Pose Estimation and Map Formation with Spiking Neural Networks: towards Neuromorphic SLAM

Raphaela Kreiser[1], Panin Pienroj[2], Alpha Renner[1], and Yulia Sandamirskaya[1]

Abstract— In this paper, we investigate the use of ultra low-power, mixed signal analog/digital neuromorphic hardware for implementation of biologically inspired neuronal path integration and map formation for a mobile robot. We perform spiking network simulations of the developed architecture, interfaced to a simulated robotic vehicle. We then port the neuronal map formation architecture on two connected neuromorphic devices, one of which features on-board plasticity, and demonstrate the feasibility of a neuromorphic realization of simultaneous localization and mapping (SLAM).

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is one of the basic tasks in mobile robotics and amounts to building a map of an unknown environment while at the same time keeping track of the agent's location. When the robot estimates its location based on its own sensory information (e.g., visual or odometry), errors accumulate, leading to an increasing uncertainty. Probabilistic models and graph-based methods are typically used to integrate sensory inputs and update the map of the environment, delivering successful solutions to the SLAM problem [1]. However, when computational and power resources are limited, as in mobile applications, aerial vehicles, or robotic insects, more efficient solutions are needed to enable real-time processing and long operating time for embedded SLAM systems [2], [3], [4].

In contrast to the high computational demand of robotic SLAM, even animals with rather small brains, such as bees or ants, are amazingly good at keeping track of their position and storing a map of an environment [5]. Seeking inspiration in biology for the development of SLAM systems is therefore a promising path and many neurally-inspired SLAM systems have been developed [6], [7], [8], [9], [10], [11], [12], [13]. For instance, inspired by the navigation system of rats, a competitive biologically-inspired neural SLAM system – RatSLAM – was proposed [11], [6]. Using the RatSLAM algorithm, a mobile robotic platform can estimate its pose using a 3D attractor network (for position and orientation) and simultaneously use landmark information to build a map in indoor and outdoor environments.

A drawback of neuronal SLAM is that simulations of neuronal systems on conventional computers waste a lot of computing resources on traversing the "von Neumann memory bottleneck" of this sequentially processing hardware. This renders real-time low-power processing of neuronal algorithms practically impossible. However, neuromorphic hardware offers a computing substrate that is inspired by the structure and dynamics of biological neural systems, which allows us to fully exploit advantages of neuronal SLAM. First approaches towards neuromorphic SLAM have been attempted recently using neuromorphic computing hardware or sensors [14], [15]. Neuromorphic hardware is massively parallel and event-based, and, in case of mixed-signal analog/digital devices used in this work, particularly energy efficient and compact [16], [17], [18]. The latter systems, however, suffer from device mismatch, leading to noisy, unreliable, and hard to control computing units [19]. By drawing inspiration from biology, robust and reliable spiking neural network architectures can be realized on mixed signal hardware [20], [21]. Such architectures could lead to a breakthrough in energy-efficient and fast real-world SLAM.

Recently, we have reported first steps towards fully neuromorphic SLAM in mixed signal analog/digital neuromorphic hardware. In particular, we have introduced a 1D path integration architecture, realized with a spiking neural network on the neuromorphic device Reconfigurable On-Line Learning Spiking (ROLLS), interfaced to a robotic vehicle [22]. Here, we extend this work to a fully-fledged path integration in a 2D environment and realize a collision map using plastic synapses of the spiking neural network on the neuromophric chip. We first validate the simulated spiking neural architecture with a robotic simulator, then we present a proof of concept realization of the 2D path integration in a prototype neuromorphic device. We show how this system can be extended to learn a map using a combination of two neuromorphic devices, one of which features an on-chip plasticity (learning) mechanism [23], [17]. In simulation, we show how the learning rule, realized on the ROLLS chip, can be used both to initially learn and update (correct) the map.

II. METHODS

A. Biological neural networks and neuromorphic hardware

In the hippocampus and entorhinal cortex of rats, different types of spatially-tuned neurons were found: the head-direction cells are sensitive to the heading direction of the animal, place cells are active each time the animal visits a particular part of the environment, and grid cells presumably perform path integration [25]. In this work, we draw inspiration from the rat's navigation system and build

(A) Principle of working of silicon neurons **(B) Scheme of the DYNAPSE chip** **(C) Scheme of the ROLLS chip**
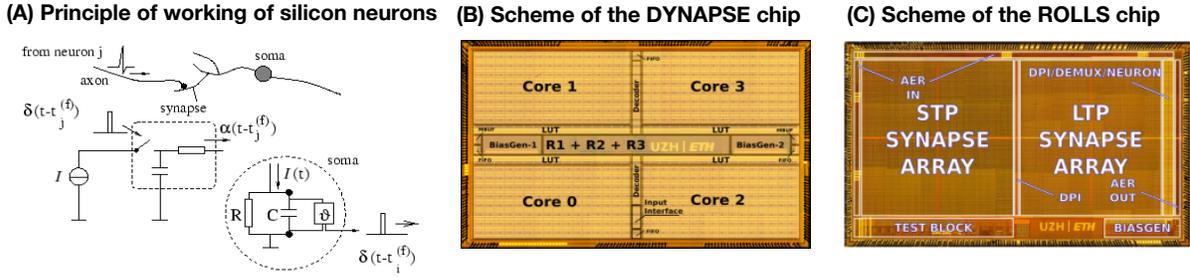

Fig. 1: Neuromorphic hardware used in this project. (A) A basic principle of realization of a simple model of biological neuron and synapse using electrical circuits, from [24]. (B), (C) layout of neuromorphic chips, used in this work.

a spiking version of RatSLAM [11] that is suited for a neuromorphic realization.

