SLAM Bioinspirado con Neuronas Tipo Spiking Memristivas

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 José Fernández León
INTIA, Centro de Investigaciones en Física e Ingeniería del Centro UNICEN – CICpBA – CONICET
B7000JWI, Tandil, Argentina
jafernandez@intia.exa.unicen.edu.ar

5th Gerardo Gabriel Acosta
INTELYMEC, Centro de Investigaciones en Física e Ingeniería del Centro UNICEN – CICpBA – CONICET
B7400JWI, Olavarría, Argentina
ggacosta@fio.unicen.edu.ar

***Resumen*—La incorporación de Inteligencia Artificial (IA) en robótica permitió disponer de capacidades cognitivas que mejoraron sus desempeños en múltiples aplicaciones. A partir de investigaciones de la Neurociencia, se sabe que la información es procesada a nivel neuronal mediante impulsos eléctricos, que luego son transportados a las neuronas vecinas mediante procesos electroquímicos. Este descubrimiento inspiró el desarrollo de las redes neuronales artificiales, y entre ellas, las denominadas Redes Neuronales Spiking (RNS). Con el propósito de implementar RNS a nivel *hardware*, con circuitos electrónicos cuyo comportamiento sea equivalente al de las estructuras neurobiológicas del sistema nervioso, en este trabajo presentamos el modelo *Leaky Integrate-and-Fire*, que es ampliamente utilizado en RNS. La capacidad de memoria necesaria para almacenar pesos sinápticos luego de los entrenamientos se implementó con memristores. Este resistor con memoria es considerado el cuarto elemento pasivo de la electrónica. Hemos seleccionado como caso de estudio para probar estas ideas, un sistema de localización y mapeo simultáneos (SLAM) de inspiración biológica, conocido como Rat-SLAM, para ser evaluado en un robot móvil autónomo (RMA). En efecto, la investigación del cerebro de los roedores, permitió conocer la existencia de zonas encargadas de la localización y orientación espacial, que aprovecharemos en los módulos de guiado y navegación del RMA. Para ésto, en este trabajo proponemos un tipo de RNS, las *Memristive Leaky Integrate-and-Fire* (MLIF) para conformar las *pose cells* del Rat-SLAM.**

***Palabras clave*—Rat-SLAM, Memristores, Robótica Móvil Autónoma, Neurociencias, Computación Neuromórfica.**

I. Introducción

Los robots actuales disponen de capacidades cognitivas que les permiten aprender a resolver con éxito diversas tareas, ampliando por ende, sus capacidades de aplicación. Esta importante cualidad es resultado de la incorporación de la Inteligencia Artificial (IA) a la robótica. La IA, si bien tiene un origen bioinspirado, no procesa la información de manera análoga al sistema nervioso de los seres vivos. Este último conformado por el cerebro, la médula espinal y las redes de células nerviosas sensitivas o motoras, llamadas neuronas, ha sido ampliamente investigado por la Neurociencia. De esta manera se sabe que la información tratada en el sistema nervioso es codificada en el tiempo o frecuencia de impulsos eléctricos. Estos son generados por las neuronas en respuesta al estímulo de entrada que reciben. El resultado del procesamiento de la información recibida se transmite mediante la liberación de neurotransmisores. Estos son "mensajeros químicos" que transportan la información de las neuronas presinápticas a las postsinápticas, propagando o no así la información en toda la red de neuronas.

La forma biológica de comunicación neuronal inspiró el desarrollo de distintos modelos de redes neuronales artificiales. Uno de ellos son las Redes Neuronales Spiking (RNS) [1], cuya forma de procesar la información no se produce en cada ciclo de propagación, sino que transmiten información sólo cuando el potencial de membrana de la neurona alcanza un valor específico llamado umbral. Cuando esto ocurre, la neurona genera un pulso que se transmite a otras produciendo un aumento o disminución de sus potenciales en respuesta a esta señal. Posteriormente, el potencial de membrana de la neurona activada se restablece a su valor predeterminado.

