Liquid Neural Dynamics for Temporal Signal Modeling: Applications in EMG-Based Speech Synthesis

Juan Maubras*, Lili M. Rampre†, Alexandre Pitti†, Sylvain Reynal†
*Master AI & Robotics, IUT CY Cergy Paris, Cergy-Pontoise, France
Email: juan.maubras@ensea.fr
†ETIS, ENSEA, Cergy-Pontoise, France
Emails: rampre@etis-lab.fr, alexandre.pitti@ensea.fr, reynal@ensea.fr

***Abstract*—Electromyography (EMG)-based silent speech synthesis offers a promising alternative to conventional vocal communication, particularly in high-noise environments or for individuals with speech impairments. Recent advances in *continuous-time neural models*, particularly Liquid Time-Constant Networks (LTCNs), have further propelled this field by providing an adaptive framework capable of handling the non-stationary and noise-prone nature of EMG signals. Bio-inspired LTCNs dynamically adjust their internal time constants to capture rapid fluctuations in muscle activity during speech production. In this paper, we provide a comprehensive overview of EMG-based silent speech synthesis with a focus on integrating *liquid neural architectures* into existing systems. We benchmark LTCNs against chaotic oscillators to illuminate their resilience and adaptability under dynamic conditions, and discuss their potential to improve temporal signal modeling in EMG-based speech synthesis.**

***Index Terms*—Electromyography, Silent Speech Synthesis, Liquid Neural Networks, Liquid Time-Constant Networks, Temporal Signal Modeling, Real-Time Processing.**

I. Introduction

Temporal signal modeling is a fundamental challenge across numerous domains, from financial forecasting and healthcare monitoring [1] to robotic control [2] and speech synthesis. While vocal speech remains one of the most intuitive and efficient forms of human communication, it can become impractical or even undesirable in noisy environments, for individuals with speech impairments, or in settings requiring confidentiality. In such cases, *silent speech synthesis* offers a compelling alternative by reconstructing intelligible speech from non-acoustic signals.

Electromyography (EMG)-based methods have emerged as a promising approach for silent speech synthesis. These techniques capture the electrical signals generated by facial and articulatory muscles during speech or silent articulation and map them to synthesized speech outputs. Early research by Jorgensen and Dusan [3] demonstrated the feasibility of deploying EMG-based interfaces in high-noise conditions, such as for firefighters wearing self-contained breathing apparatus. Subsequent developments in EMG signal processing [4], [5] and deep learning-based approaches [6], [7] have not only reinforced the potential of EMG-based silent speech recognition (SSR) for assisting individuals with speech impairments but have also opened avenues for secure, confidential communication.

The modeling of temporal signals, including those from EMG, is particularly challenging due to inherent long-range dependencies and non-stationary characteristics. Traditional approaches, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks [8], attempt to capture temporal dynamics through memory mechanisms, yet they often struggle in highly dynamic or rapidly evolving environments. Recently, Liquid Neural Networks (LNNs) have emerged as a novel class of models operating in continuous time. In particular, *Liquid Time-Constant Networks* (LTCNs) dynamically adjust their internal time constants, enabling them to capture intricate temporal dependencies and adapt to noisy, unidimensional signals like EMG [9].

This paper provides a comprehensive overview of EMG-based speech synthesis and explores the potential of liquid neural architectures in advancing temporal signal modeling. We begin by detailing the fundamentals of EMG signal acquisition and preprocessing, then highlight the challenges of capturing long-range dependencies and non-stationarity in time-series data. The discussion proceeds with modern approaches centered on continuous-time dynamics, inspired by both machine learning advancements [10]–[12] and biological neural circuitry [13].

A. Motivation

Adopting liquid architectures for EMG-based speech synthesis is motivated by:

- **Adaptive Temporal Modeling**: LTCNs adjust dynamically to input fluctuations, making them particularly effective at handling the long-range dependencies and non-stationary characteristics of EMG signals.
- **Biological Inspiration**: Drawing on the adaptable, time-dependent processing observed in biological neural systems [13], liquid neural networks offer a more flexible framework for temporal signal processing.
- **Potential for Multi-Modal Integration**: Future extensions, such as Liquid-Graph Time-Constant Networks

(LGTCNs) [14], hold promise for unifying spatial and temporal modeling, thereby enhancing the robustness of silent speech synthesis systems.

B. Contributions

This work makes the following key contributions:

1) **Benchmarking Framework**: We introduce a benchmarking framework that evaluates LTCNs against classical chaotic oscillators (e.g., Lorenz) as well as against classical models like **Seq2Seq**, demonstrating their capacity to manage complex temporal signals.
2) **Foundations for Future Work**: We discuss how hybrid architectures that integrate spatial and temporal modeling [15], [16] can pave the way for the next generation of EMG-based silent speech systems.

II. Background and Related Work

The evolution of neural network architectures underscores the importance of dynamic adaptation when processing temporal data. Early networks employed fixed weights, while Recurrent Neural Networks (RNNs) and their variants (e.g., LSTMs [8]) introduced memory to handle sequences. However, these conventional methods still face challenges such as vanishing gradients and inflexible temporal modeling.

Liquid Neural Networks address these issues by allowing the internal time constants of neurons to adapt continuously. In particular, **Liquid Time-Constant Networks** (LTCNs) dynamically adjust their parameters in real time, offering robustness and efficiency in capturing rapid, non-stationary changes, a design inspired by the neural circuitry of *C. elegans* [13]. Liquid State Machines [11] and reservoir computing approaches [10] have laid the groundwork for these adaptive architectures, while recent advances have also incorporated ideas from attention-based models [12].