Mixed-signal neuromorphic hardware emulates the dynamics of biological neurons and synapses using properties of transistors in subthreshold regime, leading to ultra low power consumption (on the order of several mW when 4K neurons fire at a rate of 100Hz) and compact device size ($< 40mm^2$) [16], [17], [18]. This inherently parallel and event-based hardware offers an efficient computing substrate for architectures of spiking neural networks.

Fig. 1 shows the basic working principle of an artificial silicon neuron that emulates a leaky integrate-and-fire model[1] (A) and layouts of the two neuromorphic hardware platforms, for which we develop the SLAM system. The Dynamic Neuromorphic Asynchronous Processor (DYNAP) (B) comprises 4K neurons, divided into four chips with four cores each [27]. This board enables efficient communication between neurons and other devices using the Address-Event Communication Protocol (AER) and 64 analog synapses per neuron. The synapses are non-plastic however (no on-chip learning). Fig. 1(C) shows the ROLLS, a smaller chip with 256 neurons that features on-chip learning using dedicated plasticity circuits [17].

B. Spiking neural network simulations

We use the spiking network simulator Brian2 [28] to test feasibility of the developed neuronal architectures for path integration and map formation before porting them on the prototype neuromorphic devices. Brian2 allows us to simulate the neuronal dynamics and the plasticity rule that are emulated on the neuromorphic hardware.

To provide the neural simulation with realistic input, we interfaced Brian2 to the robot simulator V-REP [29]. In our experiments with neuromorphic hardware, a physical robotic vehicle was used[2].

III. THE MODEL

Fig. 2 shows an overview of the spiking neural network (SNN) architecture for localization and map formation. The proposed neural architecture models neural networks that

were discovered in the cortical-hippocampal parts of the rodents' brain. The SNN consists of 3 main networks: the collision detection (CD) network, the heading direction (HD) network, and the position network (PN). The PN is connected to the CD network via plastic synapses to enable learning of the locations where obstacles have been encountered.

The HD network uses the wheels' encoder output to perform 1D angular path integration to obtain the robot's heading direction. The estimated heading direction of the robot is passed to the PN, which performs 2D path integration of motor commands to obtain the robot's position. The CD network receives input from the robot's collision sensors (IR sensors in V-REP and a bumper sensor on the physical robot) and forms an activity "bump" representing the position of the collision relative to the robot's heading direction, if a collision is detected. Finally, an association between the position of the robot and a collision event is learned (or unlearned) by plastic synapses that realize a spike-timing dependent plasticity rule [30]. We have demonstrated previously that such a synaptic "memory" representation is capable of solving the SLAM problem in a "haptic-SLAM" scenario [31], [32].

A. Collision detection network

The CD network shown in Fig. 3 represents the location of collision events around the robot relative to its current heading direction. Each neuron in this neural population
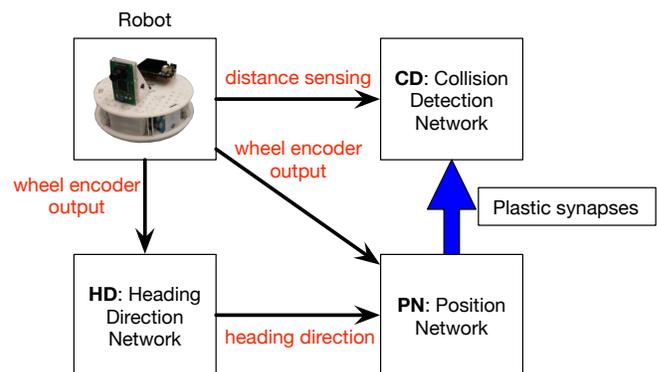

Fig. 2: Overview of the path integration and map formation SNN.

---

[1] A more sophisticated circuitry is needed to make this basic principle work in a reliable, configurable, and scalable manner [26], [17].

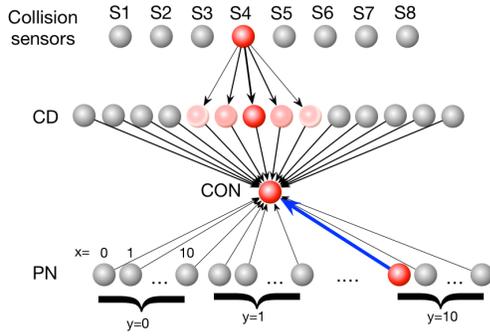[2] Custom-made platform "Omnibot" provided by Jörg Conradt, TUM

Fig. 3: Collision detection network: CD neurons are driven by 8 collision-sensors and represent the position of the collision around the robot in an activity "bump". The "collision-or-not" (CON) neuron signals a collision independent of its position around the robot. Plastic synapses between the position network (PN) and the CON neuron represent the collision map (one of the plastic synapses that is currently updated is shown in blue).

Fig. 4: Heading direction network (only TR and SR neurons are shown. Line with circle show inhibitory and arrow excitatory connections.

represents a range of angles around the robot. The CD neurons are connected in a winner-take-all (WTA) network, which helps to filter out sensor noise [33], [21]. Thus, each neuron in the CD population inhibits all other CD neurons, apart from its two neighbors. In simulation, CD neurons receive input from 8 Poisson spike generators that fire whenever the 8 IR sensors of the robot detect presence of an object in a specific direction. In the neuromorphic hardware implementation, the CD neurons are stimulated directly by the output of the omnibot's bumper sensors.

