Neural Localization In Non-Linear Stochastic Environment

Adyant Srinivasan
dept. of ECE, UCSB
5774443
adyant@ucsb.edu

Jash Shah
dept. of ECE, UCSB
A3H4H35
jashshah@ucsb.edu

Pratyush Sinha
dept. of ECE, UCSB
A2M1R38
psinha@ucsb.edu

***Abstract*—We aim to implement a transparent object localization and tracking algorithm for non-linear dynamic models in an arbitrarily modeled noisy environment. The proposed particle filters using spiking neurons do not rely on the traditional weighted re-sampling update step, and are thus computationally prone from sampling variances and weight-decay issues. Furthermore, we propose that our architecture is computationally faster for a small number of particles, because of its event-driven nature. Finally, we show how this modeling allows the network to capture complex relationships, update online, and handle noisy data compared to traditional approaches.**

***Index Terms*—Particle Filtering, Spiking Neuron Architecture(SNN), Localization, Stochastic Noise**

I. Topic Introduction

In a dynamic noisy environment, where temporal varying signals need to be estimated online, using the previous observations, a recursive nonlinear Bayesian inference is often known as 'filtering'. A standard filtering approach uses prior observations to estimate posterior state estimations using a known set of stochastic distributions [1]. But, Particle filter is a non-parametric and brute force approach which can model arbitrary distributions with non-linear dynamics. Two majorly known bottlenecks in such approaches are: 1. The curse of dimensionality and 2. Bayesian weight disparity, leading to weight collapse. SNN based event-driven neural approaches have a general tendency to filter out temporal irrelevant intricacies with less computational load. But the discrete dynamics of such methods suffer from the lack to feedback updated weights and implement re-sampling paradigms.

II. Related Work

The paper [2] presents the concept of the Neural Particle Filter (NPF), which is considered an improvement over the standard particle filter. NPF is more efficient at handling complex, high-dimensional problems, especially when dealing with limited particle numbers. Using the powerful learning capabilities of neural networks, we can better approximate posterior distributions and address the "curse of dimensionality" that traditional particle filtering methods often face. This enables NPF to learn complex data relationships without requiring explicit model design, making it more versatile in various scenarios. NPF is particularly effective in handling problems with many variables, where traditional particle filters may struggle due to the need for a large particle set. Furthermore, NPF can continuously adapt to changing conditions by learning from the incoming data.

III. Proposed Method

A general methodology for particle filtering is presented in Algorithm 1. We propose 2 SNN-adaptations of particle filtering. The first method involves mimicking the classical approach in a spiking environment using a 3-layer LIF neuron structure each layer accounting for the input dynamics, particle dynamics, and weight dynamics, respectively. This method eliminates the use of weighted-resampling and outputs the particle belief directly. The spiking modeling has a natural tendency to eliminate temporal noise over time. This proposed simplex model can reduce computational complexity but is a one-shot learning algorithm which can raise concerns in cases where multiple particles report similar beliefs.

Algorithm 1 SNN Particle Filter Algorithm

Initialize a neurons $N_1 = \{n_1^{(0)}, ..., n_N^{(0)}\}$ with the state space $X_0 = \{x_1^{(0)}, ..., x_N^{(0)}\}$ drawn from the prior $p(x_0)$.
for $t = 1$ to T **do**
  for $i = 1$ to N **do**
    Sample Input $N_0^{(t)}$ to update the layer N_1 using motion model $\sim p(x_t|x_i^{(t-1)}, u_t)$.
    Sample Input "Layer 0" to provide the sensor readings to Calculate likelihood: $w_i^{(t)} = w_i^{(t)} \cdot p(z_t|x_i^{(t)})$.
  end for
  for $i = 1$ to N **do**
    Estimate the output belief based on the spiking activity, by normalizing the weights: $w_i^{(t)} = \frac{w_i^{(t)}}{\sum_{j=1}^{N} w_j^{(t)}}$
  end for
  The output layer depicts plasticity, to update all other output neurons to follow the spiking neuron.
end for

The second proposal aims to eliminate this issue by developing a recurrent-SNN to account for the filter dynamics, thus updating the weighted beliefs directly through the network. This method increases the system complexity but is much more robust for a complex environment.

Future Works

The proposed SNN-filtering methodologies does not fully utilize the temporal advantages of a spiking system, which can furthermore reduce the time complexity of the system. Furthermore, an extensive research on model-free approaches could adhere better results cultivating from a bias-free learning curve.

Acknowledgments

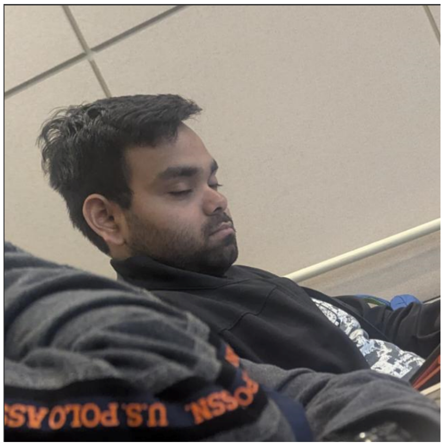

Fig. 1. Pratyush in the Lecture

References

[1] Juárez-Lora A, García-Sebastián LM, Ponce-Ponce VH, Rubio-Espino E, Molina-Lozano H, Sossa H. Implementation of Kalman Filtering with Spiking Neural Networks. Sensors. 2022; 22(22):8845.

[2] Yamazaki K, Vo-Ho VK, Bulsara D, Le N. Spiking Neural Networks and Their Applications: A Review. Brain Sci. 2022 Jun 30;12(7):863. doi: 10.3390/brainsci12070863. PMID: 35884670; PMCID: PMC9313413.

[3] Ma, X., Karkus, P., Hsu, D., & Lee, W.S. (2019). Particle Filter Recurrent Neural Networks. ArXiv, abs/1905.12885.