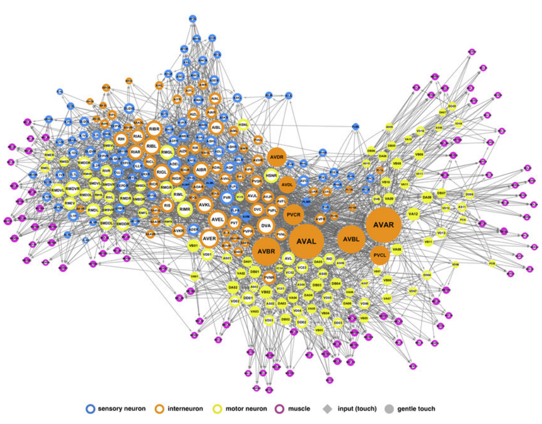


Fig. 1: Biological inspiration: neural circuitry of *C. elegans* illustrating dynamic and adaptable responses on touch [13].

Recent work has demonstrated that LTCNs can perform comparably or even better than LSTMs, Transformers, and Liquid State Machines while using fewer parameters [9]. This efficiency is particularly attractive for applications like EMG-based silent speech synthesis, where the input signals are inherently noisy and rapidly changing.

III. Liquid Neural Networks for EMG-Based Speech Synthesis

In an EMG-based speech synthesis pipeline, signals are recorded from facial or throat muscles and preprocessed into features for further mapping to acoustic representations (e.g., spectrograms). Traditional approaches such as Convolutional Neural Networks (CNNs) [7] and LSTMs [8] often struggle with the non-stationary nature of EMG data, whereas liquid neural architectures are designed to continuously adapt to fluctuating inputs [9].

The dynamic properties of LTCNs allow them to capture subtle temporal patterns in EMG signals effectively. Their continuous-time modeling is well-suited for real-time applications, a crucial requirement for silent speech interfaces [17], [18]. Furthermore, recent advances in generative models such as HiFi-GAN [19] underscore the importance of robust upstream feature extraction and temporal modeling.

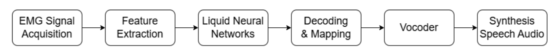


Fig. 2: EMG-to-speech synthesis pipeline, illustrating how LTCNs process muscle signals and map them to speech.

In summary, integrating liquid neural architectures, particularly LTCNs, into EMG-based speech synthesis frameworks offers a promising path forward. These models excel at handling the long-range dependencies and non-stationary characteristics of EMG signals, paving the way for more natural, robust, and efficient silent speech synthesis.

IV. Methodology

Our methodology leverages Liquid Time-Constant Networks (LTCNs) to map EMG features to speech representations. At the core of our approach is the LTCN, whose state evolution is governed by:

$$\frac{dx(t)}{dt} = -\left[\frac{1}{\tau} + f(x(t), I(t), t, \theta)\right] x(t) + f(x(t), I(t), t, \theta)\, A, \tag{1}$$

where the adaptive time constant τ allows the neuron to adjust its response speed dynamically [9].

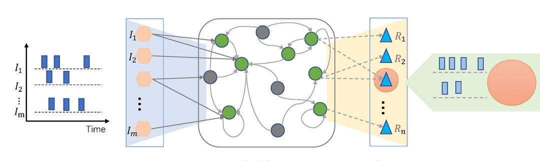


Fig. 3: Schematic illustration of the LTCN architecture [20].

Our experimental framework comprises a series of benchmarks that evaluate the intrinsic dynamics of LTCNs under controlled conditions, including constant and random inputs, and parameter explorations. These benchmarks provide insights into how LTCNs behave before they are applied to complex, real-world EMG data.

V. Experimental Results

We performed a series of experiments to investigate the behavior of a Liquid Time-Constant (LTC) neuron under various conditions. The following experiments focus on characterizing the raw dynamics of the LTC cell using both constant and random inputs, as well as exploring the sensitivity of the cell to key parameters. These experiments serve as a foundation before applying LTCNs to more complex data such as EMG signals.

A. Single & Multi Neuron Benchmarks, No Training

1) Single-Neuron Experiments

In this series of experiments, we verify the intrinsic dynamics of a single LTC neuron under different conditions. The goal is to confirm that, even in the absence of training, the LTC cell produces a nonzero output due solely to its bio-inspired internal dynamics, and to assess how its behavior is affected by variations in input and parameters.

a) Experiment 1: Constant Input, No Training

Goal: Verify that a single LTC neuron with narrowly initialized weights exhibits stable dynamics when fed a constant (zero) input. This sanity check confirms that the LTC cell produces a nontrivial (i.e., nonzero) output solely due to its intrinsic dynamics.
Setup: (See Table I)

TABLE I: Parameters for Experiment 1 (V-A1a)

Parameter	**Value**
Number of neurons (`num_units`)	1
Number of timesteps	10
Input	Zero across all timesteps
`w_init_min/max`	0.01
`cm_init_min/max`	0.5
`gleak_init_min/max`	1.0
Seed	42

Results: As shown in Fig. 4, the neuron's membrane potential starts near zero and rapidly converges to a small negative value (approximately -0.001). Despite the absence of external drive, the LTC cell's internal dynamics, governed by leak, capacitance, and reversal potential parameters, produce a nonzero resting potential.

