

PAPER • OPEN ACCESS

Range and angle estimation with spiking neural resonators for FMCW radar

To cite this article: Nico Reeb *et al* 2025 *Neuromorph. Comput. Eng.* **5** 024009

View the [article online](#) for updates and enhancements.

You may also like

- [Broadband terahertz time-domain spectroscopy and fast FMCW imaging: Principle and applications](#)
Yao-Chun Shen, , Xing-Yu Yang et al.
- [Silicon-based FMCW signal generators: A review](#)
Wei Deng, Haikun Jia and Baoyong Chi
- [A lightweight interference suppression technique for FMCW automotive radar](#)
Tai-Hsing Lee and Nur Azis Salim



PAPER

OPEN ACCESS

RECEIVED

29 November 2024

REVISED

1 April 2025

ACCEPTED FOR PUBLICATION

17 April 2025

PUBLISHED

8 May 2025

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Range and angle estimation with spiking neural resonators for FMCW radar

Nico Reeb^{*} , Javier Lopez-Randulfe , Robin Dietrich and Alois C Knoll

TUM School of Computation, Information and Technology, Technical University of Munich, Munich, Germany

^{*} Author to whom any correspondence should be addressed.E-mail: nico.reeb@tum.de**Keywords:** neuromorphic computing, automotive radar, spiking neural network, Fourier transform

Abstract

Automotive radar systems face the challenge of managing high sampling rates and large data bandwidth while complying with stringent real-time and energy efficiency requirements. Neuromorphic computing offers promising solutions because of its inherent energy efficiency and parallel processing capacity. Yet, most sensor systems, such as radars, do not produce suitable data for further neuromorphic processing of the data. This research presents a novel spiking neuron model for signal processing of frequency-modulated continuous wave (FMCW) radars that outperforms the state-of-the-art spectrum analysis algorithms in latency and data bandwidth. These spiking neural resonators are based on the resonate-and-fire neuron model and optimized to dynamically process raw radar data while simultaneously emitting an output in the form of spikes. We designed the first neuromorphic neural network consisting of these spiking neural resonators that estimates range and angle from FMCW radar data, evaluated the network on simulated automotive datasets and compared the results with a state-of-the-art pipeline for radar processing. The proposed neuron model significantly reduces the processing latency compared to traditional frequency analysis algorithms, such as the Fourier transformation (FT), which needs to sample and store entire data frames before processing. The evaluations demonstrate that these spiking neural resonators achieve state-of-the-art detection accuracy while emitting spikes simultaneously to processing and transmitting only 0.02% of the data compared to a float-32 FT. The results showcase the potential for neuromorphic signal processing for FMCW radar systems and pave the way for designing neuromorphic radar sensors.

1. Introduction

As society and industry depend on increasingly complex signal processing systems, solutions become more energy-intensive. This trend drives the need for optimized processing pipelines that minimize energy consumption and data bandwidth while improving performance and reliability [1]. The automotive industry is paradigmatic due to the limited energy availability in cars and the need for precise and reliable low-latency systems to achieve fully autonomous driving [2, 3]. One of the most critical operations in advanced driving assistance systems (ADASs) is sensor signal processing. Subsequent crucial tasks, such as navigation, collision avoidance, or adaptive cruise control, depend on accurate and fast sensor data processing. The frequency-modulated continuous-wave radar (FMCW) sensor is a prominent example utilized in ADASs. Its low cost and long-range detection, combined with its robustness to bad lighting and weather conditions, make it a fundamental element for perceiving the car's surroundings [4]. Common radar processing techniques are, however, energy intensive due to the high data throughput.

Neuromorphic computing is a promising approach to creating low-power processing systems by mimicking the efficiency of the human brain, which processes sensor data and makes decisions while consuming just 20 W [5–7]. This efficiency stems from parallel, asynchronous, and event-driven processing, with communication through binary spikes. Spiking neural networks (SNNs) are a key neuromorphic

algorithm designed to replicate these brain characteristics [8]. These networks consist of artificial neurons that transmit information via spikes, driven by incoming data. Research in SNNs aims, among others, to improve understanding of the brain [9], to replace deep neural networks (DNNs) [10], optimize data center operations [3], and to develop algorithms for embedded systems [11]. The most common neuron model in SNNs is the integrate-and-fire (IF) model, which is widely used in neuromorphic chips due to its effectiveness in DNN conversion [10, 12, 13], biologically inspired learning [14], and optimization problems [12]. To build an energy-efficient neuromorphic pipeline for radar processing, low-level signal processing is needed to process and convert radar data directly into suitable spike trains for high-level neuromorphic processing.

Object detection with FMCW radars is based on the frequency analysis of the continuous analog signal generated by the sensor. Traditional signal processing methods typically convert discrete time sampled data into the frequency domain, with the Fourier transform (FT) being the most widely used technique for frequency analysis. Some neuromorphic algorithms have already emerged for computing the frequency spectrum. Jiménez-Fernández *et al* [15] applied sequential spiking band-pass filters to process audio signals based on their frequency spectrum. López-Randulfe *et al* [16, 17] introduced an electric circuit and a neuron model that encode analog signals to temporal spikes and provide a mathematically equivalent representation of the FT. Nevertheless, these approaches still buffer the input data leading to an increased memory usage and latency.

Izhikevich proposed the resonate-and-fire (RF) neuron model, which enables continuous frequency analysis on streaming input data [18]. The researchers in [19–21] studied the application of RF neurons to oscillating signals. Previous approaches utilizing RF neurons focused on one-dimensional frequency analysis, whereas FMCW radar processing relies on multi-dimensional frequency analysis for angle and velocity information. To the best of our knowledge there is no research on how resonating neurons compare with classic frequency analysis methods, such as the FT, on multi-dimensional FMCW radar data.

In this work, we advance the RF neuron model [18] to concurrently process incoming radar data and optimally convey information through spiking. Traditional frequency analysis techniques, such as the FT, calculate the spectrum after storing all data samples in a given time window. By allowing concurrent processing and spiking, we can achieve three benefits over traditional approaches:

- (1) Reduce the latency of target detection, as each data sample incoming updates the estimate.
- (2) Reduce data bandwidth due to sparse spiking.
- (3) Remove the need for memory storage for sensor data due to the immediate processing by the neurons.

We arranged these neural resonators into a single layer capable of performing two-dimensional frequency analysis on FMCW radar data allowing to estimate range and angle simultaneously (section 3). To overcome the limited availability of labeled raw radar data, we simulated an automotive FMCW radar datasets with exact positional information of point targets, a level of accuracy not available in real radar datasets (section 4). In section 5, we evaluated the approach on these datasets, and compared the results with the FT method. Finally, we applied the spiking neural resonators to public radar datasets, allowing us to visually assess the model's generalization capabilities (section 5.4). We implemented the neuron model on GPU and published the code¹.