B. Heading direction network

The neuromorphic HD network was first developed by us in [22]. This work demonstrated path integration of a constant angular velocity of the robot on the ROLLS chip. Here, we generalize this HD architecture by adding an array of "turning speed" neurons that indicate different angular velocities. Fig. 4 shows part of the HD network. The network consists of HD neurons, which represent the angle of the robot's heading relative to a fixed external reference frame; integrating heading direction neurons (IHD) that update the heading representation in HD as the robot moves; shift left (SL) and shift right (SR) layers that realise different shifts between HD and IHD, and turn left (TL) and turn right (TR) motor neurons that represent different turning speed of the robot. A reset population can correct the activity in the HD based on another sensory modality (e.g., vision or IMU [22]). The HD and IHD neurons form WTA networks. The number of layers in the SL/SR populations corresponds to the number of neurons in TL/TR.

In HD network in Fig. 4, each neuron of the HD layer has inhibitory connections to all neurons of every Shift layer, except for the neurons with the same index. We call this connectivity pattern the *inhibitory masking*, which provides a robust mechanism for activation of a neuron on crossing of two inputs on the noisy neuromorphic hardware. The
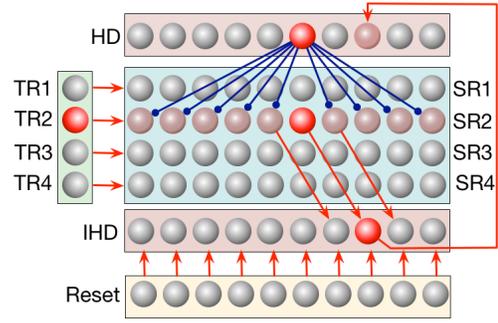
"turning" neurons (TL, TR) excite a corresponding shift layer with a one-to-all connectivity pattern. Due to the inhibition from the HD layer, only a single shift neuron in SL/SR becomes active for each combination of an active HD and TL/TR neuron.

The Shift layers connect to IHD neurons such that the post-synaptic neuron's index is shifted (by one or more, depending on the current angular velocity and the, respectively, the active shifting layer). The IHD neurons, in their turn, strongly excite the HD neurons in a one-to-one manner. The activity in the IHD layer, shifted relative to the previous activity in the HD layer, leads to an update of the estimated heading direction.

As the HD network integrates motor commands (the TR/TL activity), it accumulates errors. A reset population periodically corrects the activity in the IHD population. In our experiments, we periodically acquire the robot's orientation from its compass and use it to reset activity in the IHD network. We have previously shown how visual reset can be used in a similar manner [22].

C. Position network

The position network uses the heading direction output of the HD to perform path integration and to estimate the robot's position. Fig. 5 shows the overview of the position network, which has a similar structure as the HD network.

The PN forms a 2D array of neurons that represents the Cartesian coordinates on the map. In addition, the network comprises a number of shifting layers, one for each estimated heading direction (eight directions were used here). Each layer shifts the neural activity to the corresponding direction. The shifting layers drive activity in the integrating position neurons (IPN), which, in their turn, shift the activity peak in the PN population. For example, the North shifting layer connects to IPN such that the index of each neuron is increased in the y direction, whereas, e.g., the South-West shifting layer connects to IPN such that the neuron index is decreased both in x and y direction. The magnitude of the shift depends on the robot's speed and is kept constant in our experiments. A layer of reset units can reset the position estimate based on an external signal.
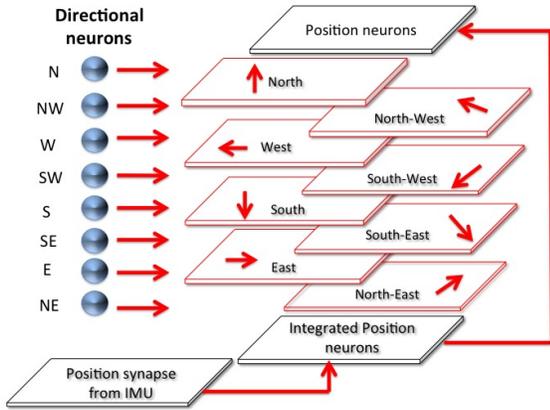
Fig. 5: Overview of the position network.

D. Plastic connections: learning a map

Our goal here is to learn a map of walls in an arena using plastic synapses between the CD and the PN populations. These plastic synapses store an association between the representation of a collision event and the robot's position. We use a spike-timing dependent learning rule that updates the synaptic weight at each pre-synaptic spike depending on the state of the post-synaptic neuron [30], which is realized in plastic synapses on our neuromorphic chip [17].

Every neuron in the CD network connects to a Collision-or-not (CON) neuron with all-to-one excitatory synapses (Fig. 3). The CON is active whenever a collision happens[3]. All neurons in the PN network connect to the CON with plastic synapses, in order to learn the correlation between the firing of CON and position neurons.

According to the learning rule [30], simultaneous firing of two neurons leads to potentiation (strengthening) of the connecting synapse. On the other hand, when the pre-synaptic neuron (PN) fires without the post-synaptic neuron firing (CON), the synapse is depressed (weakened). Weight depression enables the correction of falsely learned plastic synapses resulting from error in heading direction or position networks. In addition, this unlearning mechanism allows updating of the map if the environment changes.