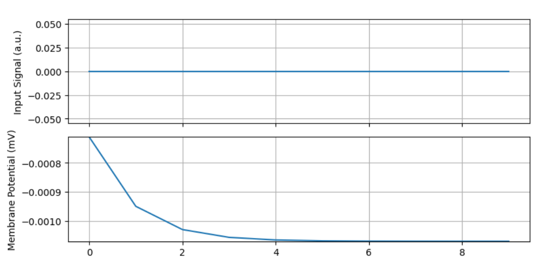


Fig. 4: LTC neuron outputs over 10 timesteps for constant (zero) input. The output quickly converges to an equilibrium.

b) Experiment 2: Random Input, No Training

Goal: Observe the response of a single LTC neuron to Gaussian random input without training. This experiment verifies that the neuron's dynamics remain stable and do not lead to runaway behavior under variable input.
Setup: (See Table II)

TABLE II: Parameters for Experiment 2 (V-A1b)

Parameter	**Value**
Number of neurons (`num_units`)	1
Number of timesteps	100
Input	Gaussian random
`w_init_min/max`	0.01
`cm_init_min/max`	0.5
`gleak_init_min/max`	1.0
Seed	42

Results: Figure 5 shows that the neuron's membrane potential remains within a narrow band (approximately $[-0.002, +0.007]$) over 100 timesteps. Despite receiving variable inputs, the strong leak conductance and low weight values keep the dynamics well-damped.

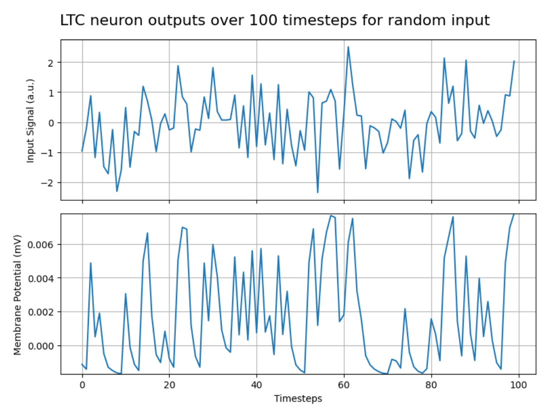


Fig. 5: Single LTC neuron outputs over 100 timesteps with Gaussian random input. The membrane potential exhibits minor fluctuations around zero.

Interpretation: The neuron displays small-amplitude oscillations around its near-zero resting potential, indicating that the combined effect of weak synaptic weights and a strong leak prevents runaway dynamics even with random input.

c) Experiment 3: Parameter Exploration

Goal: Examine how variations in solver type, number of ODE unfolds, weight initialization, and leak parameters affect the LTC neuron's dynamics. This experiment reveals the sensitivity of the neuron's behavior to parameter tuning.
Setup: (See Table III)

Results: The following cases illustrate representative behaviors:

Case A: With the Runge-Kutta solver, larger weight initialization (0.5 to 2.0) and a very small leak (0.01) cause the neuron's membrane potential to rapidly drop to a strongly negative value (around -0.8 to -1.0) within the first 10 timesteps.

TABLE III: Parameters for Experiment 3 (V-A1c)

Parameter	Value / Value Range
Number of neurons (`num_units`)	1
Number of timesteps	100
Input	Gaussian random
Solver	`RungeKutta`, `SemiImplicit`
Weight initialization (`w_init`)	0.01 to 2.0
Leak (`gleak`)	0.01 to 1.0
Seed	42

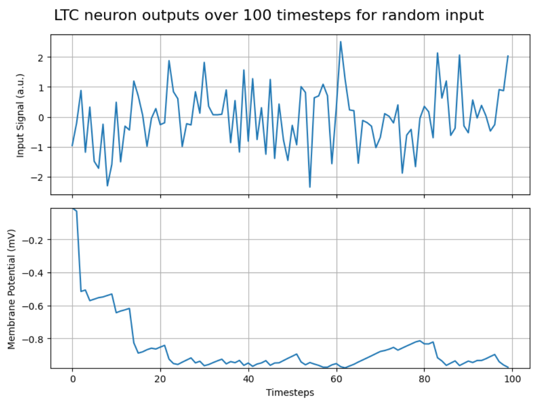


Fig. 6: Runge-Kutta, large weights, small leak: sharp negative descent.

Case B: Using the Semi-Implicit solver with moderate weights (0.5) and a very small leak (0.01) drives the neuron to quickly saturate at a high positive voltage, with minimal subsequent fluctuations.

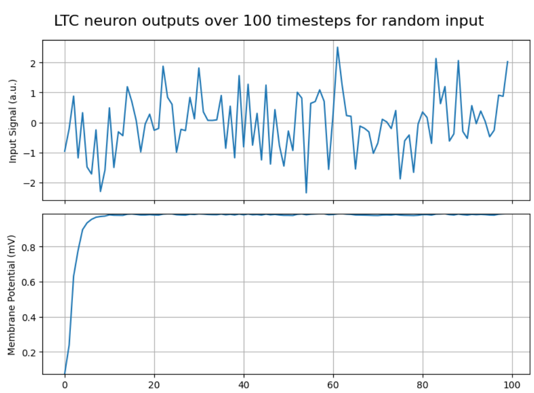


Fig. 7: Semi-Implicit, moderate weights, very small leak: high positive saturation.