2. Background on FMCW radar sensors

Typically, automotive radar systems rely on multiple-input-multiple-output FMCW radar sensors with multiple transmitting as well as receiving antennas. This enables a high range and angular resolution as well as unambiguous velocity detection. Various multiplexing methods [22–24] enable simultaneous transmission from multiple antennas while maintaining orthogonality between the signals. The combination and mixing of multiple transmit and receive antennas leads to $N_{\text{vx}} = N_{\text{tx}} \times N_{\text{rx}}$ virtual antennas. The m th virtual antenna returns an intermediate-frequency (IF) signal $x_m(t)$, leading to a continuous data vector $\vec{x}(t) \in \mathbb{C}^{N_{\text{vx}}}$ for the entire radar sensor (see figure 1). Throughout the paper, we assume sawtooth frequency modulation and a simplified antenna layout with one transmitting antenna and $N_{\text{rx}} = N_{\text{vx}}$ receiving antennas in one line parallel to the ground. Reflections of K objects lead to an IF signal that can be modeled for one chirp $t \in [0, T_c]$ as

$$x_m(t) = \sum_k^K a_k e^{im\phi_k} e^{i\omega_k t}. \quad (1)$$

¹ <https://github.com/ndotr/Spiking-Neural-Resonator-Network>.

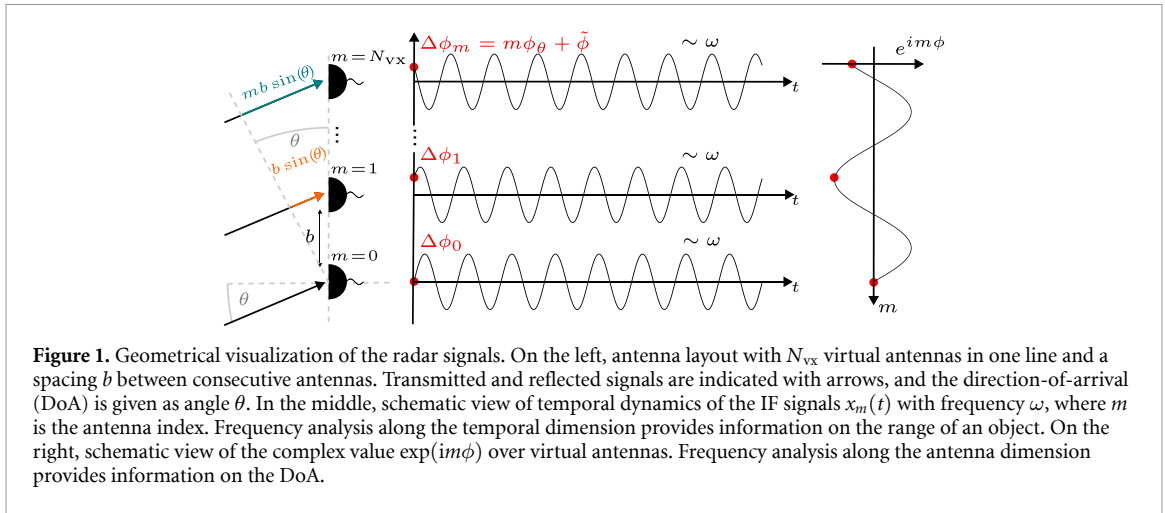


Figure 1. Geometrical visualization of the radar signals. On the left, antenna layout with N_{vx} virtual antennas in one line and a spacing b between consecutive antennas. Transmitted and reflected signals are indicated with arrows, and the direction-of-arrival (DoA) is given as angle θ . In the middle, schematic view of temporal dynamics of the IF signals $x_m(t)$ with frequency ω , where m is the antenna index. Frequency analysis along the temporal dimension provides information on the range of an object. On the right, schematic view of the complex value $\exp(im\phi)$ over virtual antennas. Frequency analysis along the antenna dimension provides information on the DoA.

The frequency ω_k is directly proportional to the radial range r_k between the object k and the sensor. The phase shift $m\phi_k$ depends on the angle between object k and the sensor, and the position of the m th antenna within the antenna layout. The amplitude a_k depends on various factors, such as the radar cross-section (RCS) of the object, the distance between the radar and object d_k , the dynamics of the sensor [25], and properties of the medium. More details on the fundamentals on radar signal processing can be found in [26].

A single spiking neural resonator, defined in the following section, receives and continuously processes the radar signal vector $\vec{x}(t)$ (see figure 1).

3. Neuron model and network architecture

We describe the different processing steps of an extended RF neuron model in the following. First, a complex weight matrix multiplication extract the angle information of the sensor data (section 3.1). Second, the typical resonator dynamics of RF neurons provide information about the distance (section 3.2). Third, novel neuron dynamics estimate the envelope and gradient of the neuron's state dynamics to produce informative spikes (section 3.3). Finally, we incorporate three different spiking functions into our model, that rely on rate-coded and time-coded approaches, respectively (section 3.4). In section 5 and table 5, we evaluate these approaches and compare the results with the FT method.

3.1. Angle estimation—dendritic vector multiplication

The relative phase shifts between the antenna signals of a multi-array antenna system (see figure 1, red) contain information on the direction-of-arrival (DoA) of the objects. Due to a much larger range r_k compared to the distance b between consecutive antennas ($r_k \gg b$), we can assume that the incoming signals are aligned parallel on all antennas. Due to the geometric setup of the antenna array, the range r_k between object k and receiving antenna changes slightly for each antenna (see figure 1). The relative displacements of the range between the 0th and m th antenna $\Delta d_k = mb \sin(\theta_k)$ depends on the angle θ_k between target k and sensor, and on the antenna distance mb . This displacement results in a phase shift between the IF signals of two antennas. By assuming the antenna distance to match multiples of half of the wavelength of the radar sensor $b = \lambda/2$, the relative phase shift between antenna signals $x_0(t)$ and $x_m(t)$ can be stated as $m \cdot \phi(\theta_k) = m \cdot \pi \sin(\theta_k)$.

Frequency spectrum analysis, such as the discrete FT (DFT), estimates the angle information in the signal of a multi-array antenna system. The DFT is a matrix multiplication of a complex matrix W with the IF signal vector $\vec{x}(t)$. The matrix W consists of matrix elements $W_{lm} = e^{-im\phi_l}$, where $l \in [0, N_{vx}]$ indicates phase shifts depending on the resolution of the DFT. Accordingly, we rewrite the DFT as

$$y_l(t) = \sum_{m=0}^{N_{vx}-1} W_{lm} x_m(t) \quad (2)$$

$$= \sum_{n=0}^{N_{vx}-1} e^{-im\phi_k} \sum_k a_k e^{im\phi_k} e^{i\omega_k t} \quad (3)$$

$$= \sum_k^K a_k \underbrace{\sum_m e^{im(\phi_k - \phi_l)} e^{i\omega_k t}}_{\beta_{kl}} \quad (4)$$

$$= \sum_k^K a_k \cdot \beta_{kl} \cdot e^{i\omega_k t}. \quad (5)$$

The case of matching phase shifts for a single target ($l = k$) results in $\beta_{kl} = 1$ and hence in the highest amplification of the signal $y_k = N_{vx} a_k e^{i\omega_k t}$, whereas non-matching phase shifts ($l \neq k$) result in complex factors β_{kl} with $|\beta_{kl}| < 1$. After matrix multiplication, high amplifications indicate an object at a given angle θ_l .

In our proposed network, a single neuron l performs a vector multiplication $y_l(t) = \vec{w}_l \cdot \vec{x}(t)$ of the weight vector $\vec{w}_l = (W_{l0}, \dots, W_{lm})$ and the radar signal vector $\vec{x}(t)$, therefore each neuron corresponds to an angle θ_l . The resulting signal $y_l(t)$ is further processed in the neuron.