La Computación Neuromórfica (CN) es una forma de realizar cálculos con elementos de procesamiento cuyo soporte *hardware* imita las neuronas biológicas. Esta forma de implementación tiene la ventaja de reducir el esfuerzo de cómputo necesario para resolver ciertos problemas, como en nuestro caso, localización y mapeo simultáneo (SLAM). Uno de los modelos de neuronas más utilizados debido a su similitud con el comportamiento de las biológicas son las de tipo *spiking*. Estas son implementadas con el circuito *Leaky*

Integrate-and-Fire (LIF) [2], formado por una resistencia y un capacitor vinculados entre sí a través de una conexión eléctrica del tipo paralelo, motivo por el cual, se lo conoce como circuito RC paralelo. El nivel de activación es dado por la ecuación diferencial que modela el comportamiento circuital en función de las entradas a la neurona.

Una solución muy adoptada consiste en reemplazar la resistencia del modelo LIF por un memristor [3], dando lugar al modelo *Memristive Leaky Integrate-and-Fire* (MLIF) [4]. El memristor es considerado el cuarto elemento de la electrónica y fue propuesto por León Chua en 1971. Su nombre es una contracción de "memory resistor (resistor con memoria)", por lo que dispone de la capacidad de variar y recordar su último valor de resistencia sin consumo de energía, que lo hacen ideal para aquellas aplicaciones donde el consumo es un punto crítico de implementación.

Para probar este tipo de CN con MLIF, en este trabajo presentamos una implementación de SLAM bioinspirado en el cerebro de los roedores. Los estudios realizados por las Neurociencias revelaron la existencia de ciertas zonas cerebrales conocidas como *células de lugar o place cells* [5], *células de dirección de la cabeza o head direction cells* [6], y *células de grilla o grid cells* [7]. Estas zonas reciben señales sensoriales tales como visión, olfato, entre otras, para localizar al animal en un entorno nunca visitado y recordar la localización de lugares por donde ya ha pasado, con lo que se conoce como mapa de experiencias. Este tipo de ubicación espacial, inspiró el desarrollo de sistemas que permitan resolver el problema de SLAM en la Robótica Móvil Autónoma (RMA). Esta aproximación es conocida como Rat-SLAM [8].

Para que la capacidad de memoria intrínseca de las redes neuronales implementadas tenga además características de mínimo consumo de energía, y posibilidad de implementación en un microchip, en este trabajo proponemos una *Pose Cell Memristiva Spiking* (PCMS), para conformar una red de células nerviosas o red neuronal artificial con circuitos MLIF, que implemente las *pose cell* del Rat-SLAM [8], como se explicará en la sección IV.

II. Modelo de Memristor

El modelo de memristor utilizado es el propuesto por HP Labs [9]. Físicamente es un elemento pasivo de dos terminales formado por dos electrodos de platino que contienen entre sí un material de longitud D formado por dos capas. Una de ellas se compone de dióxido de titanio (TiO_2) altamente resistivo, y la otra de dióxido de titanio pobre en oxígeno (TiO_{2-X}) altamente conductor. Cuando se hace circular una corriente eléctrica en una dirección, el límite entre los dos materiales w se desplaza, produciendo un aumento de la conductividad ocasionado por un mayor porcentaje de la capa TiO_{2-X}. Mientras que, cuando la corriente eléctrica fluye en la dirección contraria a la anterior, se produce un aumento de la resistencia del memristor ocasionada por el desplazamiento de w que aumenta el porcentaje de la capa TiO_2. De lo anterior se deduce que, cuando no existe circulación de corriente, w no cambia y el memristor mantiene su estado, es decir, que "recuerda" el último valor de resistencia adoptado. El comportamiento del memristor antes mencionado es dado por la Ec.(1), donde R_{OFF} es el valor de resistencia máxima que se puede tener cuando $w(t) = 0$, y R_{ON} es el mínimo valor de resistencia correspondiente a la condición $w(t) = D$. Es importante notar que, en el instante inicial $t = 0$, $w(0)$ tendrá un valor entre [0, D], por lo cual el valor del memristor ($M(0)$) se encontrará en el intervalo [R_{OFF}, R_{ON}].

