Underwater Rat-SLAM with Memristive Spiking Neural Networks

1st Bernardo Manuel Pirozzo
INTELYMEC, Centro de Investigaciones en Física e Ingeniería del Centro UNICEN – CICpBA – CONICET
B7400JWI, Olavarría, Argentina
berpirozzo@gmail.com

2nd Mariano De Paula
INTELYMEC, Centro de Investigaciones en Física e Ingeniería del Centro UNICEN – CICpBA – CONICET
B7400JWI, Olavarría, Argentina
mariano.depaula@fio.unicen.edu.ar

3rd Sebastián Aldo Villar
INTELYMEC, Centro de Investigaciones en Física e Ingeniería del Centro UNICEN – CICpBA – CONICET
B7400JWI, Olavarría, Argentina
svillar@fio.unicen.edu.ar

4th Gerardo Gabriel Acosta
INTELYMEC, Centro de Investigaciones en Física e Ingeniería del Centro UNICEN – CICpBA – CONICET
B7400JWI, Olavarría, Argentina
ggacosta@fio.unicen.edu.ar

***Abstract*—Autonomous Underwater Vehicles (AUVs) are suitable platforms for a wide variety of applications in oceanic environments. However, to successfully carry out their tasks, they require a navigation system that can estimate their location with bounded error. Localization in underwater environments is a non-trivial problem with various proposed solutions, such as the use of buoys, computer vision, acoustic beacons, among others. Nevertheless, all these solutions require expensive sensors and hardware to keep positioning errors below a threshold. For these reasons, the use of bioinspired systems with cognitive capabilities is attractive and valuable for addressing this type of problem, as they allow for learning to fuse sensors and reset growing positioning errors through loop closure. To take advantage of these features in underwater environments, we present in this work a method for solving AUV localization using Rat-SLAM. Our proposal has two central modules, the Pose Cells (PC) network, implemented with Spiking Neural Networks (SNN), which allow us to have a form of information processing closer to the biological one and with the possibility of hardware implementation to have on-board Neuromorphic Computing (NC). Each spiking neuron in the SNN is modelled using the fourth element of electronics, the memristor. With this building block, a network of Pose cells estimates the 3D positioning coordinates (x, y, z) from linear velocities, while other network of Pose Cells estimates the heading (yaw) for each position in the plane, from the x and y components of the linear velocities and the z components of the angular velocity. The linear velocities are provided by a DVL. A monocular grayscale camera provides input to Local View Cells (LVC) network to perform loop closure when a known scene is seen again. The information from the LVC, together with that from the DVL and the activity in the PC-MLIF, is used to build the experience map, which will contain all the localization estimates. To evaluate the performance of this proposal, we carried out different trajectories using a Rexrov AUV in the Gazebo simulation environmet, which is part of the Robot Operating System (ROS). From these results, we obtained some conclusions and determined the future work of our research.**

***Index Terms*—Rat-SLAM, Memristors, Neuromorphic Computing, Neuroscience, Autonomous Underwater Vehicles.**

I. Introduction

Underwater environments are reservoirs of natural resources whose exploration and exploitation are extremely complex. The presence of currents, changing underwater lighting and visibility conditions, deterioration of certain types of materials caused by high sea salt content, and other factors lead to a reduction in operational capacity below sea level. In addition to the aforementioned variables, one of the most limiting and important factors to consider in underwater environments is the pressure exerted by the medium on the bodies within it. Since its magnitude increases with depth, a person would not be able to go beyond a certain limit. For this reason, using small manned submarines to reach deeper areas seems to solve the problem at first glance. However, this solution carries a high risk since any structural failure can cause an implosion, leading to the inevitable consequence of loss of human lives. A less risky alternative is using Remotely Operated Vehicles (ROVs). Nonetheless, the need for an operator to control the vehicle and a physical connection to the surface for communication and power supply limits their application to short-range operations. These disadvantages have driven the search for other solutions, which have come hand in hand with significant technological advancements, in the development of Autonomous Underwater Vehicles (AUVs). Features such as lack of crew, lack of physical connection to the surface, and high versatility, currently make them the preferred solution for tasks in a huge variety of fields, such as scientific study [1], offshore industry [2], defence [3], among others.

Being able to perform various tasks autonomously, AUVs require certain capabilities. Perhaps one of the most important capability is localization. Solving the localization of an AUV involves estimating the pose, consisting of the 3D spatial location and orientation, at each sampling (decision) instant. However, in underwater environments, this problem becomes

highly complex because the medium influences the robot's perception and navigation capabilities. Thus, it is not possible to directly localize the AUV using, for example, a GPS sensor. An alternative, initially suitable approach, is to use a Doppler Velocity Log (DVL) to obtain the position from the time integration of linear velocities. However, this localization mean leads to the serious consequence of accumulating error over time. This is unsuitable for long trajectories. For these reasons, this type of problem is addressed using alternative solutions, such as triangulation from a reference system consisting of buoys [4] or acoustic beacons [5], estimates derived from information extracted by Parallel Tracking and Mapping (PTAM) [6] from images obtained by computer vision [7], and others. Most of these approaches require expensive hardware and software resources to maintain the pose error estimates below a certain threshold.