3.2. Distance estimation—neural resonators

A spectrum analysis along the temporal dimension extracts the range from a single chirp of the IF signal $x_m(t)$ or the weighted IF signal $y_l(t)$ of an FMCW radar as the frequency is directly proportional to the range of an object to the sensor. We use the RF neuron model [18, 19] as oscillating neuron model,

$$\dot{s}_{jl}(t) = a e^{i\omega_j} s(t) + y_l(t), \quad (6)$$

with j indexing the eigenfrequencies that represent different distances. Typically, we set the initial value of s at $s(0) = 0$ for each chirp. Assuming a general input $y_l(t) = \sum_k^K a_k \beta_{kl} e^{i\omega_k t}$, we can determine the analytic solution to the differential equation (6),

$$s_{jl}(t) = \sum_k^K \frac{ia_k \beta_{kl} e^{i\omega_j t}}{\omega_j - \omega_k} \left(e^{-i(\omega_j - \omega_k)t} - 1 \right). \quad (7)$$

To reduce computational costs, we use the discrete version of the neuron model in the following work and no decay by setting $a_{kl} = 1 \forall k$. Because an analog-to-digital converter typically samples the radar data, the discrete version of the neuron model is sufficient,

$$s_{jl,t+1} = e^{i\Delta\omega_j} \cdot s_{jl,t} + y_{l,t}. \quad (8)$$

The complex state s of a neuron rotates around an angle $\Delta\omega_j$ per sampling time Δt leading to an angular velocity $\omega_j = \frac{\Delta\omega_j}{\Delta t}$. The sampling time Δt and the chirp time T_c define the number of samples N for one chirp. By iteratively applying equation (8) it can be shown, that the neuron state $s(T_c)$ at time T_c with N samples resembles the result of a DFT of the data vector $y_l = (y_0, \dots, y_{N-1})^T$ by setting $\Delta\omega_j = \frac{2\pi j n}{N}$,

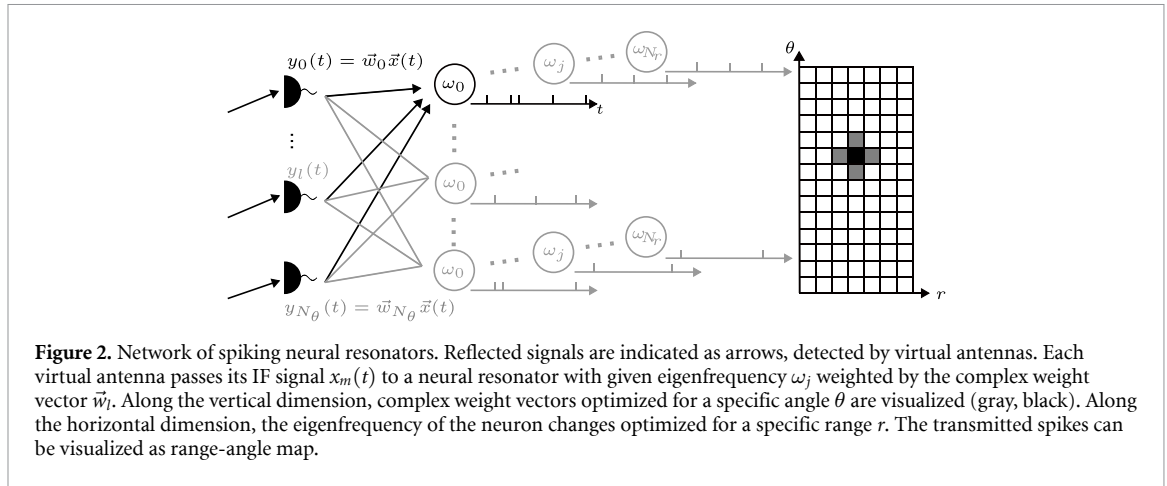
$$\begin{aligned} s_{jl}(T_c) &= e^{iN\Delta\omega_j} y_0 + e^{i(N-1)\Delta\omega_j} y_{l,1} + \dots + e^{i\Delta\omega_j} y_{l,N-1} \\ &= \sum_{n=0}^{N-1} e^{-i2\pi j n / N} y_{l,n}. \end{aligned} \quad (9)$$

Each neuron s_{jl} is indexed by l arising from the DFT along the antenna dimension. The phase shift ϕ_l represents an angle θ_l . Each neuron is also indexed by j depending on the angular velocity ω_j , which represents a range r_j . Hence, each neuron is specific for a DoA angle θ_l and a range r_j . Figure 2 visualizes the layout of the network architecture. As depicted in (9), the state of the neuron at time T_c represents the result of an FT. In the next section, we work out neuron dynamics and spiking functions that emit spike trains before this state is reached while maintaining the accuracy of the FT.

3.3. Envelope estimation and gradient estimation

To estimate the angle and range of an object, we need to detect neurons that match the phase (5) and the frequency (7) of an object. Matching frequencies (resonance) lead to small oscillations $\Delta\omega_{jk} = \omega_j - \omega_k \ll 1$ of the neuron state, and we apply Taylor-approximation on the magnitude of (7) for a single target k around $\Delta\omega_{jk}$,

$$|s_{jl}(t)| \simeq \beta_{kl} t^1 + \mathcal{O}(t^2). \quad (10)$$



A matching phase of an object k leads also to an increased amplitude β_{kl} of the neuron state (7), i.e. $\beta_{kl} \approx 1$. Therefore, the first-order term of the magnitude of the neuron state stores the information of interest. We assume that the remaining parts of the sum (non-resonant) affect only the offset and higher-order oscillations.

Ultimately, two criteria distinguish a neuron that matches the phase and frequency of an object and a non-matching neuron: (a) the magnitude $|s_j(t)|$ follows a linearly increasing function $\beta_{kl} \cdot t$ over time and (b) the gradient of the linear function is large, $\beta_{kl} \gg 1$.

Applying this knowledge, we develop a neuron that estimates an envelope $\Lambda(t)$ of the magnitude of the neuron state by removing uninformative high-order oscillations. We start by estimating the maximum of the neuron's magnitude $s_{\max}(t)$,

$$s_{\max}(t) = \begin{cases} \|s(t)\|, & \text{if } \|s(t)\| > s_{\max}(t) \\ s_{\max}(t), & \text{else.} \end{cases} \quad (11)$$

By estimating only the upper boundary, we neglect any decays or oscillations in the magnitude, which indicate non-resonating behavior. Therefore, we also estimate the maximum width $w_{\max}(t)$ between the neuron's magnitude and the upper boundary to extract the additional information on non-resonance. The maximum of the width $w_{\max}(t)$ is only updated when the width increases,

$$w_{\max}(t) = \begin{cases} s_{\max}(t) - \|s(t)\|, & \text{if } s_{\max} - \|s\| > w_{\max} \\ w_{\max}(t), & \text{else.} \end{cases} \quad (12)$$

The difference $\Lambda(t) = s_{\max}(t) - w_{\max}(t)$ follows the neuron's magnitude as a lower boundary and replicates an envelope estimate removing higher-order oscillations.

We estimate the gradient of $\Lambda(t)$ using exponential filtering,

$$g(t) = (1 - \alpha_g)g(t) + \alpha_g \Delta \Lambda(t) \quad (13)$$

$\Delta \Lambda$ being the update step for the envelope which depends on the updates in $s_{\max}(t)$ and $w_{\max}(t)$, $\Delta \Lambda = \Delta s_{\max} - \Delta w_{\max}$.