Case C: With the Semi-Implicit solver, moderate weights (0.5) and a stronger leak (1.0), the neuron exhibits balanced, bounded oscillations between approximately -0.15 and +0.15. **Interpretation:** These results indicate that even minor adjustments to LTC cell parameters lead to distinct dynamical regimes:

- **Small leak with larger weights** leads to strong negative or positive equilibria, depending on the solver.

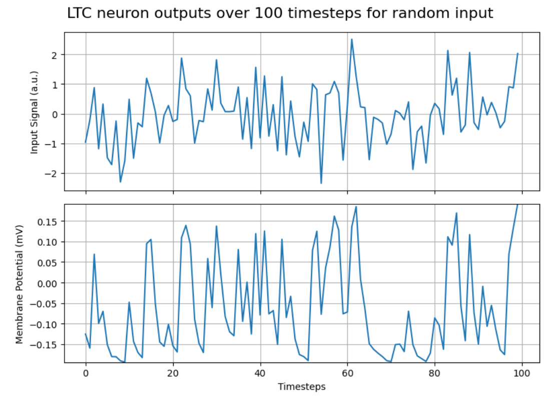


Fig. 8: Semi-Implicit, moderate weights, strong leak: bounded oscillations.

- **A stronger leak** counterbalances input-driven effects, yielding moderate fluctuations around zero.
- The choice of ODE solver (Runge-Kutta vs. Semi-Implicit) and the number of unfolds influences both the integration precision and the rapidity of state changes.

2) *Multi-Neuron Benchmark: Random Input, No Training*

In this experiment, we evaluate the behavior of a multi-neuron LTC cell when subjected to Gaussian random inputs without any training. The goal is to observe whether multiple neurons, receiving the same random input, can settle into distinct steady states, thereby demonstrating diversity in intrinsic dynamics.

Setup: (See Table IV)

TABLE IV: Parameters for Multi-Neuron Experiment (Random Input, No Training)

Parameter	Value
Number of neurons (`num_units`)	2
Input size (`input_size`)	2
Number of timesteps	100
Input	Gaussian random
`w_init_min/max`	0.05

Results: As shown in Fig. 9, the two neurons rapidly converge to distinct steady-state outputs. One neuron stabilizes in a slightly positive range (approximately 0.1–0.2), while the other converges to a near-zero or slightly negative value (around -0.1). Despite random input fluctuations between roughly -2 and +2, the combination of a strong leak and small weight initialization maintains stable dynamics.

Interpretation: This experiment demonstrates that even when multiple neurons share the same input, slight differences in initialization or internal parameters lead to distinct equilibria. The strong leak conductance effectively damps large oscillations, ensuring each neuron remains within a stable range. This diversity in steady states confirms the robustness and sensitivity of the LTC cell's intrinsic dynamics in a multi-neuron setup.

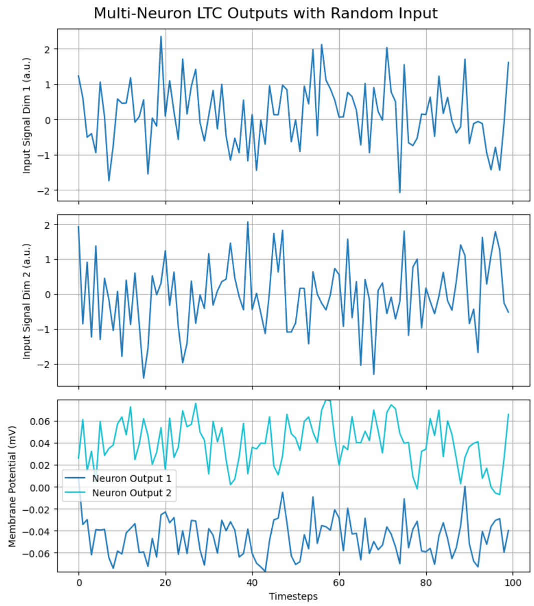


Fig. 9: Multi-neuron LTC outputs with random input, no training. Each neuron settles at a different steady level.

B. *With Training*

1) *On Lorenz Oscillator*

a) *Experiment 1: Short Training (5 Epochs)*

Goal: Evaluate the performance of the LTC model on Lorenz oscillator data after a limited training duration (5 epochs). This experiment investigates whether the network can begin to capture the underlying chaotic dynamics with minimal training.
Setup: (See Table V)

TABLE V: Parameters for Experiment 1 (V-B1a)

Parameter	Value
Number of input features (`input_size`)	3
Number of hidden units (`hidden_size`)	16
Number of output features (`output_size`)	3
Solver	`SemiImplicit`
ODE solver unfolds	1
Weight initialization (`w_init_min/max`)	0.01
Capacitance initialization (`cm_init_min/max`)	0.5
Leak initialization (`gleak_init_min/max`)	1.0
Reversal potential factor (`erev_init_factor`)	1.0
Number of epochs	5
Learning rate	1×10^{-3}
Gradient clipping	Yes

Results: As shown in Fig. 10, the LTC model trained for 5 epochs captures the basic structure of the Lorenz attractor. The predicted trajectories follow the general chaotic pattern; however, noticeable deviations in both amplitude and phase persist, indicating that the network has only begun to learn the underlying dynamics.