Biologically, Neuroscience has demonstrated, through the analysis of the brains of rodents and other mammals, the existence of certain brain regions that, based on environmental perceptions provided by the five senses, either individually or combined, using for example, vision alone or vision and touch, are responsible for estimating the orientation and spatial location of the animal. Additionally, these regions inherently have the ability to recognize previously visited places, closing the loop when the animal returns to a familiar area. All these extremely attractive and desired characteristics in localization systems allow maintaining estimation errors within an acceptable threshold for robotic operations. These have been achieved with the development of the bioinspired algorithm known as Rat-SLAM. It consists of three modules: Experience Map (EM), Pose Cells (PC), and Local View Cells (LVC). In our work, the required environmental information was obtained from monocular grayscale images and estimates of speed and heading. This Rat-SLAM algorithm, as described, has been used to solve the localization of robots in aerial [8] and terrestrial [9] environments, demonstrating promising results.

However, despite the bioinspired origin of Rat-SLAM, it still lacks the ability to process information in the same way as the biological brain. In the latter, neuronal activity is governed by electrical impulses carried out through electrochemical processes. This form of processing inspired the development of Spiking Neural Networks (SNN) [10] in the field of Artificial Intelligence (AI). In this type of network, information is stored in the timing or frequency of the impulses, so neurons must be modeled in a way that replicates biological behavior. An attractive way to achieve this, particularly from an energy consumption perspective, is through the development of hardware known as Neuromorphic Computing (NC). This discipline studies the modeling of biological structures with electronic circuits. An example of such circuits is the parallel RC circuit that models the Leaky Integrate-and-Fire (LIF) neuron [11]. However, the emergence of the memristor as the fourth element of electronics [12], made it possible to replace the resistor with a memristor in the previous neuronal model, incorporating the inherent properties of intrinsic memory and low energy consumption of the memristor, giving rise to the Memristive Leaky Integrate-and-Fire (MLIF) model [13].

In this work, we propose to address the localization of an AUV for underwater environments using an adapted version of the Rat-SLAM algorithm. Our modifications are, on one hand, incorporating the MLIF model into the central module of this algorithm to represent the Pose Cells (PC), in a clear case of NC application. Thus, we will refer to the resulting PCs as PC-MLIF. They exhibit information processing capabilities closer to biological ones, component neurons that incorporate the intrinsic properties of the memristor, and the potential for implementation in hardware devices. On the other hand, as our goal is to apply this algorithm to underwater environments, it is mandatory to estimate at least four variables that define the localization. These are the 3D spatial location and the heading (yaw). To this end, we propose two PC-MLIF networks, one to estimate the spatial location and another to estimate the orientation. In summary, the contributions of our work are as follows:

- Solving the localization of AUVs in underwater environments using a modified Rat-SLAM algorithm. This modification consists of implementing the PCs with the MLIF spiking neuron model.
- Encoding the poses of the AUV in the frequencies or timings of these spikes.
- Validating the possibility of implementing the PC-MLIF in a hardware device to increase the energy efficiency of the positioning system.
- Providing two PC-MLIF networks with intrinsic memory that relate the excitation level of each neuron to a pose of the AUV.

II. Neuronal Model

One of the earliest representations of biological neurons proposed a Processing Element (EP) that performs the sum of its weighted inputs. This sum is then passed through a non-linear function or activation function. When a predefined threshold (v_{th}) is exceeded, the EP fires a pulse, exhibiting typical on-off behavior. After firing, the EP remains refractory to new excitations for a period, and its potential is reset to a specified value (v_{reset}), akin to neurons studied in biology by McCulloch and Pitts [14]. This model lacks a notion of when input signals occur. Information is stored in the value of intersynaptic weights, providing significant plasticity to model the transfer of input to output signals. Imagine what would happen if we could incorporate temporal information into neuron input signals. Information could be stored in the timing or frequency of spikes. Thus, neuron activity would be characterized by spike generation at specific times, in response to the inputs received by the EP. This form of communication between neurons in this model would occur via these spikes, enabling richer and more efficient temporal coding compared to models based solely on binary activation. These considerations encompass the Spiking Neural Networks (SNN) model, as presented in [11]. They take into account the timing and nature of neuronal signals, making them a

closer model to biological neural networks. In fact, they have provided significant insights into information encoding in neural networks, memory, their dynamic behavior, and more recently, deep learning [15]. SNNs can be implemented in hardware using an RC circuit. The resistor can be replaced by a memristor, known as the MLIF neuron model [13]. Fig.1 shows the neuron circuit, where I_{ext} is the excitation current crossing the neuron membrane; C is a capacitor, representing the bilipid membrane, with a value of $9e^{-7}$ Farads; M is a memristor, analogous to the ionic channel, with characteristics explained in the previous section, and v_{th} is the threshold voltage set to 20mV. When the voltage value in the memristor exceeds this threshold, the switch closes, resetting the neuron potential to 0V, and then reopens until the previous condition is met again.

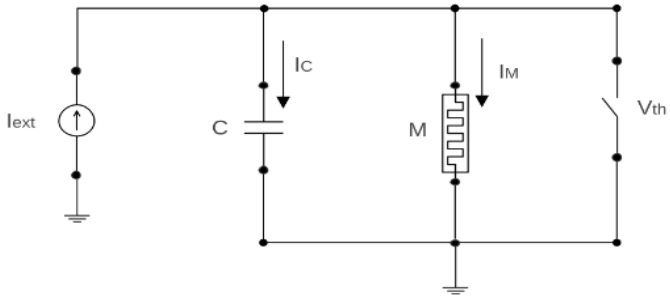


Fig. 1: Circuit diagram of the implemented MLIF neuronal model.