$$M(t) = R_{OFF} + (R_{ON} - R_{OFF})\frac{w(t)}{D} \tag{1}$$

La velocidad de cambio de w, considerando un modelo de deriva lineal con movilidad media de las deficiencias de oxígeno μ_v, es dado por la Ec.(2). Por lo cual, integrando con respecto al tiempo, se obtiene el valor de la ubicación del límite entre las dos zonas del memristor w, como se muestra en la Ec.(3). De esta manera, sustituyendo la Ec.(3) en la Ec.(1), se llega a la Ec.(4), donde $k = (R_{ON} - R_{OFF})\left(\frac{\mu_v R_{ON}}{D^2}\right)$.

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{\mu_v R_{ON}}{D} i(t) \tag{2}$$

$$w(t) = \frac{\mu_v R_{ON}}{D} q(t) + w(0) \tag{3}$$

$$M(t) = M(0) + kq(t) \tag{4}$$

Despejando la carga eléctrica $q(t)$ de la Ec.(4) y sabiendo que los valores límites que puede adoptar del memristor son R_{ON} y R_{OFF}, es posible observar la relación existente entre estas dos magnitudes, como se muestra en la Ec.(5). Luego, teniendo en cuenta que la variable k siempre es menor a cero, y que el valor del memristor es una función monótona decreciente de la carga eléctrica, se puede decir que el modelo de memristor controlado por la carga eléctrica es un función continua que se calcula con la Ec.(6).

$$\underbrace{\frac{R_{OFF} - M(0)}{k}}_{c_1} \leq q(t) \leq \underbrace{\frac{R_{ON} - M(0)}{k}}_{c_2} \tag{5}$$

$$M(t) = \begin{cases} R_{OFF} & \text{si} \quad q(t) < c_1 \\ M(0) + kq(t) & \text{si} \quad c_1 \leq q(t) < c_2 \\ R_{ON} & \text{si} \quad q(t) \geq c_2 \end{cases} \tag{6}$$

Por otra parte, es posible relacionar el flujo con la carga eléctrica mediante la Ec.(7). Con lo cual se obtiene una expresión que vincule flujo con carga eléctrica, tal como se muestra en la Ec.(8). Despejando el flujo de esta última expresión, obtenemos la Ec.(9), que sustituyéndola en la Ec.(6), permite obtener las ecuaciones que rigen el comportamiento del modelo de memristor controlado por el flujo, tal como se muestra en la Ec.(10), donde los valores de las constantes c_3 a c_6 son dados por las Ec.(11).

$$\varphi(t) = \int_0^t M(t) i(t) \mathrm{d}t = \int_{q(0)}^{q(t)} M(t) \mathrm{d}q(t) \tag{7}$$

$$\varphi(t) = \begin{cases} R_{OFF}q(t) + c_3 & \text{si} \quad q(t) < c_1 \\ \frac{k}{2}q^2(t) + M(0)q(t) & \text{si} \quad c_1 \leq q(t) < c_2 \\ R_{ON}q(t) + c_4 & \text{si} \quad q(t) \geq c_2 \end{cases} \tag{8}$$

$$\varphi(t) = \begin{cases} \frac{\varphi(t)-c_3}{R_{OFF}} & \text{si} \quad \varphi(t) < c_5 \\ \frac{\sqrt{2k\varphi(t)+M^2(0)}-M(0)}{k} & \text{si} \quad c_5 \leq \varphi(t) < c_6 \\ \frac{\varphi(t)-c_4}{R_{ON}} & \text{si} \quad \varphi(t) \geq c_6 \end{cases} \tag{9}$$