Figure 3 shows the estimated gradient $g(t)$, $s_{\max}(t)$, and $\Lambda(t)$. Table 1 summarizes all variables in the neuron model and indicates their dimensions, temporal behavior, and a short description. For the first chirp, we set the initial value to $g(0) = 0$. Otherwise, we use the last gradient estimate $g(T_c)$ as the initial value for the next chirp. Instead of depending on the result of $s(T_c)$, as it replicates the result of an FT, we utilize information about the gradient estimate $g(t)$ to produce informative spikes. This procedure allows us to already transmit information during the sampling process.

3.4. Spiking functions

Neuromorphic applications rely on different spike encoding schemes, such as rate-coding and temporal-coding [27]. In the following section, we describe three approaches that offer individual advantages and trade-offs. The adaptive threshold spiking function builds on rate-coding and benefits from fewer computation steps, but transmits the most spikes. The rate-coded LIF reduces the spike number and allows for early spike transmission, but increases the computational complexity. The temporal-coded LIF reduces

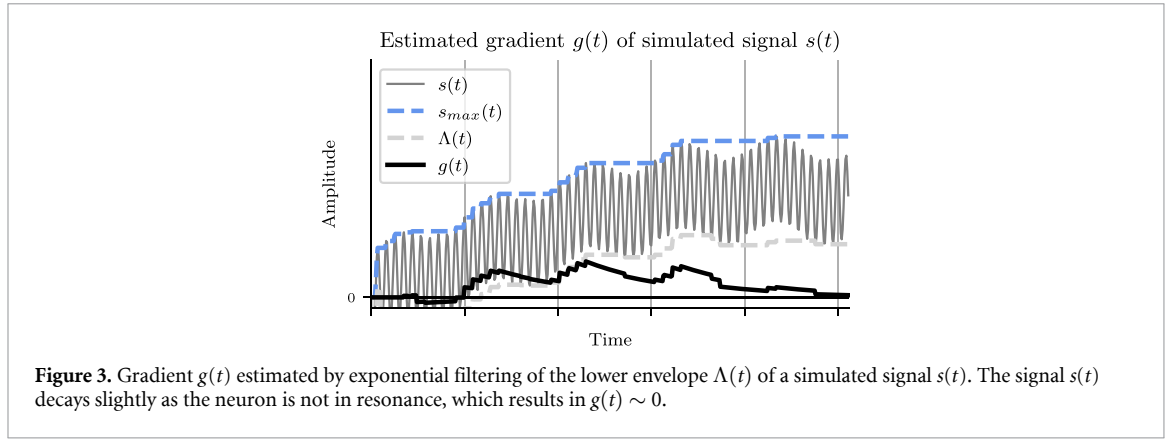


Table 1. Neuron states and parameters for neuron model excluding spiking functions.

Variable	Dim.	Time	Description
$\mathbf{x}(t)$	r	Dyn.	Radar data from N_{vx} antennas
$y(t)$	1	Dyn.	Phase-shifted radar data $y(t) = \vec{w}\vec{x}(t)$
$\mathbf{s}(t)$	2	Dyn.	Neuron state of RF neuron
$\ s(t)\ $	1	Dyn.	Magnitude of neuron state
$s_{\text{max}}(t)$	1	Dyn.	Estimated maximum of magnitude
$w_{\text{max}}(t)$	1	Dyn.	Estimated maximum of width between magnitude and its maximum
$\mathbf{w}$	r	Static	Weight vector of complex weight matrix
ω	1	Static	Eigenfrequency

the spike number even more, but does not benefit from early spike transmission. Figure 4 compares the dynamics and spiking behavior of all spiking functions for a single chirp. Table 2 summarizes the variables and parameters of all the spiking functions.

Adaptive threshold spiking function: The first spiking function relies on an adaptive threshold proposed in [20]. Once the neuron's state reaches a threshold u_{th} , the neuron spikes and increases this threshold by Δu_{th} . Therefore, the total number of spikes is proportional to the maximum value of the neuron within a time window, independent of the negative gradient within this window. Since the magnitude of the neuron state $|s|$ oscillates in case of no resonance $\omega_k \neq \omega_j$, the decrease of the neuron state also contains information. Hence, we propose positive and negative spikes that depend on adaptive thresholds u_{th}^s and u_{th}^w . The neuron follows the spike generation rules,

$$\delta^+(t) = \begin{cases} 1 & s_{\text{max}}(t) > u_{\text{th}}^s(t), \\ 0 & \text{else,} \end{cases} \quad (14)$$

and

$$\delta^-(t) = \begin{cases} -1 & w_{\text{max}}(t) > u_{\text{th}}^w(t), \\ 0 & \text{else,} \end{cases} \quad (15)$$

where s_{max} and w_{max} describe the estimated maxima in section 3.3. The initial thresholds $u_{\text{th}}^{s/w}$ are updated according to

$$u_{\text{th}}^{s/w}(t) = u_{\text{th}}^{s/w} + \Delta u_{\text{th}} \int_0^t \delta^\pm(t') dt' = u_{\text{th}}^\pm + \gamma N_{\text{spikes}}^\pm. \quad (16)$$

For the readout, the number of negative spikes is subtracted from the number of positive spikes $N_{\text{spikes}}^+ - N_{\text{spikes}}^-$. This result gives an estimate of the envelope gradient $\Lambda(t)$ without calculating equation (13) directly. For consecutive chirps, the thresholds are reset to $\Gamma^\pm(0) = 0$.