The Eq.(1) is the differential equation that models the behavior of this type of neuron, where $\tau = M(t)C$. To solve this equation, the finite difference method was used, as shown in the Eq.(2), where Δt is the sampling period, with a value adopted in this work of $1e^{-3}$ s.

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = \frac{1}{\tau}(-V(t) + I_{ext}M(t)) \tag{1}$$

$$V(t + \Delta t) = \frac{\Delta t}{\tau}(-V(t) + I_{ext}M(t)) + V(t) \tag{2}$$

In this way, the MLIF neuron will generate different peaks in membrane potential in response to the external stimulus applied, given by the current I_{ext}. A case worth analyzing is when the applied external stimulus has the form of a step function. The neuron responds with an increase in membrane potential until the reset voltage (v_{th}) is reached, at which point the membrane potential is reset to 0V, as shown in Fig.2. This test allows us to visualize the impulse generation process and how the reset acts when the membrane potential reaches the threshold voltage. Finally, when the neuron is stimulated with a train of pulses of different amplitudes, we observe that during the first excitation condition, the membrane potential responds with an increase in its value for the 10 milliseconds during which the stimulus is applied, and decreases during the following 10 milliseconds where no stimulus is applied to the neuron, as observed during the first 200 ms in Fig.3. Note that in this excitation condition, the duration of each pulse is not sufficient for the membrane potential to reach the threshold voltage, so the reset does not occur. Then, there is a pause in excitation between 200 to 300 ms, where the neuron's behavior is similar to that shown in Fig.**??**. After this time interval, we change the amplitude of the excitation current. In this case, the current I_{ext} takes a non-zero value for 18 ms, and then returns to zero for the next 10 ms. For this condition, the membrane potential increases until it reaches the threshold value, where the potential is reset to zero, as seen in the interval between 300 to 600 ms in Fig.3.

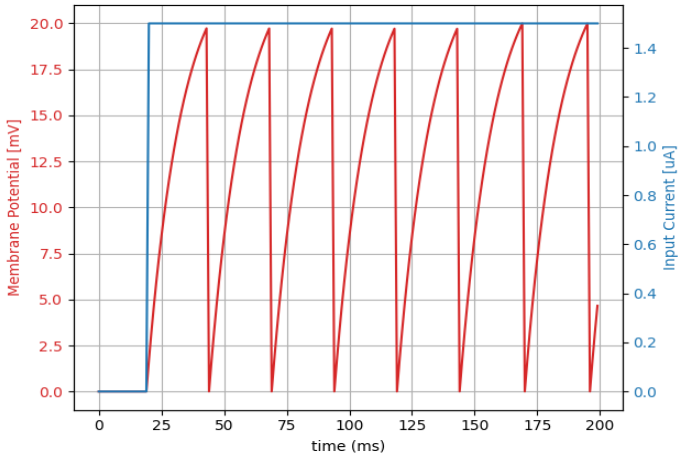


Fig. 2: Response of the MLIF neuron when an external stimulus is applied by a current step of magnitude equal to $1.5e^{-6}A$.

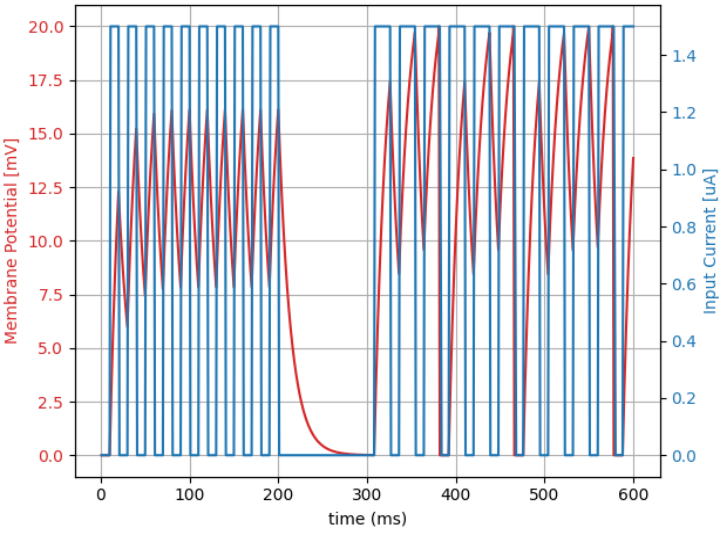


Fig. 3: Response of the MLIF neuron to an external excitation current stimulus given by a pulse train of varying amplitudes, with a maximum value of $1.5e^{-6}A$.

