges communication and sensing into a unified framework. LNNs are well-suited for ISAC due to:

- Real-time learning and adaptation
- Handling complex/dynamic environments
- Effective generalization
- Low computational complexity

4.2 Self-Organizing Networks (SONs)

SONs adapt and evolve autonomously in response to changing conditions. LNNs can:

- Predict and address potential network congestion
- Enhance QoS by adapting to communication link quality

- Optimize handovers in mobile networks
- Identify and mitigate faults proactively

---

5. Challenges and Future Research (Section 6)

5.1 Zero-Shot Learning

LNNs have shown some capacity for out-of-distribution data, but deeper understanding is needed. Combining LNNs with data augmentation strategies holds promise.

5.2 Distributed LNNs

Distributing LNNs across devices is vital for large-scale systems. Federated learning is an attractive solution for collaborative local training.

5.3 Multi-Modality Fusion

Combining sensor data, audio, video, and text can improve LNN performance. Challenges include data synchronization and fusion across modalities.

5.4 Training and Inference Latency

Critical for 6G deployment. The computational time for solving ODEs must be compared against sub-millisecond URLLC requirements. Future research needed in:

- Optimized numerical solvers for LNNs
- Model simplification/approximation techniques
- Hardware acceleration platforms
- End-to-end training latency evaluation

---

6. Case Studies (Section 7)

6.1 Channel Prediction with LTCs

Scenario: Urban microcell, outdoor BS serving outdoor/indoor users. User moves at 2 m/s random walk.

Task: Use 20 historical CSI samples to predict 5 future CSI samples.

Simulation parameters (Table I for Fig. 4):

Parameter	Value
BS Antenna Number	4
BS Antenna Spacing	0.5λ
User Number	1
User Antenna Number	1
Central Frequency	6 GHz

Result: The LTC-based approach consistently outperforms all baselines (lower MSE), with the performance gap widening as prediction length increases, particularly when it exceeds 6 steps. This highlights the potential of LTCs for accurate channel prediction in practical and dynamic scenarios.

6.2 Beamforming with NCPs

Scenario: MIMO beamforming system. BS with M antenna elements serves K users, each with N_r antennas. Users at varying velocities (6, 15, 30 m/s) across 700, 600, 500 time intervals.

Simulation parameters (Table I for Fig. 5):

Parameter	Value
BS Antenna Number	64
BS Antenna Spacing	0.5λ
User Number	4
User Antenna Number	2
Central Frequency	28 GHz
Liquid Neuron Number	30

Result: The gradient-based liquid neural network (GLNN) approach leveraging NCPs rapidly surpasses the WMMSE algorithm after a short initial learning period, then maintains superior spectral efficiency (SE) across all dynamic conditions.

---

7. Summary for Implementation

This paper does not introduce new mathematical equations beyond what is in Hasani et al. (2018, 2021) and Hasani et al. (2022, CfC). Its implementation-relevant contributions are:

What This Paper Adds for Implementation:

1. **CfC concept:** The idea that the LTC ODE can be approximated by a closed-form neural network, eliminating the iterative fused solver. For full CfC implementation details, refer to Hasani et al. (2022) "Closed-Form Continuous-Time Neural Networks" in Nature Machine Intelligence.
2. **NCP architecture specification:** The 4-layer sparse structure (Sensory -> Inter -> Command -> Motor) with sparse intra- and inter-layer connectivity. Implementation: use binary masks on weight matrices.
3. **Practical scale:** 19 neurons for autonomous driving, 30 neurons for beamforming — demonstrating that LTCs/NCPs are extremely parameter-efficient.
4. **Telecom application patterns:** Channel prediction (time-series regression with LTCs) and beamforming (real-time optimization with NCPs).

---

8. Parameter and Variable Tables

Since this paper uses the same mathematical framework as Hasani et al. (2021), the parameters and variables are identical. The paper does introduce one additional architectural parameter:

Parameter	Symbol	Description	Typical Value
Liquid neuron number	N	Number of LTC/CfC neurons in the network	19–30 for real-world tasks
NCP layer structure	—	4-layer: Sensory, Inter, Command, Motor	Architecture choice
Sparse connectivity mask	—	Binary mask on weight matrices	Problem-specific
BS antenna number	M	Base station antennas (telecom-specific)	4–64
User number	K	Number of users served	1–4
User antenna number	N_r	Antennas per user device	1–2
Central frequency	f_c	Carrier frequency	6–28 GHz

---

9. Equation Index

This paper does not introduce numbered equations in the formal mathematical sense. The equations it references are:

Reference	Source	Description
LTC ODE	Hasani et al. (2021) Eq. 1	Core LTC hidden state ODE
Fused solver	Hasani et al. (2021) Eq. 3	Closed-form ODE solver update
CfC solution	Hasani et al. (2022)	Closed-form approximation of LTC ODE
NCP architecture	Lechner et al. (2020)	4-layer sparse neural circuit policy

---

10. Key References from This Paper (for further implementation detail)

Ref	Paper	What It Provides
[10]	Hasani et al. (2021) — LTC Networks, AAAI	Core LTC ODE, fused solver, BPTT training, bounds, expressivity
[11]	Lechner et al. (2020) — Neural Circuit Policies, Nature MI	NCP 4-layer architecture, sparse wiring, auditable autonomy
[12]	Hasani et al. (2022) — Closed-Form CT Neural Networks, Nature MI	CfC closed-form solution, eliminates ODE solver
[18]	Chahine et al. (2023) — Robust Flight Navigation, Science Robotics	Out-of-distribution generalization with LNNs
[24]	Yin et al. (2021) — Channel Prediction with LTCs	LTC-based channel prediction methodology
[25]	Wang et al. (2024) — Robust Beamforming with GLNN	NCP-based beamforming