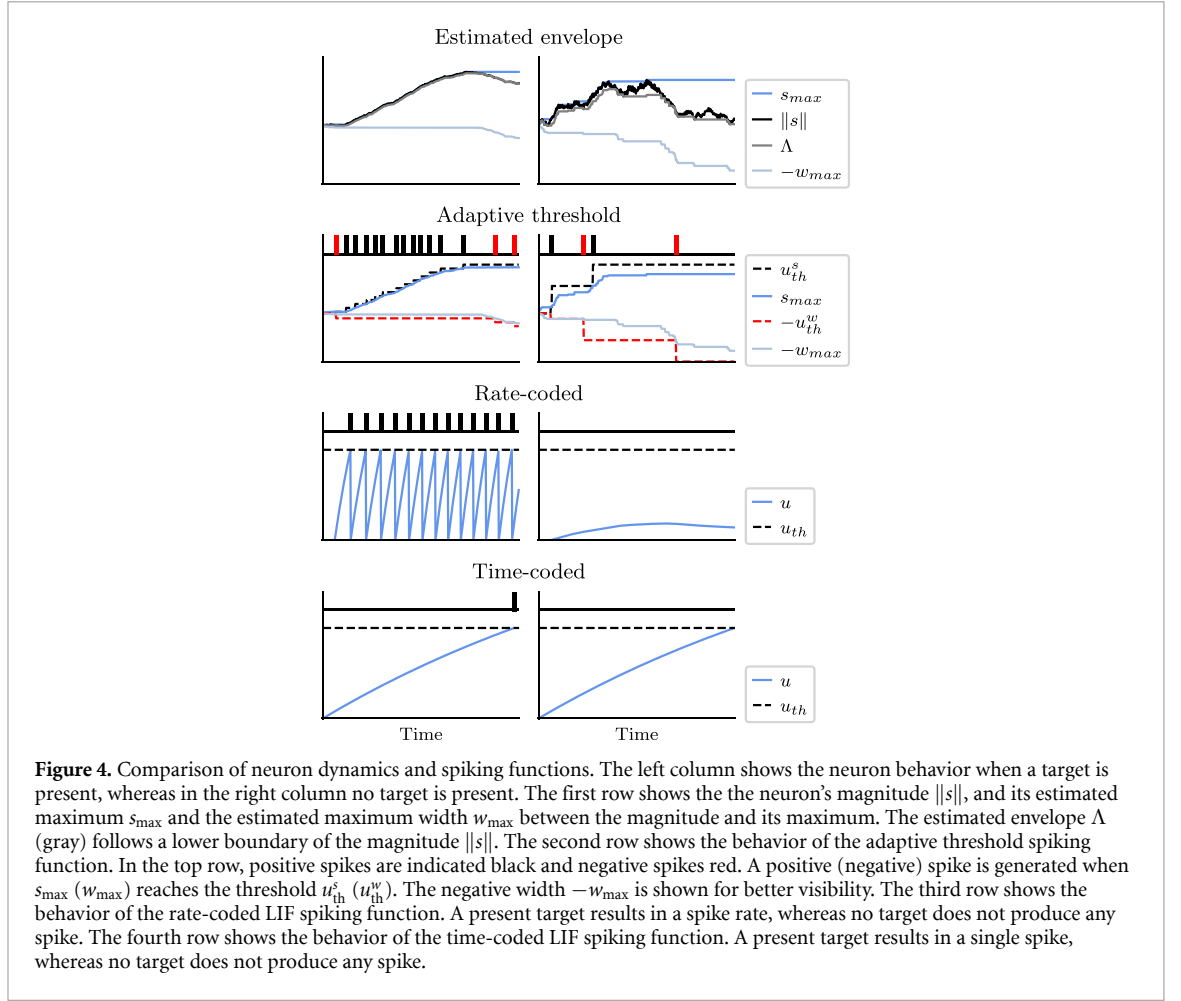


Table 2. Neuron states and parameters for spiking functions.

Adaptive threshold			Rate-coded LIF			Time-coded LIF		
Var.	Dim	Time	Var.	Dim	Time	Var.	Dim	Time
—	—	—	$g(t)$	1	Dyn.	$g(t)$	1	Dyn.
$\Gamma^\pm$	2	Dyn.	$u(t)$	1	Dyn.	$u(t)$	1	Dyn.
—	—	—	α_g	1	Stat.	α_g	1	Stat.
γ	1	Stat.	u_{th}	1	Stat.	u_{th}	1	Stat.
—	—	—	u_{rest}	1	Stat.	u_{rest}	1	Stat.
—	—	—	τ	1	Stat.	τ	1	Stat.

Rate-coded LIF spiking function: Alternatively, we can extend the neuron model using the dynamics of a LIF neuron model,

$$\tau \frac{du}{dt} = -u + g(t) + u_{rest}, \quad (17)$$

to produce spikes with a spike rate $r(t) \sim g(t)$. The LIF neuron spikes once the neuron membrane $u(t)$ reaches the threshold u_{th} and resets its state by subtracting u_{th} . A constant gradient g creates a spike rate

$$r(g) = \frac{1}{-\tau \ln \left(1 - \frac{u_{th}}{g + u_{rest}} \right)}. \quad (18)$$

In the case of no leak, the IF model creates a spike rate that is linearly proportional to g within the time window $[0, T_c]$,

$$r(g) = \frac{g + u_{rest}}{u_{th}} \cdot \frac{T_c}{\tau}. \quad (19)$$

For consecutive chirps, the membrane is reset to $u(0) = 0$.

Time-coded LIF spiking function: Instead of transmitting the information about $g(t)$ via the spike rate, we can use a more efficient encoding. As in [16, 17], a single spike encodes a constant value by charging a LIF model and using an adaptive refractory period. Similar to the rate-coded approach, we calculate the spike time depending on the constant input g as

$$t_s(g) = -\tau \ln \left(1 - \frac{u_{\text{th}}}{g + u_{\text{rest}}} \right). \quad (20)$$

Once the neuron spikes, we can read out the value g by using a linear decoding scheme,

$$g \simeq T_c - t_s(g). \quad (21)$$

Figure 4 illustrates the charging of the neuron and the transmitted spike. For consecutive chirps, the membrane is also reset to $u(0) = 0$. Table 2 summarizes the variables and parameters of the time-coded spiking function and compares them with the alternative spiking functions.

4. Data simulation and metrics

The evaluation of temporal neuron models is time- and energy-consuming, as the processor needs to calculate each time step of each neuron. Therefore, we implemented our neuron model on GPU to process all neurons of the range-angle map in parallel. Each neuron runs independently on a single CUDA core, removing waiting or communication time between different cores. This parallelization drastically improved the computation time by a factor ~ 1000 and allowed us to evaluate our network model on large amounts of data.

4.1. Dataset simulation

For the evaluation, we generated raw radar data from a radar simulator by Infineon Technologies AG. Table 3 summarizes the parameters of the simulated radar sensor. The simulation includes phase noise, thermal noise, and phase-locked loops. The simulator creates targets as single points with a given position, velocity, and RCS. With these parameters, we created a labeled dataset consisting of the raw radar data and a list of point targets, including their attributes. Using two different seeds, we simulated five scenarios, summarized and described in table 4. The scenarios, while being constructed in a simulator, are inspired by automotive scenarios, where usually multiple targets are present. The dataset with seed 0 will be used as a training dataset to optimize the parameters, whereas seed 1 is the evaluation dataset.

4.2. Metric

Since our model resembles the concept of frequency analysis, we compare our results with the FT. To evaluate the performance of our model, we measure the signal-to-noise ratio (SNR) and the accuracy of the target detection by applying the CA-CFAR algorithm [28] to the neurons' output to create classification maps. These maps indicate whether a target is present at a given frequency bin for a distance or an angle. The two approaches, SNN and FT, are compared by their SNR, F -score, precision, and recall values.

The sparsity of the SNN approaches leads to a majority of zeros in the range-angle map. Hence, we slightly adjust the calculation of the SNR by dividing the signal, i.e. the data point d_{target} at the range and angle index of a target, by the sum of data points in the whole map to avoid divisions by zero,

$$\text{SNR}_{\text{target}} = \frac{d_{\text{target}}}{\sum_{ij} d_{ij}}. \quad (22)$$

This adjustment leads to $\text{SNR} = 1$ for no noise. For simplicity, we fix the number of neighboring cells and use no guard cells for all tests. Hence, we parameterize the CA-CFAR algorithm by only two parameters: an offset o that adds to the threshold for the cell under test (CUT) derived from the window and a weight factor α that scales the resulting threshold. The threshold for CUT c_{ij} is given by

$$\Theta_{\text{CUT}}(c_{ij}) = \alpha \cdot \left(\frac{1}{K} \sum_{kl} c_{kl} + o \right), \quad (23)$$

with K as a set of neighboring cells. We fixed $K = 3 \times 5$ for all experiments in the range and angle dimensions, respectively. Typically, the SNN output contains mainly zeros. Therefore, the offset o is needed to generate useful results of the CA-CFAR algorithm.

Table 3. Parameters of the simulated radar sensor.