E. Neuromorphic hardware realization of the architecture

Fig. 6 shows an overview of the hardware realization of the 2D path integration architecture (only most important populations are shown here). In this proof of concept, we used the DYNAP device to implement the heading direction, position, and turning networks, whereas we used the ROLLS device that features on-chip plasticity to realize map learning. We have used sensory inputs recorded from the robot, driving in an environment and colliding with walls, to stimulate neuron populations on the neuromorphic devices: the TL and TR populations on the DYNAP were stimulated based on the robot's turning speed, and the CD population on the

[3]Thus, we disregard the egocentric location of the collision here, which could be used to obtain a more precise collision map
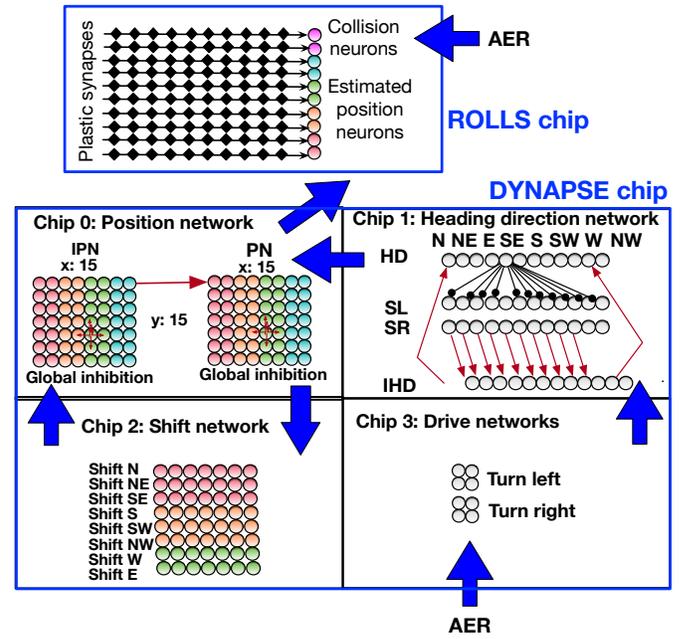

Fig. 6: Spiking neural network architecture for 2D path integration and map formation, realized in the neuromorphic devices ROLLS and DYNAP.

ROLLS was stimulated when the robot's bumpers signaled a collision.

The DYNAP processor is limited in the number of neurons (4096), and each neuron can maximally receive 64 synaptic inputs. The 32x32 map that was learned in simulation required 1024 position neurons and a directional shifting layer with a total of 8192 neurons. Using a 16x16 map for pose estimation, the shifting layer alone would already require 2048 neurons. Thus, we used a modified shifting mechanism in hardware by reducing the 2D shifting layers to one-dimensional vectors. Each neuron in the vector excites a slice of neurons in the IPN network towards the direction that is represented by the activity peak in the HD network.

IV. RESULTS

A. Precision of the neural path integration

In the heading direction network, we simulated 72 spiking neurons in the HD, IHD, and Reset populations (leading to a resolution of $5°$), and 6 angular velocity neurons in the TR and TL populations. Thus, there were 6 layers of 72 neurons in SL and in SR populations. In the 2D path integration network, a 32×32 populations of neurons were used to represent the x and y axes of the map, leading to a total of 1024 neurons in the PN, IPN, and shift populations. Therefore, a total of $1024 \times 8 = 8192$ neurons were used in the position shifting layers. Resolution of the PN networks in the hardware realization was 16×16.

Fig. 7 shows the outcome of a 270 seconds simulation, comparing ground-truth for position estimation (converted to neuron units, red curve) and activity of spiking neurons in the PN neural population. Fig. 7(a) shows that without
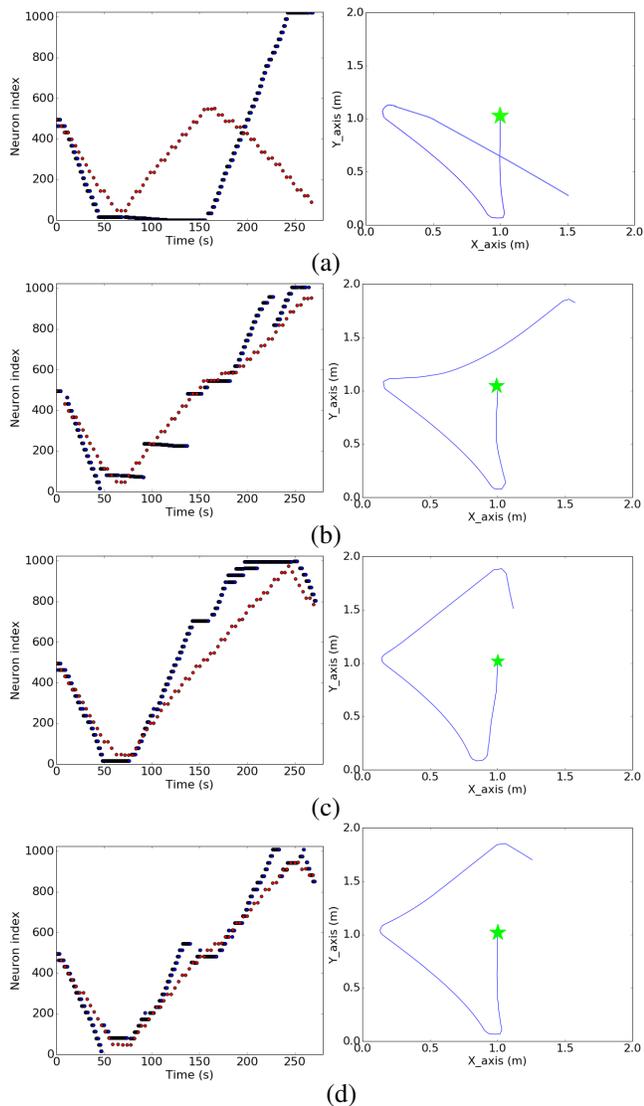
Fig. 7: The comparison of the robot's true position in neuron units (red dots) and the position read from the PN network (blue dots): (a) without reset in both HD and PN networks; (b) without reset in HD network but reset every 47 seconds in PN network; (c) with reset every 28 seconds in HD network but without reset in PN network; and (d) with reset every 28 seconds in HD network and every 47 seconds in PN network. Plots on the right show the trajectory of the robot from the top view; green star indicates the starting point.

the reset in either heading direction or position networks, the error that originates from the error in the HD neural population accumulates and activity in the position network quickly diverges from the ground truth. In Fig. 7(b), the position network is reset every 47 seconds and in Fig. 7(c), only the heading direction network is reset every 28 seconds. Here, the remaining error results from quantization in the position network.