III. RAT-SLAM ADAPTED TO UNDERWATER ENVIRONMENTS

The Rat-SLAM consists of three main modules, with the most important being the PC network, which encodes the robot's different poses. The initial applications of this algorithm were in terrestrial environments, with three DoF (position in the plane and heading). However, in underwater or aerial environments, the robot's DoF increase from three to six, as it can now move in 3D space and adjust its orientation angles: Roll, Pitch, and Yaw in Euler representation. This increase in DoF increases the complexity of the problem, as it now requires estimating six variables to locate the robot in the environment. However, as the AUV used in our experimental tests can independently move along z axis, due to constructive features, we could simplify the estimation of pose to spatial location (x, y , z) and heading or Yaw. This enables to use two Pose Cell (PC) networks: one dedicated to encode spatial location and another devoted to store the heading corresponding to each position in the plane (x, y).

In Fig.4, on the left, the interaction among environment, perception systems, navigation, and robot control is shown. Among these three systems, the first one gathers information from the surrounding underwater environment, allowing the second module to estimate, with bounded error, the robot's pose. This estimate is then used by the third module, in charge of determining control actions that should be applied to the AUV to ensure movements for either exploring the environment or following a mission plan. Note that poor pose estimations can lead to significant deviations in trajectories.

Many contributions from navigation systems for AUVs have shown great results in underwater environments. However, the aforementioned features of cognition, information processing closer to biological brain functions, loop closure capabilities upon returning to a known area, among others available in Rat-SLAM, are very appealing for solving the localization of AUVs. For this reason, in this work, we decided to adapt this algorithm to operate in underwater environments, as shown on the right side of Fig.4, detailing our perception and navigation system. The first module provides monocular grayscale images, linear velocity, angular velocity, and heading. These come from a monocular camera and a DVL mounted on the experimental AUV. This environmental information is taken by the second module, where the LVCs receive an image of the environment and the activity centroids of both PC-MLIF networks at each sampling time, establishing a relationship between these three quantities. The LVCs are composed of cells that store a visual template for each visited location. When a familiar scene is seen again, the cell corresponding to that view is activated; otherwise, a new cell is created to represent the new view. These visual templates are obtained using the intensity profile of the grayscale image. When the difference between the stored profiles and the current one exceeds a threshold, the current template is stored in a new cell. The comparison between the current profile *Ij* and the stored profiles *Ik* is carried out using the average absolute intensity difference function as shown in Eq.(3). This comparison is performed over a small range of pixel displacements *s* to provide some robustness against rotation.

$$k_m = \arg\min_k f\left(s, I^j, I^k\right) \tag{3}$$

Regarding the two PC-MLIF networks, they are arranged in a toroidal 3D grid cell prismatic array [16]. Their activity is governed by the dynamics of Continuous Attractor Networks (CAN) [17]. One of the PC-MLIF network encodes the location in 3D space, while the other encodes the heading corresponding to each position in the plane (*x, y*) of the AUV. Each of the component neurons in these networks uses the model explained in Section II, where the connection via excitatory and inhibitory connections is governed by the excitation current entering each neuron, allowing a unique group of units to be active in each possible state. Thus, the so-called activity packet or energy packet is established, with its centroid encoding the estimation of the current pose of the AUV. The activity in both networks follows the 3D CAN dynamics, consisting of three stages as explained in the sequel. The difference in the update process carried out in each PC-MLIF network lies in the speed used for updating activity along the *z*-axis. Therefore, the PC-MLIF network encoding the 3D pose uses linear velocity, while the other uses angular velocity. The three stages of update are: local excitation, local-global inhibition, and activity normalization. The first stage requires a matrix of excitation weights modeled by three-dimensional Gaussian distributions, as shown in Eq.(4). In this expression, var_{axis} represents the variance of each coordinate axis, constants for spatial distribution. The variation in excitation current is given by Eq.(5), where $n_{\hat{x}}$, $n_{\hat{y}}$ and $n_{\hat{\theta}}$ are the dimensions of the activity matrix, and $\varepsilon_{a,b,c}$ denotes the multiplication of the Gaussian distribution for each axis.

$$\varepsilon_j = \left(\frac{1}{var_{axis}\sqrt{2\pi}}\right) e^{-j^2/2var_{axis}^2} \tag{4}$$

$$\Delta i_{PC-MLIF}^{(exc)} = \sum_i^{n_{\hat{x}}} \sum_j^{n_{\hat{y}}} \sum_k^{n_{\hat{\theta}}} P_{\hat{x},\hat{y},\hat{\theta}}^{(t)} \cdot \varepsilon_{a,b,c} \tag{5}$$

The indices a, b and c, represent the distances between the coordinates $\hat{x}$, $\hat{y}$ and $\hat{\theta}$, and are calculated using the expressions shown in Eq.(6).

$$\begin{aligned} a &= (\hat{x} - i) \cdot mod(dim(\hat{x})) \\ b &= (\hat{y} - j) \cdot mod(dim(\hat{y})) \\ c &= (\hat{\theta} - k) \cdot mod(dim(\hat{\theta})) \end{aligned} \tag{6}$$

Once the variation in excitation current is obtained, the variation in inhibition current is calculated using an inhibitory weight matrix $\psi_{a,b,c}$ modeled in the same way as $\varepsilon_{a,b,c}$, but with negative weight values, as shown in Eq.(7).

$$\Delta i_{PC-MLIF}^{(inh)} = \sum_i^{n_{\hat{x}}} \sum_j^{n_{\hat{y}}} \sum_k^{n_{\hat{\theta}}} P_{\hat{x},\hat{y},\hat{\theta}}^{(t)} \cdot \psi_{a,b,c} \tag{7}$$