Parameter name	Parameter value
Initial frequency f_0	76 GHz
Bandwidth B	507.6 MHz
Number of samples N_{samples}	512
Number of chirps N_{chirps}	32
Number of receiving antennas N_{rx}	32
Number of transmitting antennas N_{tx}	1
Duration of chirp t_{chirp}	20.52 μs
Duration of wait time t_{wait}	5.96 μs
Chirp frequency f_{chirp}	37.76 kHz

Table 4. List of simulated data sets.

Dataset name	Description
<i>Close targets 2010</i>	Two targets close by with $\sigma_0 = 10$ and $\sigma_1 = 20$
<i>Close targets 0010</i>	Two targets close by with $\sigma_0 = 10$ and $\sigma_1 = 00$
<i>5 mixed objects</i>	Five targets randomly distributed with $\vec{\sigma} = [0, 5, 10, 15, 20]$
<i>5 persons</i>	Five targets randomly distributed with $\vec{\sigma} = [0, 0, 5, 5, 10]$
<i>8 targets</i>	Up to 8 targets randomly distributed

4.3. Parameter optimization

We distinguish two parts in our neuron model: the gradient estimation and the spiking function. To tune the parameters of the whole neuron model, we optimized the two parts consecutively. First, we optimized the parameters of the gradient estimation and CA-CFAR using a grid search on a training data set by maximizing the F -score. Second, we set the gradient estimation parameters and optimized the spiking functions' parameters with the same approach.

5. Results

To assess the performance of the extended neuron model and the different spiking functions, we measure the F -score, precision, recall and SNR values for two different processing setups. First, we evaluate the processing of a single chirp (section 5.1) and discuss the capabilities of early target detection within a chirp (section 5.2). Second, we evaluate the processing of multiple chirps, analyze the continuous processing capabilities of the network, and compare the results with the average of the multiple chirps (section 5.3).

5.1. Model evaluation for a single chirp

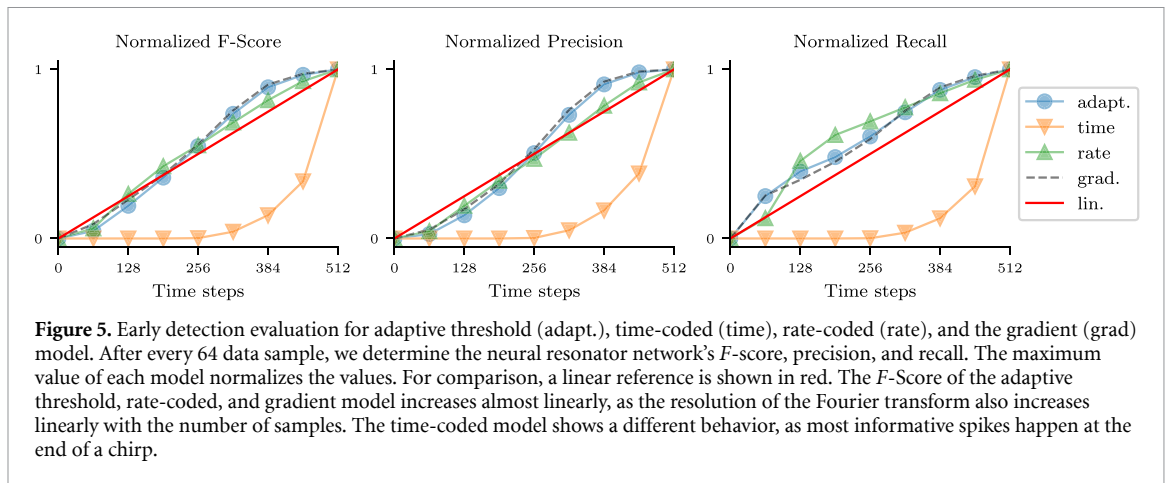
The three spiking functions were evaluated and compared to the FT results. In addition, we analyzed the results of the gradient estimation by directly using the value of the estimated gradient. Table 6 summarizes the optimized parameters of each model, and table 5 shows the F -score, precision, and recall for each spiking model and the comparable FT approach over all datasets. These results show that the rate- and time-coded spiking functions reach higher F -scores than the gradient or adaptive threshold model. Because the results of the gradient model and FT are similar, we can conclude that the advantage of the two spiking models mainly lies in the non-linear mapping of the spiking function itself. The adaptive threshold spiking function, which utilizes a linear mapping from values to spikes, shows only a slight loss in the F -score, affirming the previous assumption. An intrinsic cut-off can explain the increased SNR of all neuron models, as only estimated gradients larger than 0 are transmitted. Even though our model eliminates sensor data storage by continuous processing and transmits only spikes, it performs similarly in classification accuracy as the FT. The intrinsic cut-off and the usage of binary spikes drastically reduce the data bandwidth, as the number of generated spikes shows. The entire network consists of $256 \times 32 = 8192$ neurons. A network utilizing the adaptive threshold spiking function transmits, on average, 27 128 spikes, which is more than the other spiking functions, as it sends positive and negative spikes. The rate-coded spiking function transmits 910 spikes per

Table 5. Evaluation results on simulated data after a single chirp, after averaging eight chirps and after eight consecutive chirps.

Model name	<i>F</i> -score	Prec.	Recall	SNR	Avg. # spikes
Single chirp					
Gradient model	0.61	0.66	0.58	0.012	–
Adaptive threshold	0.60	0.65	0.58	0.010	~27128
Rate-coded LIF	0.61	0.59	0.64	0.104	~910
Time-coded LIF	0.61	0.57	0.66	0.105	~53
FT	0.59	0.64	0.56	0.006	–
Consecutive chirps					
Gradient model	0.72	0.73	0.71	0.013	–
Adaptive threshold	0.60	0.65	0.58	0.010	~2579
Rate-coded LIF	0.69	0.67	0.72	0.117	~126
Time-coded LIF	0.68	0.66	0.71	0.103	~5
Averaged chirps					
Gradient model	0.72	0.71	0.72	0.012	–
Adaptive threshold	0.64	0.75	0.56	0.010	~20632
Rate-coded LIF	0.63	0.59	0.67	0.105	~372
Time-coded LIF	0.63	0.59	0.68	0.105	~43
FT	0.60	0.63	0.59	0.006	–

Table 6. Optimized parameters of the spiking and non-spiking gradient models for the single chirp use-case.