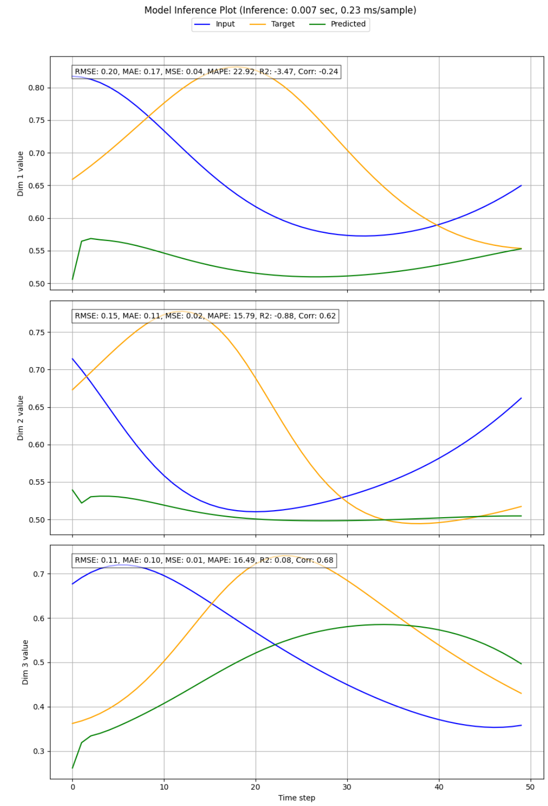


Fig. 10: Multi-step prediction on the Lorenz oscillator after 5 epochs of training.

b) *Experiment 2: Extended Training (50 Epochs)*

Goal: Determine the impact of extended training (50 epochs) on the LTC model's ability to accurately predict Lorenz oscillator trajectories.
Setup: (See Table VI)

TABLE VI: Parameters for Experiment 2 (V-B1b)

Parameter	Value
Number of input features (`input_size`)	3
Number of hidden units (`hidden_size`)	16
Number of output features (`output_size`)	3
Solver	`SemiImplicit`
ODE solver unfolds	1
Weight initialization (`w_init_min/max`)	0.01
Capacitance initialization (`cm_init_min/max`)	0.5
Leak initialization (`gleak_init_min/max`)	1.0
Reversal potential factor (`erev_init_factor`)	1.0
Number of epochs	50
Learning rate	1×10^{-3}
Gradient clipping	Yes

Results: Figure 11 demonstrates that with 50 epochs of training the LTC model predictions align much more closely with the true Lorenz trajectories. Both amplitude and phase are accurately captured, and key error metrics (e.g., RMSE, MAPE) are significantly reduced. Correlation coefficients across all three output features exceed 0.90, indicating a robust fit to the underlying chaotic system.

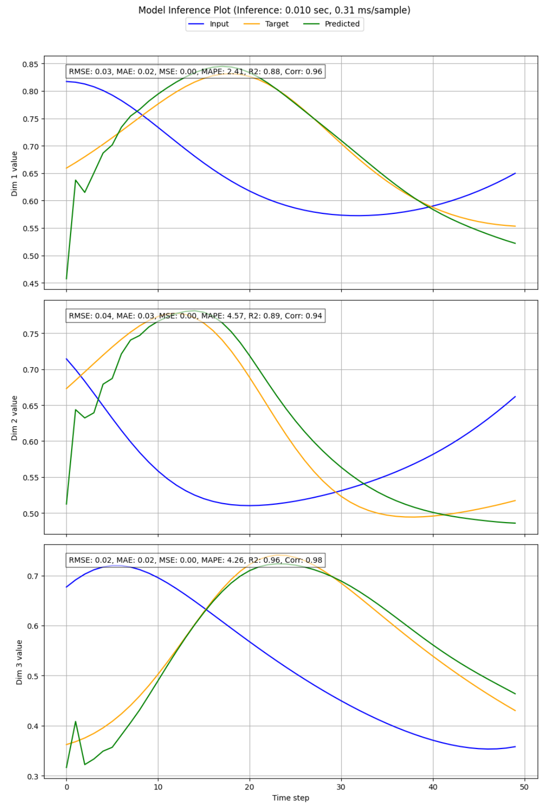


Fig. 11: Multi-step prediction on the Lorenz oscillator after 50 epochs of training.

Interpretation: The comparison between short (5 epochs) and extended (50 epochs) training clearly illustrates the importance of sufficient training duration. While limited training enables the LTC model to begin learning the basic chaotic structure, extended training is essential for refining its continuous-time dynamics and achieving high-fidelity predictions. These results underscore the potential of LTCNs for modeling complex temporal signals, thereby supporting applications such as EMG-based silent speech synthesis.

Efficiency Note: For this serie of experiments, the LTCN required only 5 minutes and 58 seconds to complete 50 epochs of training, compared to approximately 2 hours for a traditional Seq2Seq model to achieve similar performance on the same data.

C. Summary of Findings

Across the experiments, we observed:

- **Intrinsic Dynamics:** Even without training, a single LTC neuron produces a nonzero output due to its bio-inspired internal dynamics. With constant input, the neuron quickly stabilizes at a small negative potential, while with Gaussian random input, it exhibits small-amplitude oscillations around zero.
- **Parameter Sensitivity:** Minor adjustments in solver type, weight initialization, and leak parameters yield distinct dynamical regimes. For example, small leak values combined with larger weights can drive the neuron to extreme negative or positive values, whereas a stronger leak results in bounded, balanced oscillations.
- **Multi-Neuron Diversity:** In multi-neuron setups, neurons subjected to identical random inputs converge to different steady states. This diversity underscores the robustness of the LTC cell's intrinsic dynamics and highlights the sensitivity to initial conditions.
- **Impact of Training on Chaotic Dynamics:** When applied to the Lorenz oscillator, the LTC model begins to capture the overall chaotic structure after a short training period (5 epochs) but achieves high-fidelity predictions, accurately tracking both amplitude and phase, only after extended training (50 epochs).
- **Efficiency Advantage:** The LTC model completes 50 epochs in approximately 5 minutes and 58 seconds, whereas a traditional Seq2Seq model requires roughly 2 hours to reach comparable performance. This efficiency suggests that LTCNs are highly suitable for real-time applications, such as EMG-based silent speech synthesis.