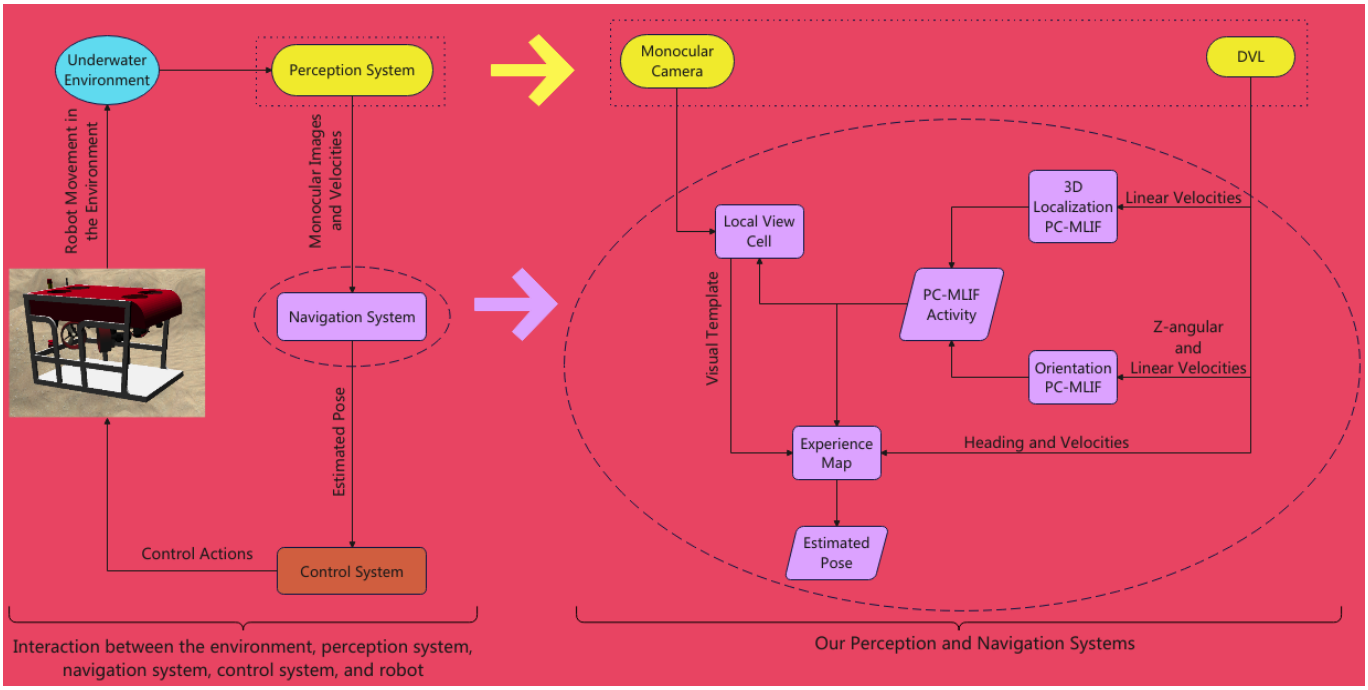


Fig. 4: Block diagram and information flow of our Rat-SLAM proposal for underwater environments.

With these two processes completed, the current applied to each neuron of the PC-MLIF network in the next sampling time is given by Eq.(8). As a result of applying a current to each neuron, the membrane potential of each cell is obtained. Each potential represents the activity matrix of the PC-MLIF network, from which a constant value is subtracted to perform global inhibition, and finally, it is normalized.

$$i^{(t+1)}_{PC-MLIF} = \Delta i^{(exc)}_{PC-MLIF} + \Delta i^{(inh)}_{PC-MLIF} \tag{8}$$

Up to this point, the process of updating activity in the PC-MLIF network has completed the three steps of 3D attractor dynamics, without considering the perceived information from the environment. To incorporate this information into the update process, two steps are performed, known as path integration and local view calibration. The first step projects the activity of the 3D cells to nearby cells by shifting the activity packet in the plane ($\hat{x}$, $\hat{y}$) according to linear translational velocity, and in the third dimension according to angular velocity representing the change in heading ($\hat{\theta}$). In Eq.(9), the change in the activity matrix due to environmental perception is shown, where $\delta_{\hat{x}_0}$, $\delta_{\hat{y}_0}$ and δ_{θ_0}, are calculated according to Eq.(10), with $k_{\hat{x}}$, $k_{\hat{y}}$, and $k_{\hat{\theta}}$ representing the initial offsets for each axis, and γ is calculated according to Eq.(11), where $\delta_{\hat{x}_f}$, $\delta_{\hat{y}_f}$ and δ_{θ_f} are the final offsets given by Eq.(12).