Model name	Optimized parameters
Gradient model	$\alpha_x = 0.6, \alpha_g = 0.001$
Adaptive threshold	$\gamma = 0.1$
Rate-coded LIF	$u_{\text{threshold}} = 0.35, u_{\text{rest}} = 0, \tau = 100$
Time-coded LIF	$u_{\text{threshold}} = 231, u_{\text{rest}} = 250, \tau = 200$



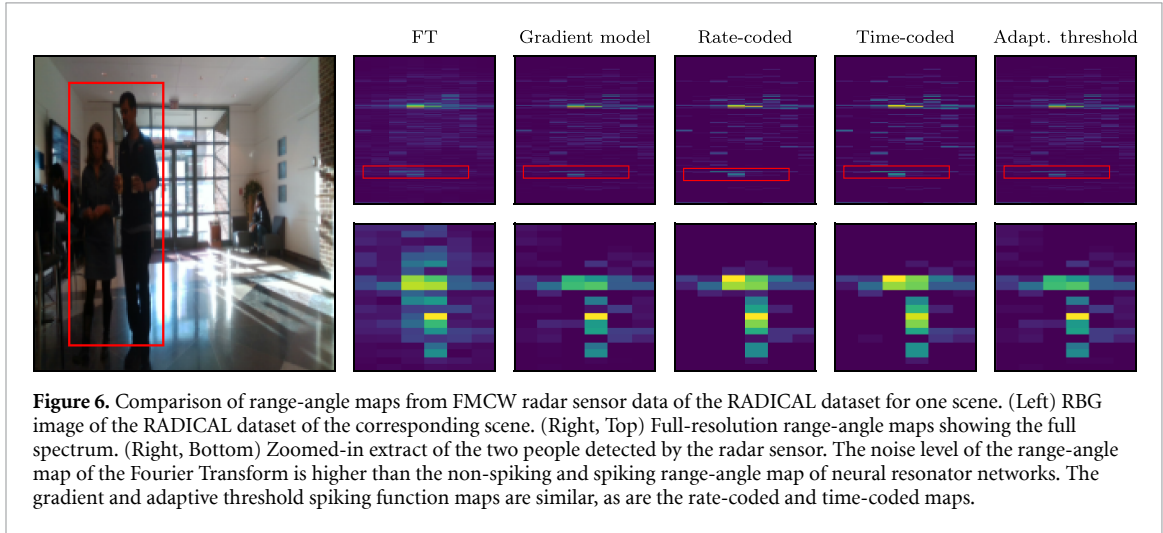
network, leading already to less than one spike per neuron on average. By utilizing the time-coded spiking function, the entire network transmits, on average, only 53 spikes. In contrast, the classic FT approach transmits $256 \times 32 = 8192$ floating values, i.e. 262 144 bits per chirp, assuming float-32. The spiking neural resonator network transmits only 0.02% of the data transmitted by the classic float-32 FT approach.

5.2. Model evaluation of early detections for a single chirp

Because our neuron model continuously processes sensor data and transmits information by sending spikes, objects are already detected during data sampling. In contrast, the FT relies on sampling and storing the data before performing any calculation. Therefore, we also studied the early detection accuracy of our neuron model by evaluating the spike output at every 64th time step to analyze whether early predictions of close-by objects are possible. We performed the evaluation on the *close targets 2010* dataset. The recall results depicted in figure 5 show that for the adaptive threshold and rate-coded spiking functions already after half of the available samples $N_{\text{samples}}/2$ more than 75% of the detectable objects are recognized. The time-coded spiking function, however, needs due to its internal dynamics more time to produce a spike. Since most of the peaks in the frequency spectrum are close to zero, the spikes typically happen at the end of a chirp.

Table 7. Optimized parameters of the spiking and non-spiking gradient models for the consecutive chirps use-case.

Model name	Optimized parameters
Gradient model	$\alpha_x = 0.6, \alpha_g = 0.001$
Adaptive threshold	$\gamma = 0.1$
Rate-coded LIF	$u_{\text{threshold}} = 1.5, u_{\text{rest}} = 0, \tau = 100$
Time-coded LIF	$u_{\text{threshold}} = 232, u_{\text{rest}} = 250, \tau = 200$



5.3. Model evaluation for multiple consecutive chirps

Due to a high chirp rate ($\sim$ kHz) of FMCW radar sensor [26] changes in the scenes between consecutive chirps are hardly detected in range-angle maps. Consequently, it is natural to exploit the continuous computation capabilities of the neurons to process consecutive chirps without resetting the entire network. Therefore, we also evaluate the classification accuracy after processing multiple chirps of the same scene. We follow the optimization strategy described before and summarize the new parameters for processing 8 chirps in table 7. It is important to emphasize that the optimal parameters depend not directly on the number of processed chirps but on the state of convergence to the gradient estimation. After a specific number of chirps $\tilde{N}_{\text{chirps}}$, the estimated gradient converges. By optimizing the parameters with the assumption of a converged gradient estimation, results are optimal for all chirps $n_{\text{chirps}} > \tilde{N}_{\text{chirps}}$. Earlier processed chirps will result in reduced accuracy.

The results summarized in table 5 show an increased performance compared to the single chirp evaluation. These results are also compared to an evaluation of averaging the output over the same 8 consecutive chirps. In this case, the same parameters of the single chirp evaluation are taken (table 6). Table 5 summarizes the results for F -Scores, precision, recall, SNR, and spike numbers, including the results of averaging the FT calculations. To avoid an over-saturation of spikes for continuous chirp processing, the optimization strategy increased the spiking threshold, which leads to a reduced average spike number compared to the single or average processing presented in table 5. The detection accuracy of the neuron models is comparable and situated between the continuous processing and averaging approach, leading to the conclusion that storing outputs of multiple chirps is unnecessary. In general, the FT approach performs slightly worse than all neuron models.

5.4. Visual results on real data

To show the applicability of our model on real-world radar sensor data, we applied it to publicly available raw radar data. The availability of raw radar data including a multi-antenna layout is limited. The RADICAL dataset consists of a sample data file of indoor recordings of walking persons. Figure 6 shows a scene containing 4 persons, two walking close to the sensor. We compare the range-angle maps of the FT calculation, the gradient model, and the three different spiking functions. The comparison shows a similar frequency spectrum for all approaches (upper row in figure 6). However, our models show a significantly reduced noise level, which becomes evident in the zoomed-in extracts of the range-angle maps (lower row in figure 6).

6. Conclusion

This work extends the RF model to optimize it for radar processing applications. We focused on reducing latency and data bandwidth by allowing spikes during the sampling of radar data, eliminating the need to store any sensor data. Our continuous approach, therefore, stands out in contrast to the classic Fourier transformation, where all data points are sampled, stored, and processed afterward. In general, utilizing only a subset of all samples within a chirp reduces the resolution of the frequency analysis. However, the neuron model spikes during sampling time; we have shown that the object detection accuracy is comparable to or even better than the FT approach. This is achieved by first estimating the first-order part (linear in time) of the resonator dynamics, which filters out noise from non-resonating objects, and second, by emitting spikes proportional to the gradient of the linear part, which is comparable to a threshold function since only positive gradient above a specific value can produce spikes. The three different spiking functions compared in this work show similar target detection accuracy but have several trade-offs one must consider. The adaptive threshold spiking function is the most straightforward of these three but transmits the most spikes, including negative ones. The rate-coded spiking function has a similar complexity as the time-coded function but uses more spikes to convey information and is therefore suitable for early detection. The time-coded spiking function utilizes the least number of spikes but lacks the feature of early detection. To conclude, we have demonstrated that a network based on our spiking neural resonators can achieve the same performance as the FT while reducing the latency and data bandwidth. The data bandwidth can be reduced to 0.02% of the data bandwidth for a float-32 FT. This work pioneers using resonating neuron dynamics for radar data processing showing the potential of neuromorphic algorithms in the domain of autonomous driving. We reliably produce spikes during data recording that match the accuracy of state-of-the-art methods while drastically reducing the amount of transmitted data. This approach enables two essential benefits for autonomous driving. First, it establishes the first processing unit of a neuromorphic radar processing pipeline by converting float-point or analog radar data into sparse spikes. The subsequent algorithms can rely on the accuracy of this conversion. Second, simultaneous spiking and data recording enable early detection in critical situations, such as early pedestrian detection for collision avoidance.