Finally, in Fig. 7(d), where the HD network and the PN were reset every 28 and 47 seconds respectively, the estimation becomes more accurate. The relatively coarse position
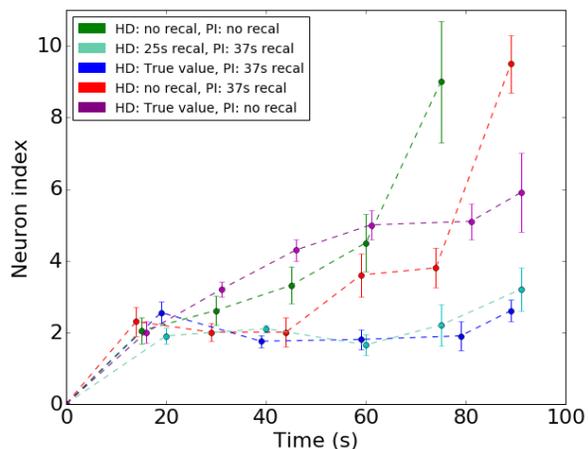

Fig. 8: Mean square error of 10 simulations; error bars indicate one standard deviation. Five different conditions on error evaluation are shown.

network is able to track the position of the robot. However, reset in both heading direction and position networks is essential to keep the accumulated error within bounds.

The errors of the robot localization under 5 reset conditions are shown in Fig. 8. These conditions were derived from the permutation of the reset conditions in heading direction and position networks, which were "no reset" or reset every 25 or 37 seconds, using the value of the robot's heading direction from its compass. The mean errors are computed from 10 simulations in the environment shown in Fig. 9(a) and the error bar is one standard deviation of each data point.

Inaccurate estimation of the heading direction in the HD population leads to fast error accumulation in the position network (see the steep slope of the red and green curves in Fig. 8). When the true heading value is used, the error increases constantly with time (purple line), indicating that also the position network contributes to error accumulation. Thus, the periodic reset using vision or IMU is required for accurate position estimation in our neuromorphic realization.

B. Learning a simple collision map

The map of the environment is stored in the plastic synaptic weights between the PN and CD populations. Fig. 9 shows maps in 3 different environments, encoded ("memorized") in the plastic synapses in simulation. In each environment, the robot is driving forwards until a wall is detected with the simulated IR sensors; the robot then rotates up to a random orientation, which is sampled from a normal distribution around $\pm 60°$ with $\sigma = 30°$. The figure shows the simulated environments, encoded maps, and true collision maps. Every simulation corresponds to a 60 minutes trial (at 50ms time step); the area of the simulated maps was $2 \times 2 m^2$. Qualitatively, the simulation results show a faithful representation of the environments.

To measure the quality of the learned maps, we assigned a score to each learned collision position. The score is evaluated using a correlation between the neuron-based map and the ground-truth of the collisions positions: If the learned
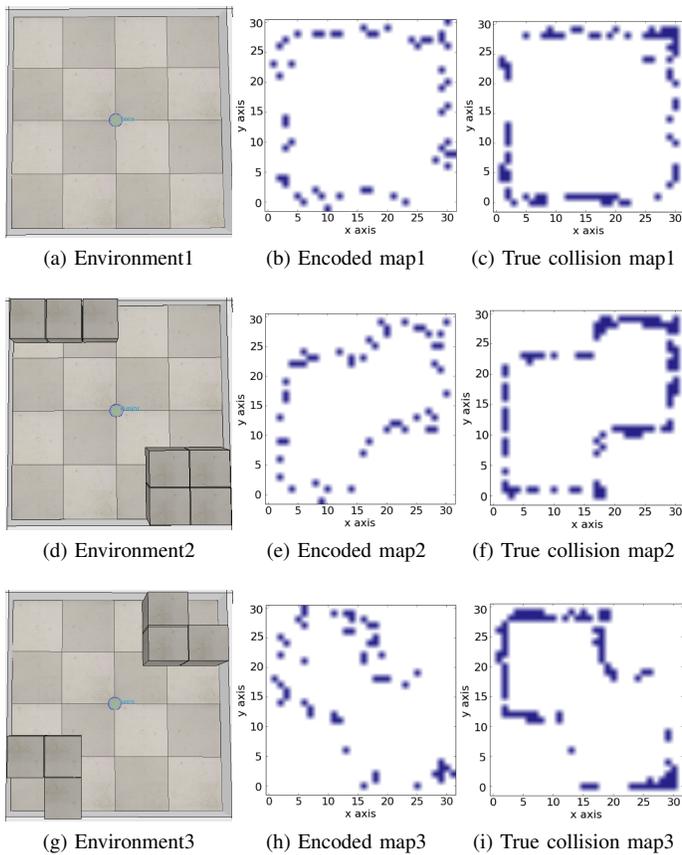
(a) Environment1   (b) Encoded map1   (c) True collision map1

(d) Environment2   (e) Encoded map2   (f) True collision map2

(g) Environment3   (h) Encoded map3   (i) True collision map3

Fig. 9: Learned maps after 60 min of simulation. **Left**: the simulated environments; **middle**: the learned maps reconstructed from plastic synaptic weights between PN and CON; **right**: the true collision map.

position is the same as the true collision position, the system scores 1 point, if the learned position deviated from the true position by 1 neuron unit in any direction, the system scores 0.5 point, etc. The score of the encoded maps grows linearly with simulation time, with a slope of 0.25-0.3 of the ideal score. The constant rate of error accumulation is due to discretization errors in both heading direction and position networks.