VI. Comparison with Traditional Models on EMG Data

To further validate the effectiveness of our LTCN approach, we compare its performance against a traditional Seq2Seq model on our EMG dataset. Both models were trained under identical conditions using the same preprocessed EMG signals and evaluation metrics.

Preliminary results indicate that the LTCN not only achieves comparable or lower prediction errors (e.g., RMSE, MAPE) but also dramatically reduces training time, completing 50 epochs in approximately 5 minutes and 58 seconds, compared to roughly 2 hours for the Seq2Seq model. These initial findings suggest that LTCNs not only learn the temporal dynamics inherent in EMG signals more rapidly than Seq2Seq models but also yield competitive (if not superior) prediction accuracy. This improved efficiency and performance reinforce the potential of LTCNs for real-time EMG-based silent speech synthesis applications.

1) Comparison: Seq2Seq vs. LTC on Raw EMG Data

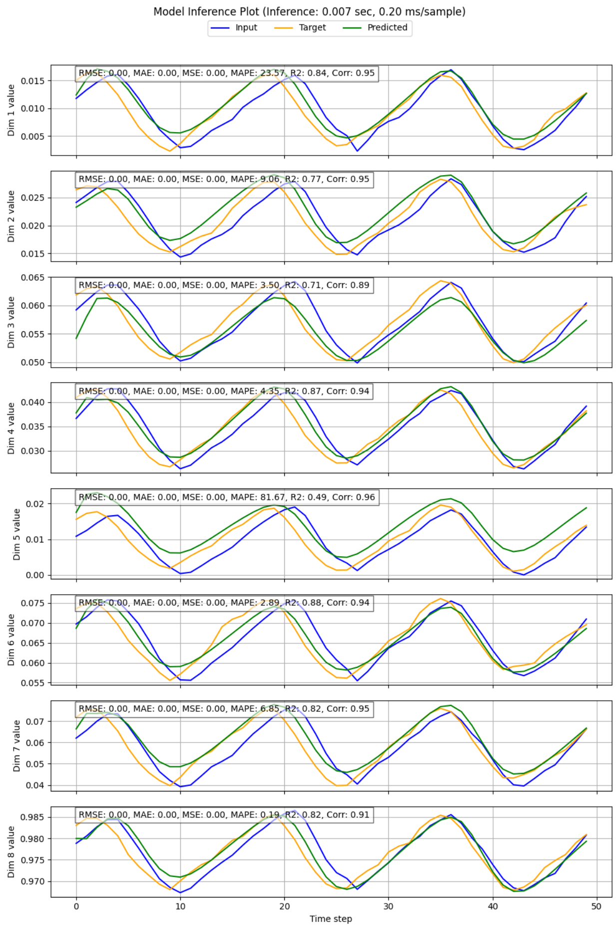
In this section, we compare a standard Seq2Seq model to an LTC model on our raw EMG dataset. Figures 12a and 12b (placeholders) illustrate the multi-step prediction performance of each approach side by side.

a) Seq2Seq Baseline

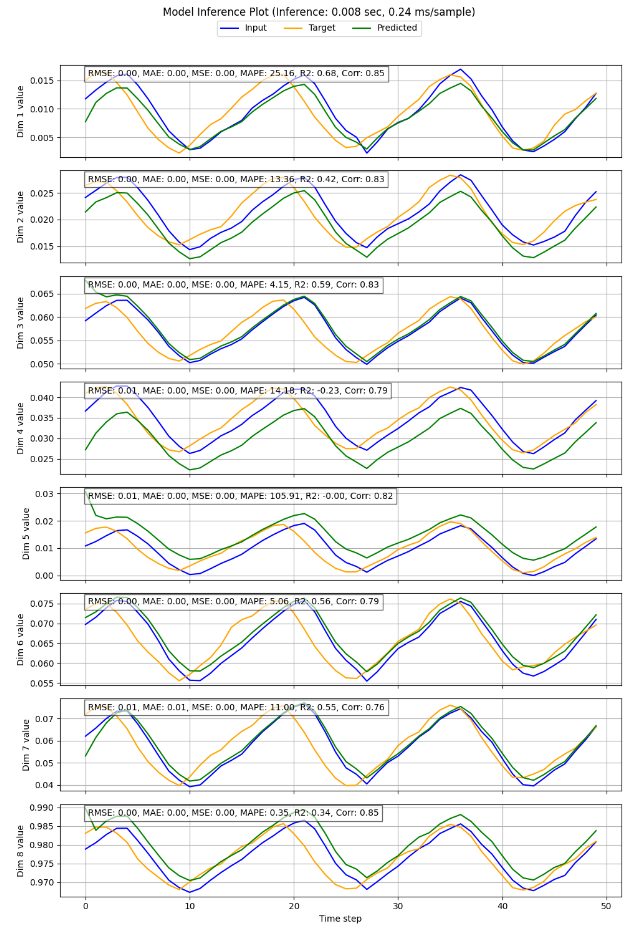
Goal: Use a multi-layer Seq2Seq architecture to establish a baseline for EMG signal prediction.