7. Future work

The current state of the network does not implement any lateral connections; therefore, each neuron processes information independently, allowing complete parallel computations and optimizations. Introducing laterally inhibiting connections between neurons can theoretically improve the detection accuracy by suppressing neighboring neurons excited by the same target.

So far, we evaluated a static data set, i.e. the targets were not moving, and we did not consider temporal changes. The continuous setup of the neuron model does not reset the internal variable of the gradient estimation and uses the information of previous chirps. Therefore, we plan to evaluate the performance and the reaction time of the network in a highly dynamic and realistic environment.

The neuron model is continuous in time and can theoretically process analog data directly from the radar sensor. Analog processors that implement this model can be highly efficient in energy consumption and should be explored in the future. First implementations of analog RF models have been studied in [29]. Directly processing analog signals and transmitting only sparse spikes for further post-processing paves the way for efficient neuromorphic radar sensors.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://github.com/ndotr/Spiking-Neural-Resonator-Network>.

Acknowledgments

This research has been funded by the Federal Ministry of Education and Research of Germany in the framework of the KI-ASIC Project (16ES0995).

ORCID iDs

Nico Reeb  <https://orcid.org/0000-0003-4617-6215>

Javier Lopez-Randulfe  <https://orcid.org/0000-0002-6922-6951>

Robin Dietrich  <https://orcid.org/0000-0001-5820-1399>

References

- [1] Leiserson C E, Thompson N C, Emer J S, Kuszmaul B C, Lampson B W, Sanchez D and Schardl T B 2020 There's plenty of room at the Top: what will drive computer performance after Moore's law? *Science* **368** eaam9744
- [2] Lin S-C, Zhang Y, Hsu C-H, Skach M, Haque Md E, Tang L and Mars J 2018 The architectural implications of autonomous driving: constraints and acceleration *Proc. 23rd Int. Conf. on Architectural Support for Programming Languages and Operating Systems* pp 751–66
- [3] Vogginger B et al 2022 Automotive radar processing with spiking neural networks: concepts and challenges *Front. Neurosci.* **16** 851774
- [4] Patole S M, Torlak M, Wang D and Ali M 2017 Automotive radars: a review of signal processing techniques *IEEE Signal Process. Mag.* **34** 22–35
- [5] Sokoloff L 1996 Cerebral metabolism and visualization of cerebral activity *Comprehensive Human Physiology* (Springer) pp 579–602
- [6] Mead C 2020 How we created neuromorphic engineering *Nat. Electron.* **3** 434–5
- [7] Marković D, Mizrahi A, Querlioz D and Grollier J 2020 Physics for neuromorphic computing *Nat. Rev. Phys.* **2** 499–510
- [8] Maass W 1997 Networks of spiking neurons: the third generation of neural network models *Neural Netw.* **10** 1659–71
- [9] Rhodes O, Peres L, Rowley A G D, Gait A, Plana L A, Brenninkmeijer C and Furber S B 2020 Real-time cortical simulation on neuromorphic hardware *Phil. Trans. R. Soc. A* **378** 20190160
- [10] Rueckauer B, Lungu I -A, Hu Y, Pfeiffer M and Liu S -C 2017 Conversion of continuous-valued deep networks to efficient event-driven networks for image classification *Front. Neurosci.* **11** 294078
- [11] Bing Z, Meschede C, Röhrbein F, Huang K and Knoll A C 2018 A survey of robotics control based on learning-inspired spiking neural networks *Front. Neurobot.* **12** 35
- [12] Davies M, Srinivasa N, Lin T -H, China Y, Cao Y, Choday S H, Dimou G, Joshi P, Imam N and Jain S 2018 Loihi: a neuromorphic manycore processor with on-chip learning *IEEE Micro* **38** 82–99
- [13] Rhodes O, Bogdan P A, Brenninkmeijer C, Davidson S, Fellows D, Gait A, Lester D R, Mikaitis M, Plana L A and Rowley A G D 2018 sPyNNaker: a software package for running PyNN simulations on SpiNNaker *Front. Neurosci.* **12** 816
- [14] Kheradpisheh S R, Ganjtabesh M, Thorpe S J and Masquelier T 2018 STDP-based spiking deep convolutional neural networks for object recognition *Neural Netw.* **99** 56–67
- [15] Jiménez-Fernández A, Cerezuela-Escudero E, Miró-Amarante L, Domínguez-Morales M J, Asís Gómez-Rodríguez F, Linares-Barranco A and Jiménez-Moreno G 2016 A binaural neuromorphic auditory sensor for FPGA: a spike signal processing approach *IEEE Trans. Neural Networks Learn. Syst.* **28** 804–18
- [16] López-Randulfe J, Reeb N, Karimi N, Liu C, Gonzalez H A, Dietrich R, Vogginger B, Mayr C and Knoll A 2022 Time-coded spiking Fourier transform in neuromorphic hardware *IEEE Trans. Comput.* **71** 2792–802
- [17] Lopez-Randulfe J, Reeb N and Knoll A 2023 Integrate-and-fire circuit for converting analog signals to spikes using phase encoding *Neuromorph. Comput. Eng.* **3** 044002
- [18] Izhikevich E 2001 Resonate-and-fire neurons *Neural Netw.* **14** 883
- [19] Orchard G, Frady E P, Rubin D B D, Sanborn S, Shrestha S B, Sommer F T and Davies M 2021 Efficient neuromorphic signal processing with Loihi 2 *2021 IEEE Workshop on Signal Processing Systems (SiPS)* pp 254–9
- [20] Auge D and Mueller E 2020 Resonate-and-fire neurons as frequency selective input encoders for spiking neural networks *Technical Report* (Technical University Munich)
- [21] Hille J, Auge D, Grassmann C and Knoll A 2022 Resonate-and-fire neurons for Radar interference detection *Proc. Int. Conf. on Neuromorphic Systems 2022 (ICONS '22)* (Association for Computing Machinery)
- [22] Hamidi S, Nezhad-Ahmadi M-R and Safavi-Naeini S 2018 TDM based Virtual FMCW MIMO Radar Imaging at 79GHz *2018 18th Int. Symp. on Antenna Technology and Applied Electromagnetics (ANTEM)* pp 1–2
- [23] Hamidi S and Naeini S S 2021 CDM based virtual FMCW MIMO Radar imaging at 79 GHz *2021 IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE)* pp 1–4
- [24] Zwanetski A, Kronauge M and Rohling H 2013 Waveform design for FMCW MIMO radar based on frequency division *2013 14th Int. Radar Symp. (IRS)* vol 1 pp 89–94
- [25] Hau F, Baumgärtner F and Vossiek M 2020 The degradation of automotive radar sensor signals caused by vehicle vibrations and other nonlinear movements *Sensors* **20** 6195
- [26] Jankiraman M 2018 *Fmcw Radar Design* (Artech House) p 430
- [27] Auge D, Hille J, Mueller E and Knoll A 2021 A survey of encoding techniques for signal processing in spiking neural networks *Neural Process. Lett.* **53** 4693–710
- [28] Barkat M and Varshney P K 1987 On adaptive cell-averaging CFAR (Constant False-Alarm Rate) radar signal detection *Final Technical Report* (Syracuse Univ. NY. Dept. of Electrical and Computer Engineering)
- [29] Lehmann H, Hille J, Grassmann C and Issakov V 2023 Direct signal encoding with analog resonate-and-fire neurons *IEEE Access* **11** 1