C. Dynamic mapping

In this section, we demonstrate how unlearning (depression) in plastic synapses can be used to update the map in a changing environment. Fig. 10 shows a 35 minute simulation, starting with an environment as shown in Fig. 10(a). After 5 minutes of simulation time (at 50ms simulation time step), the inside walls are removed while the robot continues to navigate. Fig. 10(c) shows the final environment.

After 5 minutes of simulation, plastic synapses encode the map as shown in Fig. 10(b). Fig. 10(d) shows the encoded map at the end of the simulation. These results show that the previously learned collision positions are unlearned after the walls are removed, if the robot passes the respective position without encountering a collision. Our path integration and map formation SNN is thus not only able to learn the map of

(a) Initial environment   (c) Environment after 5 minutes

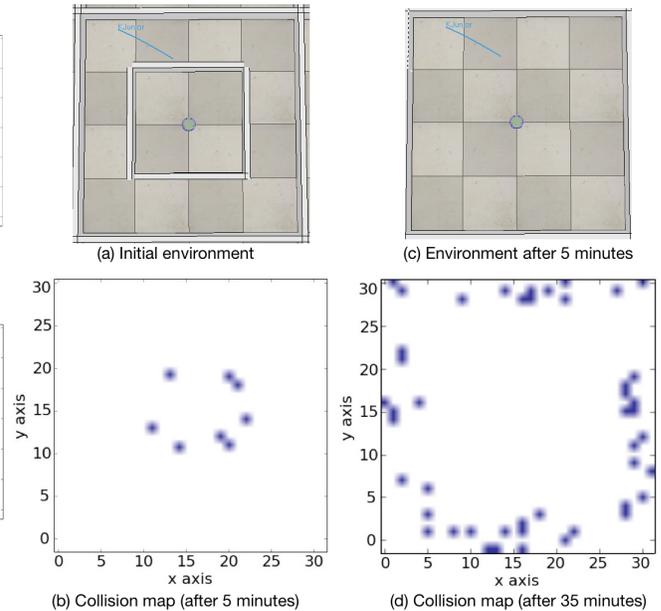(b) Collision map (after 5 minutes)   (d) Collision map (after 35 minutes)

Fig. 10: Demonstration of dynamic mapping. At the beginning, the robot is confined within walls in a small square space (a). The walls are removed after 5 minutes forming a larger square (c). The robot is able to unlearn the previous map.

a static environment, but also to update the map in a dynamic environment, using synaptic depression and potentiation.

D. Neuromorphic realization

Fig. 11 shows activity of spiking neurons on the neuromorphic device DYNAP [34] that were wired up to form the path-integrating architecture. The first plot in the figure shows the arrangement of neurons on the 4 chips of the DYNAP (each chip is divided into 4 cores of 256 neurons). Figs. 11 (2-6) show snapshots of activity on the chip as the robot turns left (0-9s), stops turning and continues moving to the west (11s), and then turns to the right, moving in the East direction after 24 seconds, Fig. 11(6). In the PN and IPN populations, one can observe the activity moving according to the HD population, which, in its turn, is driven by the TR or TL populations. The activity "bump" in the PN population is marked with a black circle around the purple spikes.

A SNN realized with neuromorphic hardware is thus able to represent and update the position on a map based on the angular velocity information using computation realized with silicon spiking neurons only.

E. Learning the map using plastic synapses on ROLLS

Finally, Fig. 12 demonstrates learning of collisions in plastic synapses on the ROLLS chip. Here, The PN neurons (#34-88) are activated one after another in a succession, resulting in spikes moving along the PN up and down. When a collision is sensed with the bumper (marked with red dots in the lower part of the plot) the collision neurons (#128-144) are stimulated and spike. At $t > 80s$, no more collisions are
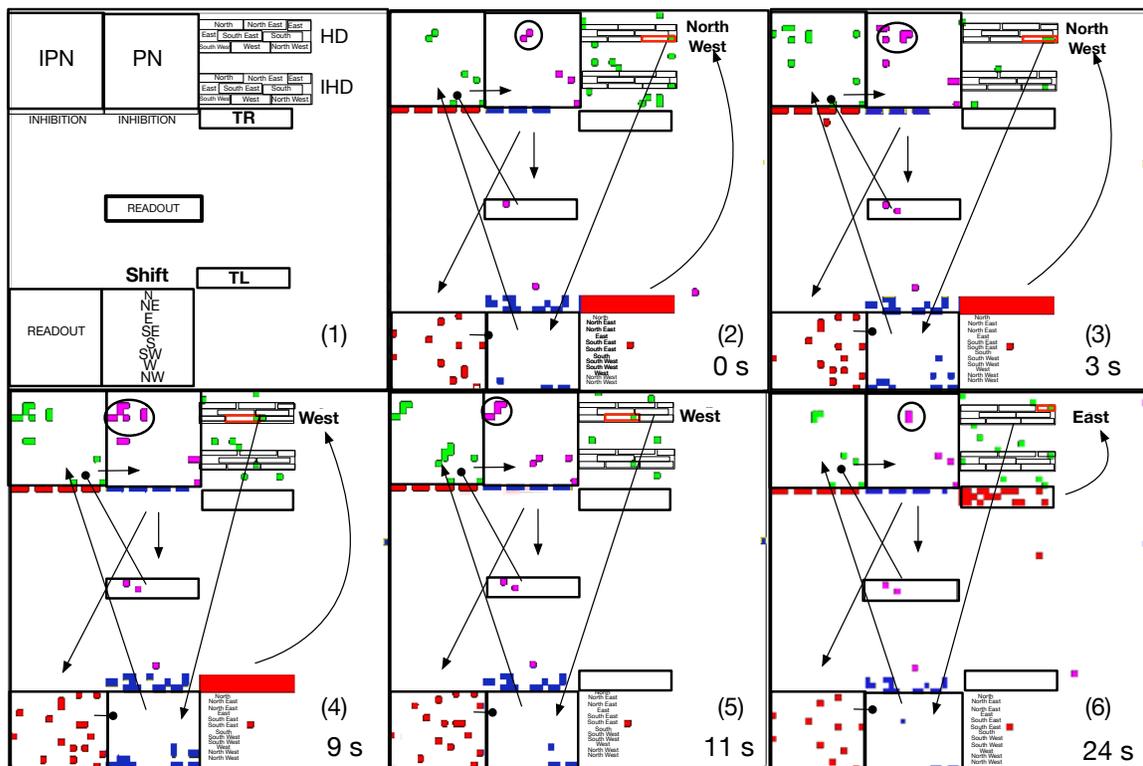
Fig. 11: (1): Neural populations configured on the DYNAP board. (2-6): snapshots of spiking activity during an experiment. Colored dots are neurons that spiked during the snapshot (see main text for details).