$$P^{(t+1)}_{l,m,n} = \sum_{x=\delta_{\hat{x}_0}}^{\delta_{\hat{x}_0}+1} \sum_{y=\delta_{\hat{y}_0}}^{\delta_{\hat{y}_0}+1} \sum_{\hat{\theta}=\delta_{\hat{\theta}_0}}^{\delta_{\theta_0}+1} \gamma \cdot P^{(t+1)}_{(l+x),(m+y),(n+\theta)} \tag{9}$$

$$\begin{aligned} \delta_{\hat{x}_0} &= k_{\hat{x}} \cdot v \cdot cos(\hat{\theta}) \\ \delta_{\hat{y}_0} &= k_{\hat{y}} \cdot v \cdot sin(\hat{\theta}) \\ \delta_{\hat{\theta}_0} &= k_{\hat{\theta}} \cdot \omega \end{aligned} \tag{10}$$

$$\gamma = f(\delta_{\hat{x}_f}, \hat{x} - \delta_{\hat{x}_0}) f(\delta_{\hat{y}_f}, \hat{y} - \delta_{\hat{y}_0}) f(\delta_{\hat{\theta}_f}, \hat{\theta} - \delta_{\hat{\theta}_0}) \tag{11}$$

$$f(a,b) = \begin{cases} a & \text{if} \quad b = 1 \\ 1-a & \text{if} \quad b = 0 \end{cases}$$

$$\begin{aligned} \delta_{\hat{x}_f} &= k_{\hat{x}} \cdot v \cdot cos(\hat{\theta}) - \delta_{\hat{x}_0} \\ \delta_{\hat{y}_f} &= k_{\hat{y}} \cdot v \cdot sin(\hat{\theta}) - \delta_{\hat{y}_0} \\ \delta_{\hat{\theta}_f} &= k_{\hat{\theta}} \cdot \omega - \delta_{\hat{\theta}_0} \end{aligned} \tag{12}$$

The second step is local view calibration. It is aimed at resetin cumulative errors from the path (trajectory) integration. To achieve this, the LVCs are associated with the activity of the two PC-MLIF networks such that when the robot returns to a previously visited location, the LVC associated with this location activates the same neurons in the PC-MLIF network as when this location was firstly visited, through an excitatory linkage (not shown in Fig.4 for clarity). To learn the relationship between the local view cell vector and the activity in the PC-MLIFs, the matrix ψ stores their synaptic interconnections, using a version of Hebb's law for learning. The connection between $P_{\hat{x},\hat{y},\hat{z},\hat{\theta}}$ and V_i is obtained through Eq.(13), where ξ is the learning rate and V_i is the activity vector of the LVC.

$$\psi^{t+1}_{i,\hat{x},\hat{y},\hat{z},\hat{\theta}} = max(\psi^{t}_{i,\hat{x},\hat{y},\hat{z},\hat{\theta}}, \xi V_i P_{\hat{x},\hat{y},\hat{z},\hat{\theta}}) \tag{13}$$

The Experience Map (EM) is a topological map composed of many individual experiences, e, each connected by transitions, t. An individual experience e_i, given by Eq.(14), contains the positions of the active cells in the PC-MLIF network, ($P^i_{\hat{x},\hat{y},\hat{z},\hat{\theta}}$), the activity vector of the LVC (V^i), and the estimated spatial position ($\mathbf{p}^i$) derived from measurements, i.e. positions from speed integration. The first experience is created at an arbitrary starting point, and subsequent experiences are built from the previous one through transitions.

$$e_i = \{P^i_{\hat{x},\hat{y},\hat{z},\hat{\theta}}, V^i, \mathbf{p}^i\} \tag{14}$$

When the comparison metric of activity in the two preceding modules exceeds a threshold S_{max}, a new experience is created with an associated transition. The metric S is obtained using Eq.(15), where μ_p and μ_v are constants that weigh the respective contributions of the two PC-MLIF netowrks and the LVC to the final score.

$$S = \mu_p |P^i_{\hat{x},\hat{y},\hat{z},\hat{\theta}} - P_{\hat{x},\hat{y},\hat{z},\hat{\theta}}| + \mu_v |V^i - V| \tag{15}$$

The transition t_{ij} stores the change in position measured from the integration of velocities obtained from the DVL. Eq.(16) shows the stored transitions, where $\Delta\mathbf{p}^{ij}$ represents the change in the estimated vehicle pose. The link between the previous experience e_i and the new experience e_j is formed according to Eq.(17).

$$t_{ij} = \{\Delta\mathbf{p}^{ij}\} \tag{16}$$

$$e_i = \{P^j_{\hat{x},\hat{y},\hat{z},\hat{\theta}}, V^j, \mathbf{p}^i + \Delta\mathbf{p}^{ij}\} \tag{17}$$

Note that a loop closure occurs when the activity in both networks (PC-MLIF and LVC) sufficiently match a stored experience. When this happens, it is highly unlikely that the accumulated change in position from transitions leads the loop closure experience to match previously stored experiences for the same position. To achieve this match, the positions of all experiences are updated using Eq.(18), where α is a constant correction rate, N_f is the number of links from experience e_i to other experiences, and N_t is the number of links from other experiences to experience e_i.

$$\Delta\mathbf{p}^i = \alpha \left[\sum_{j=1}^{N_f} \left(\mathbf{p}^j - \mathbf{p}^i - \Delta\mathbf{p}^{ij}\right) + \sum_{k=1}^{N_t} \left(\mathbf{p}^k - \mathbf{p}^i - \Delta\mathbf{p}^{ki}\right) \right] \tag{18}$$