$$M(t) = \begin{cases} R_{OFF} & \text{si} \quad \varphi(t) < c_5 \\ \sqrt{2k\varphi(t) + M^2(0)} & \text{si} \quad c_5 \leq \varphi(t) < c_6 \\ R_{ON} & \text{si} \quad \varphi(t) \geq c_6 \end{cases} \tag{10}$$

$$\begin{aligned} c_3 &= -\frac{[R_{OFF}-M(0)]^2}{2k} \\ c_4 &= -\frac{[R_{ON}-M(0)]^2}{2k} \\ c_5 &= \frac{R_{OFF}^2-M^2(0)}{2k} \\ c_6 &= \frac{R_{ON}^2-M^2(0)}{2k} \end{aligned} \tag{11}$$

En este trabajo se utilizó el modelo del memristor controlado por flujo con los valores paramétricos propuestos en [4]. De esta manera, los valores adoptados para las variables son $R_{OFF} = 20000\Omega$, $R_{ON} = 100\Omega$, $M(0) = 16000\Omega$, $\mu_v = 1e^{-14}m^2s^{-1}V^{-1}$ y $D = 10nm$.

III. Modelo Neuronal

Una de las primeras representaciones de las neuronas biológicas propuso un elemento de procesamiento (EP) que realiza la suma de sus entradas ponderadas. A esta suma la hace atravesar un función no lineal o función de activación, y si supera cierto umbral predefinido (v_{th}), hace que el EP dispare un impulso, en un típico comportamiento de *on-off*. Luego del disparo, la actividad del EP permanece refractaria a nuevas excitaciones durante un tiempo, y su potencial es reiniciado a un valor establecido (v_{reset}), tal y como sucede con las neuronas estudiadas en la biología por McCulloch y Pitts [10]. Este modelo no tiene en cuenta el tiempo, es decir, no tiene una noción de cuándo ocurren las señales de entrada. La información se almacena en el valor de los pesos intersinápticos, que proporciona una plasticidad enorme para modelar una transferencia de las señales de entrada al impulso de salida. Imaginemos qué pasaría si pudiésemos incorporar información temporal a las señales de entrada a la neurona. La información podría almacenarse en el tiempo o frecuencia en la que suceden los impulsos o *spikes* a lo largo del tiempo. De esta forma, la actividad de la neurona se caracterizaría por su generación de *spikes* en momentos determinados, como una respuesta a las entradas que está recibiendo el EP. Y de esta forma, la comunicación entre neuronas en este modelo se realizaría a través de estos *spikes*, lo que permite una codificación temporal más rica y eficiente en comparación con los modelos basados únicamente en activación binaria. Estas consideraciones tiene en cuenta el modelo de redes neuronales *spiking* (RNS), presentado en [2]. Vale decir, mientras que el modelo de McCulloch-Pitts es una simplificación binaria del comportamiento neuronal, las neuronas de tipo *spiking* son más sofisticadas y tienen en cuenta el tiempo y la naturaleza de las señales neuronales, lo que las hace más adecuadas para modelar el funcionamiento real de las redes neuronales biológicas. Por lo tanto, este último modelo proporcionó una gran comprensión sobre la codificación de la información en las redes neuronales, la memoria, su comportamiento dinámico y, más recientemente, el aprendizaje profundo. Las RNS pueden implementarse en *hardware* mediante un circuito RC, en el que se reemplaza al resistor por un memristor, conocido como el modelo de neurona MLIF, [4]. En la Fig.1, se muestra el circuito de la neurona, donde I_{ext} es la corriente de excitación que atraviesa la membrana de la neurona desde su interior hasta su exterior, C es un capacitor de valor igual a $9e^{-7}$ Faradios, M es un memristor con las características explicadas en la sección anterior, y v_{th} es la tensión de umbral ajustada a 20mV. Cuando el valor de la tensión en el memristor supera a este umbral de 20mV, el interruptor se cierra reiniciando el potencial de la neurona a 0V, y luego vuelve a abrirse hasta que la condición anterior vuelva a cumplirse.