sensed. However, the collision population is activated via the potentiated plastic synapses whenever the learned position in the PN is active.

This result serves as a proof of feasibility of a hardware implementation of map formation, robustness and precision of the neuromorphic spatial representation will be validated and reported in future work.
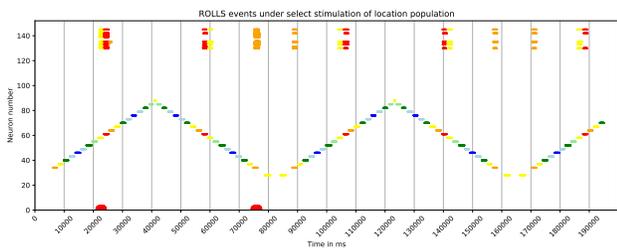

Fig. 12: Learning of two collision-position associations on the neuromorphic device ROLLS.

V. CONCLUSION

Drawing inspiration from biological neural networks that play a role in the remarkable ability of animals to navigate in novel environments could lead to new efficient solutions to the navigational tasks in robotics using embedded, ultra low-power, and compact computing devices. The proposed SNN architecture solves the localization and mapping problem using solely spiking neurons and plastic synapses. This architecture can thus be realized completely on the mixed-signal analog/digital neuromorphic device. This system could

potentially lead to an efficient SLAM solution consuming $< 10mW$ only.

Several steps are still required to achieve the goal of a fully-fledged neuromorphic SLAM. First, scaling-up of the developed architecture to better resolve all spatial representations in neural populations requires neuromorphic devices with more neurons than the prototypes currently available in our research lab. Industrial labs have started producing larger-scale devices [35], [36] that offer potential substrate for a high-resolution SLAM. The inherently parallel and event-based nature of neuromorphic computing facilitates up-scaling without increase in computing time. Second, to achieve a fully neuromorphic solution that would feature the target power efficiency values, direct interfaces to robotic sensors and motors are needed. While such interfaces exist for event-based sensors [34], spike-based motor control is an active area of research that is yet in its beginning [37].

Finally, both detection of the loop-closure events and use of the estimated errors to calibrate the path integration and correct the map require development of new spiking neural network architectures that enable "autonomous", online learning and adaptation. We have made first steps toward such architectures using continuous attractor dynamics [38], [39], [31]; their realization in neuromorphic hardware is yet an outstanding goal.

References

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots, Second Edition*. MIT Press, Cambridge MA, 2011.

[2] R. Hoffmann, D. Weikersdorfer, and J. Conradt, "Autonomous indoor exploration with an event-based visual SLAM system," *2013 European Conference on Mobile Robots, ECMR 2013 - Conference Proceedings*, pp. 38–43, 2013.

[3] F. Galluppi, C. Denk, M. C. Meiner, T. C. Stewart, L. a. Plana, C. Eliasmith, S. Furber, and J. Conradt, "Event-based neural computing on an autonomous mobile platform," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2862–2867, 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6907270

[4] N. Waniek, J. Biedermann, and J. Conradt, "Cooperative SLAM on small mobile robots," *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, pp. 1810–1815, 2015.

[5] M. Collett, "Spatial memories in insects," 2009.

[6] G. Wyeth and M. Milford, "Spatial cognition for robots," *IEEE Robotics & Automation Magazine*, vol. 16, no. 3, pp. 24–32, 2009.

[7] N. Cuperlier, M. Quoy, and P. Gaussier, "Neurobiologically inspired mobile robot navigation and planning," *Frontiers in Neurorobotics*, vol. 1, no. NOV, 2007.

[8] J. L. Krichmar, D. A. Nitz, J. A. Gally, and G. M. Edelman, "Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 6, pp. 2111–6, 2005. [Online]. Available: http://www.pnas.org/content/102/6/2111

[9] A. Barrera and A. Weitzenfeld, "Biologically-inspired robot spatial cognition based on rat neurophysiological studies," *Autonomous Robots*, vol. 25, no. 1-2, pp. 147–169, 2008.