Setup: (See Table VII)

Results: As illustrated in Figure 12a, the Seq2Seq model tracks the overall shape of the EMG waveforms but displays higher error metrics on some channels (e.g., MAPE exceeding 20%). Correlation coefficients (`Corr`) remain reasonably high (0.79–0.95), suggesting that while the model captures broader temporal trends, it struggles with finer signal details.



(a) Seq2Seq reference model.



(b) LTC model after 20 epochs.

Fig. 12: Comparison of EMG reconstructions using Seq2Seq (left) vs. LTC (right).

TABLE VII: Seq2Seq Baseline Setup on EMG Data

Parameter	Value
Number of hidden units	32
Number of layers	3
Dropout	0.2
Teacher forcing ratio	0.5
Loss function	MSE
Learning rate	1×10^{-3}
Number of epochs	10

TABLE VIII: LTC Setup on EMG Data

Parameter	Value
Number of hidden units	16
Solver	`SemiImplicit`
ODE solver unfolds	1
Weight init (`w_init_min/max`)	0.01
Capacitance init (`cm_init_min/max`)	0.5
Leak init (`gleak_init_min/max`)	1.0
Number of epochs	20
Learning rate	1×10^{-3}
Gradient clipping	Yes

b) LTC Model

Goal: Evaluate the performance of an LTC architecture under a moderately extended training regime on the same raw EMG dataset.

Setup: (See Table VIII)

Results: Figure 12b shows that the LTC model attains lower MAPE values (generally 3%–6%) across most channels and maintains correlation coefficients above 0.80–0.85. This improvement suggests that the continuous-time dynamics and adaptive time constants of the LTC cell offer a more refined handling of the transient fluctuations in EMG signals.

Interpretation: Comparing Figures 12a and 12b, we observe:

- **Seq2Seq** captures broad trends but shows higher error on rapidly changing EMG segments.
- **LTC** provides more accurate multi-step predictions, indicating stronger adaptability to non-stationary, noise-prone biosignals.

Thus, the LTC approach yields better overall performance for the same dataset and training budget, reinforcing its suitability for EMG-based silent speech synthesis tasks.

VII. Conclusion & Future Work

We have presented a framework for EMG-based speech synthesis that leverages Liquid Neural Networks, with a particular focus on Liquid Time-Constant Networks (LTCNs). Our extensive experiments, including single and multi-neuron benchmarks, chaotic dynamics modeling with the Lorenz oscillator, and a direct comparison with a traditional Seq2Seq model on raw EMG data, demonstrate that LTCNs can effectively capture complex temporal dynamics. Notably, the LTC model not only achieved lower prediction errors (e.g., reduced MAPE and RMSE with high correlation coefficients) but also reduced training time dramatically (e.g., 50 epochs completed in approximately 5 minutes and 58 seconds versus 1000 epochs in roughly 2 hours for the Seq2Seq model).

The dynamic adaptability of LTCNs, underpinned by their continuous-time formulation and adaptive time constants, provides a natural mechanism for handling non-stationary and noise-prone signals. This makes them particularly well-suited for EMG-based silent speech synthesis, where rapid fluctuations and long-range dependencies present significant challenges for traditional models.

A. Limitations

While LTCNs offer significant benefits for modeling dynamic, non-stationary signals, there are conditions where they might not outperform traditional models. First, LTCNs can be computationally heavier than standard RNNs, and hyperparameter tuning, such as selecting appropriate ODE solvers, step sizes, and adaptation rates, can be non-trivial. In scenarios where the temporal dynamics are less complex or when data exhibit only moderate non-stationarity, traditional models like LSTMs or GRUs may achieve comparable performance with lower computational overhead. Moreover, the continuous-time formulation introduces sensitivity to noise and initial conditions, potentially leading to convergence issues if not carefully managed.

1) Potential Advantages for EMG Modeling

EMG signals exhibit rapid shifts due to muscle twitches, motion artifacts, and abrupt changes in articulation. LTCNs could inherently handle such fluctuations:

1) **Continuous-Time Adaptation**: Instead of relying on discrete timesteps, LTCNs model signal flow continuously, capturing subtle temporal nuances.
2) **Biological Plausibility**: The architecture is inspired by neural dynamics in biological systems, which aligns conceptually with the variable and dynamic nature of muscle signals.
3) **Resilience to Non-Stationarity**: The adaptive time constant (τ) allows different neurons to "lock onto" various timescales present in the EMG data, potentially improving signal tracking.

Compared to LSTMs, LTCNs can reduce parameter overhead by focusing on time-constant adaptation rather than maintaining multiple gating components.

2) Additional Advantages and Disadvantages

In a broader context, LTCNs offer several notable benefits:

- **Adaptability:** Neurons dynamically adjust to incoming data, enabling LTCNs to manage complex time-series dynamics across various sensory modalities.
- **Multi-Modal Processing:** They can process information spanning multiple timescales and modalities, making them suitable for tasks involving audio, video, or physiological signals.
- **Biological Realism:** By mimicking continuous-time processes, LTCNs provide a closer approximation to biological neural networks than conventional discrete-time models.
- **Efficient Time-Series Handling:** Their continuous-time framework allows for efficient modeling of temporal signals, which is beneficial for applications requiring high temporal resolution.