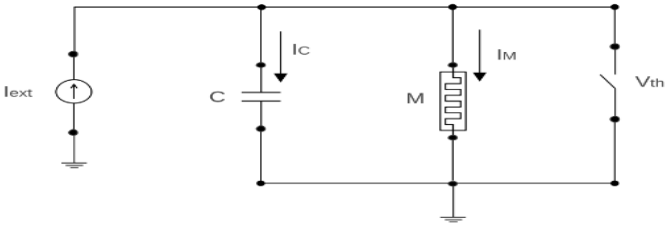

Fig. 1. Modelo de neurona MLIF utilizado.

La ecuación diferencial que modela el comportamiento de este tipo de neurona se obtiene de la siguiente manera. Sabiendo que la corriente externa es igual a la suma de la corriente que circula a través del capacitor y la que circula a través del memristor, y a su vez, sabiendo que la primera obtenida de multiplicar el valor del capacitor por la derivada de la tensión respecto del tiempo, y la segunda de dividir la tensión aplicada por el valor del memristor, se obtiene la expresión mostrada en la Ec.(12). Luego, la ecuación diferencial que rige el comportamiento de la neurona se obtiene despejando la derivada de la tensión respecto del tiempo, como se muestra en la Ec.(13), donde $\tau = M(t)C$. Finalmente, para resolver esta ecuación diferencial se utilizó el método de las diferencias finitas, como se muestra en la Ec.(14), donde Δt es el período de muestreo, cuyo valor adoptado en este trabajo es $1e^{-3}$ s.

$$I_{ext} = C\frac{\mathrm{d}V(t)}{\mathrm{d}t} + \frac{V(t)}{M(t)} \tag{12}$$

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = \frac{1}{\tau}(-V(t) + I_{ext}M(t)) \tag{13}$$

$$V(t+\Delta t) = \frac{\Delta t}{\tau}(-V(t) + I_{ext}M(t)) + V(t) \tag{14}$$

IV. RAT-SLAM CON PCMS

El algoritmo de Rat-SLAM se programó en base a las PCMS. Los aportes de nuestro trabajo son los siguientes:

- Resolver un problema bien definido de la RMA con un sistema inspirado en el comportamiento del cerebro de los roedores utilizando un modelo de neurona de impulsos.
- Codificar las poses del robot en las frecuencias o tiempos de estos impulsos o *spikes*.
- Posibilidad de implementar las *pose cell* en un dispositivo de *hardware*, disminuyendo los requerimientos de esfuerzo de cómputo requerido para la operación del sistema.
- Disponer de una *pose cell* con memoria intrínseca, que aprenda a relacionar un nivel de excitación a cada unidad con una pose del robot.

En el diagrama de bloques de la Fig.2 se muestra esquemáticamente el funcionamiento del Rat-SLAM implementado. Para poder operar, el módulo de percepción del entorno debe proporcionar imágenes monoculares en escala de grises, velocidad lineal, velocidad angular y rumbo. Esta información proviene de una cámara monocular y un odómetro (*encoder*), montados sobre el RMA terrestre, que fue la plataforma de pruebas experimentales de este trabajo. Las *Local View Cell* reciben la imagen y el centroide de actividad de la PCMS para establecer una relación entre ambas. Este módulo está formado por celdas que almacenan una plantilla visual para cada uno de los lugares visitados. Cuando se ve una escena familiar, la celda correspondiente a esa vista se activa, caso contrario una nueva celda es creada. Estas plantillas visuales se obtienen utilizando el perfil de intensidad de la imagen en escala de grises. Cuando la diferencia entre los perfiles almacenados y el actual es menor que un umbral, la plantilla actual no se almacena, caso contrario, se crea una nueva. La comparación entre el perfil actual I^j y los almacenados I^k se realiza utilizando la función de diferencia de intensidad absoluta promedio como se muestra en la Ec.(15), donde f es dada por la Ec.(16) en la que ω es el ancho de la imagen. Esta comparación se lleva a cabo en un pequeño rango de desplazamientos de píxeles s para proporcionar cierta robustez frente a la rotación.