[10] A. Arleo and W. Gerstner, "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity." *Biological cybernetics*, vol. 83, no. 3, pp. 287–99, 2000. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/11007302

[11] M. Milford, G. Wyeth, and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," *Robotics and Automation, IEEE International Conference of (ICRA)*, pp. 403–408, 2004. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1307183

[12] M. Mulas, N. Waniek, and J. Conradt, "Hebbian Plasticity Realigns Grid Cell Activity with External Sensory Cues in Continuous Attractor Models," *Frontiers in Computational Neuroscience*, vol. 10, no. February, pp. 1–11, 2016. [Online]. Available: http://journal.frontiersin.org/Article/10.3389/fncom.2016.00013/abstract

[13] C. Axenie and J. Conradt, "Cortically inspired sensor fusion network for mobile robot egomotion estimation," *Robotics and Autonomous Systems*, vol. 71, pp. 69–82, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2014.11.019

[14] S. Koziol, S. Brink, and J. Hasler, "A neuromorphic approach to path planning using a reconfigurable neuron array IC," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 12, pp. 2724–2737, 2014.

[15] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt, "Event-based 3D SLAM with a depth-augmented dynamic vision sensor," in *Proc. - IEEE Int. Conf. Robot. Autom.*, 2014, pp. 359–364.

[16] G. Indiveri, E. Chicca, and R. J. Douglas, "Artificial Cognitive Systems: From VLSI Networks of Spiking Neurons to Neuromorphic Cognition," *Cognitive Computation*, vol. 1, no. 2, pp. 119–127, 2009. [Online]. Available: http://www.springerlink.com/index/10.1007/s12559-008-9003-6

[17] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, D. Sumislawska, G. Indiveri, and G. Indiveri, "A Re-configurable On-line Learning Spiking Neuromorphic Processor comprising 256 neurons and 128K synapses," *Frontiers in neuroscience*, vol. 9, no. February, 2015.

[18] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. a. Merolla, and K. Boahen, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, may 2014.

[19] E. Neftci, E. Chicca, G. Indiveri, and R. Douglas, "A Systematic Method for Configuring VLSI Networks of Spiking Neurons." *Neural computation*, vol. 23, no. 10, pp. 2457–2497, 2011.

[20] E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. J. Douglas, "Synthesizing cognition in neuromorphic electronic systems." *Proc Natl Acad Sci U S A*, vol. 110, no. 37, pp. E3468–76, 2013.

[21] Y. Sandamirskaya, "Dynamic Neural Fields as a Step Towards Cognitive Neuromorphic Architectures," *Frontiers in Neuroscience*, vol. 7, p. 276, 2013.

[22] M. Cartiglia, R. Kreiser, and Y. Sandamirskaya, "A neuromorphic approach to path integration: a head direction spiking neural network with visually-driven reset," in *IEEE Synposium for Circuits and Systems, ISCAS*, 2018, submitted.

[23] R. Kreiser, T. Moraitis, Y. Sandamirskaya, and G. Indiveri, "On-chip unsupervised learning in winner-take-all networks of spiking neurons," in *Biological Circuits and Systems (BioCAS)*, 2017.

[24] W. Gerstner, W. M. Kistler, and Werner M, *Spiking Neuron Models: Singel Neurons, Populations, Plasticity*, 2002. [Online]. Available: http://www.langtoninfo.com/web{_}content/9780521813846{_}frontmatter.pdf

[25] K. Hardcastle, S. Ganguli, and L. M. Giocomo, "Cell types for our sense of location: where we are and where we are going," *Nature Neuroscience*, vol. 20, no. 11, pp. 1474–1482, 2017. [Online]. Available: http://www.nature.com/doifinder/10.1038/nn.4654

[26] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits." *Front Neurosci*, vol. 5, p. 73, 2011.

[27] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multi-core architecture with heterogeneous memory structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs)," no. August, 2017. [Online]. Available: http://arxiv.org/abs/1708.04198

[28] D. F. M. Goodman and R. Brette, "The brian simulator." *Frontiers in neuroscience*, vol. 3, no. 2, pp. 192–7, sep 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2751620{\&}tool=pmcentrez{\&}rendertype=abstract

[29] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1321–1326.

[30] J. M. Brader, W. Senn, and S. Fusi, "Learning Real-World Stimuli in a Neural Network with Spike-Driven Synaptic Dynamics," *Neural Comput. Massachusetts Inst. Technol.*, vol. 19, pp. 2881–2912, 2007.

[31] C. Strub, F. Wörgötter, H. Ritter, and Y. Sandamirskaya, "Using haptics to extract object shape from rotational manipulations," in *IROS*. IEEE, 2014, pp. 2179–2186.

[32] ——, "Correcting pose estimates during tactile exploration of object shape: a neuro-robotic study," in *ICDL-EPIROB*. IEEE, 2014, pp. 26–33.

[33] D. V. Buonomano and W. Maass, "State-dependent computations: spatiotemporal processing in cortical networks," *Nature Reviews Neuroscience*, vol. 10, no. 2, pp. 113–125, 2009.

[34] G. Indiveri, F. Corradi, and N. Qiao, "Neuromorphic Architectures for Spiking Deep Neural Networks," pp. 68–71, 2015.

[35] P. A. Merolla and et al., "Artificial brains. A million spiking-neuron integrated circuit with a scalable communication network and interface." *Science (New York, N.Y.)*, vol. 345, no. 6197, pp. 668–73, 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25104385

[36] M. Davies and et al., "Loihi: a Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[37] F. Perez-Pena, A. Linares-Barranco, and E. Chicca, "An approach to motor control for spike-based neuromorphic robotics," *IEEE 2014 Biomedical Circuits and Systems Conference, BioCAS 2014 - Proceedings*, pp. 528–531, 2014.

[38] C. Rudolph, T. Storck, and Y. Sandamirskaya, "Learning to reach after learning to look: a study of autonomy in learning sensorimotor transformations," in *IJCNN*, 2015.

[39] Y. Sandamirskaya and J. Conradt, "Learning Sensorimotor Transformations with Dynamic Neural Fields," in *International Conference on Artificial Neural Networks (ICANN)*, 2013.