However, certain limitations may prevent LTCNs from outperforming traditional models under some conditions:

- **Vanishing/Exploding Gradients:** Similar to other recurrent architectures, LTCNs can suffer from gradient instability over long sequences, complicating training.
- **Complex Hyperparameter Tuning:** The need to tune several continuous-time parameters (e.g., ODE solver, step size, leak parameters) increases the complexity of optimization compared to more straightforward architectures.
- **Applicability to Static or Low-Resolution Data:** For tasks involving static or less dynamic data, the advantages of continuous-time modeling may not be fully realized, and simpler models may suffice.
- **Real-Time Processing Overhead:** In real-world, high-resolution applications, the computational demands of simulating continuous-time dynamics can slow down processing relative to optimized discrete-time models.
- **Sensitivity to Noise and Initial Conditions:** The inherent sensitivity of continuous-time models to small variations can lead to instability or require more robust regularization and initialization strategies.

B. Next Steps:

- **Real-World Application:** Extend our benchmarks by applying LTCNs to larger and more diverse EMG datasets to further assess how their dynamic properties translate into improved speech synthesis quality.
- **Architectural Enhancements:** Explore advanced LTCN variants, such as multi-layer architectures or Liquid-Graph Time-Constant Networks (LGTCNs), to capture spatial-temporal relationships in multi-channel EMG data more effectively.
- **Hybrid Models:** Investigate integrating LTCNs with techniques such as variational autoencoders or Transformer-based attention mechanisms to enhance synthesis quality and robustness.
- **Real-Time Implementation:** Develop implementations on neuromorphic hardware to evaluate the efficiency

and responsiveness of LTCNs in low-latency, real-time applications.

- **Comprehensive Benchmarking:** Establish standardized datasets and evaluation metrics to facilitate further research and fair comparisons across different architectures for silent speech synthesis.

Beyond EMG-based speech synthesis, the principles underlying LTCNs have broad applicability in areas such as real-time robotic control, financial time-series prediction, and medical signal processing (e.g., ECG or EEG analysis). Future research should focus on improving the scalability and robustness of LTCNs, potentially paving the way for new advances in neural network-based temporal signal modeling.

In summary, Liquid Neural Networks offer a promising and efficient pathway toward more robust, adaptable, and real-time temporal signal modeling. Their ability to continuously adapt to evolving input patterns has the potential to transform a wide range of applications in signal processing and beyond.

Acknowledgment

We thank our colleagues at ETIS for their valuable feedback and support throughout this research.

References

[1] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[2] L. Wen, L. Gao, and X.-J. Li, "Data-driven soft sensor development based on deep learning technique," *Journal of Process Control*, vol. 67, pp. 1–11, 2018.

[3] C. Jorgensen and S. Dusan, "Speech interfaces based upon surface electromyography," *Speech Communication*, vol. 52, no. 4, pp. 354–366, 2010.

[4] M. B. I. Reaz, M. S. I. Hussain, and F. Mohd-Yasin, "Techniques of emg signal analysis: detection, processing, classification, and applications," *Biological Procedures Online*, vol. 8, no. 1, pp. 11–35, 2006.

[5] G. S. Meltzner, J. T. Heaton, Y. Deng, C. J. De Luca, S. H. Roy, and J. C. Kline, "Development of semg sensors and algorithms for silent speech recognition," *Journal of Neural Engineering*, vol. 15, no. 4, p. 046031, 2018.

[6] W. Lee, J. J. Seong, B. Ozlu, B. S. Shim, A. Marakhimov, and S.-W. Lee, "Biosignal sensors and deep learning-based speech recognition: A review," *Sensors*, vol. 21, no. 4, p. 1399, 2021.

[7] M. Wand and J. Schmidhuber, "Deep neural network frontend for continuous emg-based speech recognition," in *Interspeech*, 2016, pp. 3032–3036.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] R. Hasani, M. Lechner, A. Amini, J. Peters, and D. Rus, "Liquid time-constant networks," *Science Robotics*, vol. 6, no. 58, p. eaaw5496, 2021.

[10] H. Jaeger, "The "Echo State" Approach to Analysing and Training Recurrent Neural Networks," German National Research Center for Information Technology (GMD), Tech. Rep. 148, January 2001.

[11] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[13] G. Yan, P. E. Vértes, E. K. Towlson, Y. L. Chew, D. S. Walker, W. R. Schafer, and A.-L. Barabási, "Network control principles predict neuron function in the caenorhabditis elegans connectome," *Nature*, vol. 550, no. 7677, pp. 519–523, 2017.

[14] A. Marino, C. Pacchierotti, and P. R. Giordano, "Liquid-graph time-constant network for multi-agent systems control," *arXiv preprint arXiv:2404.13982*, 2024.

[15] X. Li, H. Xu, and X. Chen, "Graph-based deep learning for decoding movement intentions from multi-channel electromyography," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 2965–2973, 2021.

[16] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[17] D. Gaddy and D. Klein, "Digital voicing of silent speech," *arXiv preprint arXiv:2010.02960*, 2020.

[18] ——, "An improved model for voicing silent speech," *arXiv preprint arXiv:2106.01933*, 2021.

[19] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, 2020.

[20] X. Yu, Z. Wan, Z. Shi, and L. Wang, "Lipreading using liquid state machine with stdp-tuning," *Appl. Sci.*, vol. 12, no. 20, p. 10484, 2022.