IV. Experimental Results

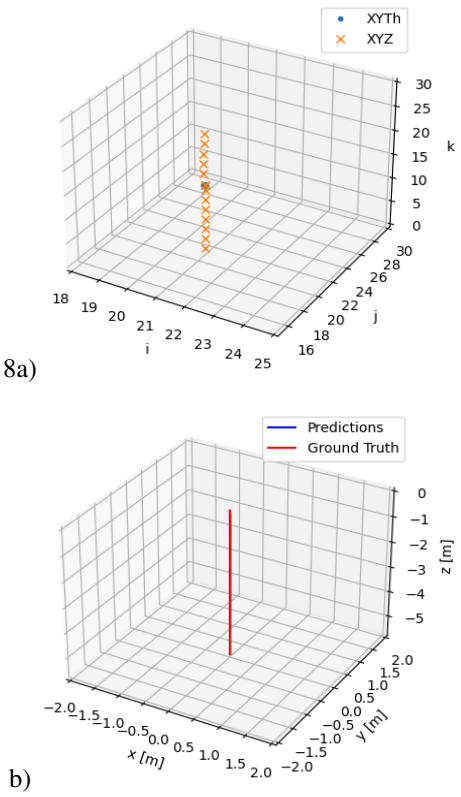
In this section, we present the experimental results conducted to validate our proposal. We used the Robot Operating System (ROS) [18] due to its multiple advantages, like versatility in the admissible programming language, availability of the Gazebo and RViz simulators that are installed along with ROS, the capability of simulating various environments and robots, rapid transition from simulation to a real robot, and others. In this work, the underwater environment simulator used is uuv_simulator, available at (https://github.com/arturmiller/uuv_simulator/tree/noetic).

The memristor model used for these trials, is the one proposed by HP Labs [19]. Physically, it is a passive two-terminal element. This component consists of two platinum (Pt) electrodes that contain between them a material of length D composed of two layers. One of these layers is highly resistive titanium dioxide (TiO_2), and the other is highly conductive oxygen-poor titanium dioxide (TiO_{2-X}). When an electric current flows in one direction, the boundary between the two materials, *w*, shifts, producing an increase in conductivity due to a higher percentage of the TiO_{2-X} layer. Conversely, when the electric current flows in the opposite direction, the resistance of the memristor increases due to the shift of *w* which increases the percentage of the TiO_2 layer. From the above, it is deduced that when there is no current flow, *w* does not change and the memristor retains its state, that is, it "remembers" the last adopted resistance value.

To evaluate the performance of the Rat-SLAM developed in this work, we propose three tests that allow us to observe the behavior of both PC-MLIF network along with the comparison between the actual trajectory (Ground Truth - GT), and the trajectory estimation carried out by the EM. The first of these evaluations was a dive, shown in Fig.5, which allows us to see the modification of the activity centroid of the PC-MLIF network that encodes the spatial location, while the PC-MLIF network that encodes the orientation shows no modification in its activity, as shown in Fig.5a). In Fig.5b), the trajectory estimation is shown together with the GT, where the descent in a straight line from sea level to five meters depth can be seen. The second test we performed consists of rotating the robot 360° around the z-axis, as seen in Fig.6. In this case, the activity in both PC-MLIF is contrary to the previous case, as shown in Fig.6a), the activity of the PC-MLIF responsible for encoding the orientation changes, while the other shows no variation in its activity. In Fig.6b), it is shown that there was no change in the spatial location since the robot rotated around one of its axes. Finally, the last evaluation consisted of a trajectory with multiple movements, shown in Fig.7. This test allows us to observe how the activities in both PC-MLIF are modified, shown in Fig.7a), where the actual and estimated trajectory is shown in Fig.7b).

The error metric used to evaluate the quality of the trajectory estimates is the Root Mean Squared Error (RMSE). This is calculated with Eq.(19), where N is the number of estimates, GT and EP are the actual and estimated trajectories, respectively.

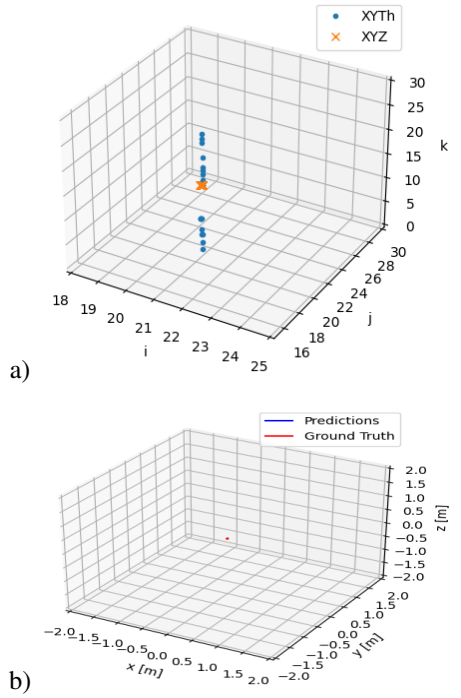
$$RMSE = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N}(GT_i - EP_i)^2\right)} \tag{19}$$



8a)

b)

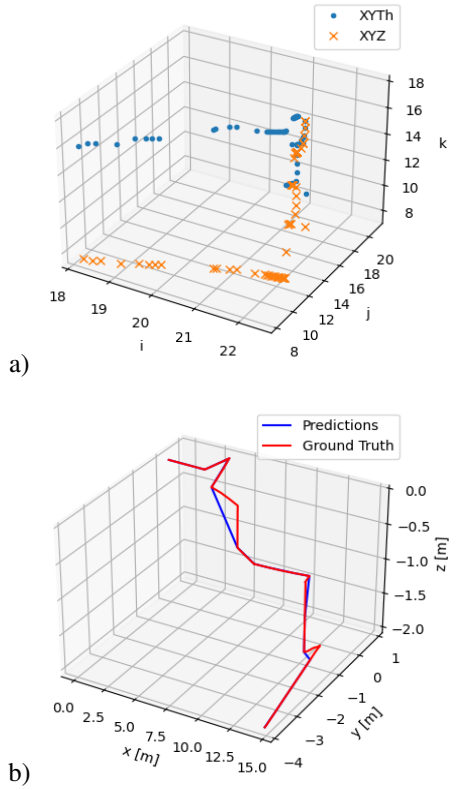
Fig. 5: a) Centroid of pose cell activity for cells i, j, k; b) straight-line descent trajectory (RMSE=0.06m).



a)

b)

Fig. 6: a) Centroid of pose cell activity for cells i, j, k; b) 360° turn trajectory.



a)

b)

Fig. 7: a) Centroid of pose cell activity for cells i, j, k; b) multi-move trajectory (RMSE=1.57m).

V. Conclusions

In this work, we presented a bioinspired approach to localization in underwater environments using monocular images and a DVL. This Rat-SLAM approach is efficient enough to estimate the pose of an underwater robot. Using the MLIF neural model will allow to make on-line computation to cope with the SLAM problem, in a bioinspired way. In addition, the possibility of constructing the PC-MLIF network on a dedicated hardware device will decrease the power consumption and speed up computation, like other neuromorphic computation approaches. These features are crucial in autonomous robotic applications since AUVs, on one hand, have limited onboard software resource capacities, and on the other hand, they have limited energy availability.

The tests carried out in this work demonstrate that the present approach can solve the localization of an AUV with an acceptable error for robotic applications. Moreover, inferring poses from linear and angular velocities and storing them in the PC-MLIF network allows for loop closure when the AUV revisits a previously traveled location, thereby reducing estimation errors. In summary, this Rat-SLAM approach is efficient enough to estimate in real-time the pose of an underwater robot, as well as to construct a semi metric map of the environment covered from the Experience Map (EM).

REFERENCES

[1] Mahdi Choyekh, Naomi Kato, Timothy Short, Masahiro Ukita, Yasuaki Yamaguchi, Hidetaka Senga, Muneo Yoshie, Toshinari Tanaka, Eiichi Kobayashi, and Hajime Chiba. Vertical water column survey in the gulf of mexico using autonomous underwater vehicle sotab-i. *Marine Technology Society Journal*, 49(3):88–101, 2015.

[2] J Jamieson and L Wilson. Effective launch and recovery of autonomous underwater vehicles. In *Offshore Technology Conference*, pages OTC–25664. OTC, 2015.

[3] Scott Reed, Yvan Petillot, and Judith Bell. Mine detection and classification in side scan sonar. *Sea Technology*, 45(11):35–39, 2004.

[4] Andrea Caiti, Andrea Garulli, Flavio Livide, and Domenico Prattichizzo. Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach. *IEEE Journal of Oceanic Engineering*, 30(1):140–152, 2005.

[5] Bruno M Ferreira, Paula A Graça, José Carlos Alves, and Nuno A Cruz. Single receiver underwater localization of an unsynchronized periodic acoustic beacon using synthetic baseline. *IEEE Journal of Oceanic Engineering*, 2023.

[6] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 83–86. IEEE, 2009.

[7] Adrian Manzanilla, Sergio Reyes, Miguel Garcia, Diego Mercado, and Rogelio Lozano. Autonomous navigation for unmanned underwater vehicles: Real-time experiments using computer vision. *IEEE Robotics and Automation Letters*, 4(2):1351–1356, 2019.

[8] Fangwen Yu, Jianga Shang, Youjian Hu, and Michael Milford. Neuroslam: A brain-inspired slam system for 3d environments. *Biological cybernetics*, 113(5):515–545, 2019.

[9] Michael J Milford and Gordon F Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics*, 24(5):1038–1053, 2008.

[10] Jilles Vreeken et al. Spiking neural networks, an introduction. 2003.

[11] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[12] Leon Chua. Memristor-the missing circuit element. *IEEE Transactions on circuit theory*, 18(5):507–519, 1971.

[13] Xiaoyan Fang, Derong Liu, Shukai Duan, and Lidan Wang. Memristive lif spiking neuron model and its application in morse code. *Frontiers in Neuroscience*, 16:853010, 2022.

[14] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[15] Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 2023.

[16] JB Rank. Head-direction cells in the deep layers of dorsal presubiculum of freely moving rats. In *Soc. Neuroscience Abstr.*, volume 10, page 599, 1984.

[17] Alexei Samsonovich and Bruce L McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience*, 17(15):5900–5920, 1997.

[18] Stanford Artificial Intelligence Laboratory. Robotic operating system.

[19] Dmitri B Strukov, Gregory S Snider, Duncan R Stewart, and R Stanley Williams. The missing memristor found. *nature*, 453(7191):80–83, 2008.