$$k_m = \arg\min_k f\left(s, I^j, I^k\right) \tag{15}$$

$$f(s, I^j, I^k) = \frac{1}{\omega - |s|}\left(\sum_{n=1}^{\omega-|s|} |I^j_{n+\max(s,0)} - I^k_{n-\min(s,0)}|\right) \tag{16}$$

La PCMS es un arreglo 3D de *grid cells*, cuya dinámica es la de Redes de Atractores Continuos (RAC) [11]. Tiene una forma prismática toroidal de tres dimensiones. Dos dimensiones corresponden a la posición en x e y, y la tercera dimensión al rumbo o *heading* del robot. Cada una de las neuronas componentes de estas redes se modelan utilizando el modelo explicado en la sección III, donde el vínculo mediante conexiones excitadoras e inhibidoras es

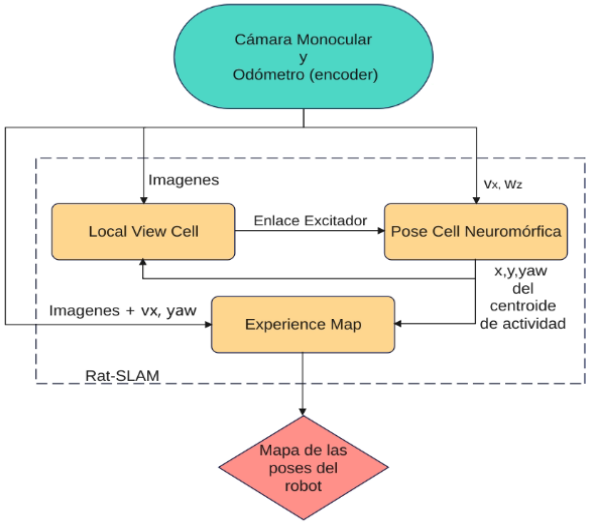

Fig. 2. Diagrama de bloques y flujo de información del Rat-SLAM.

dado por la corriente de excitación que ingresa a cada una, permitiendo que en cada posible estado, un único grupo de unidades se encuentren activas. De esta manera, se establece el denominado paquete de actividad o paquete de energía, cuyo centroide codifica la estimación de la pose actual del robot. La actividad en la PCMS es gobernada por las tres etapas de la dinámica de RAC 3D. Estas son: excitación local, inhibición local-global y normalización de la actividad. La primera de ellas, necesita una matriz de pesos de excitación modelados por distribuciones Gaussianas tridimensionales, como se muestra en la Ec.(17). En esta expresión, var_{eje} es la varianza de cada eje coordenado, constantes para la distribución espacial. El cambio en la corriente de excitación es dado por la Ec.(18), donde $n_{\hat{x}}$, $n_{\hat{y}}$ y $n_{\hat{\theta}}$ son las dimensiones de la matriz de actividad, y $\varepsilon_{a,b,c}$ es la multiplicación de la distribución Gaussiana de cada eje.

$$\varepsilon_j = \left(\frac{1}{var_{eje}\sqrt{2\pi}}\right) e^{-j^2/2var^2_{eje}} \tag{17}$$

$$\Delta i^{(exc)} = \sum_i^{n_{\hat{x}}} \sum_j^{n_{\hat{y}}} \sum_k^{n_{\hat{\theta}}} P^{(t)}_{\hat{x},\hat{y},\hat{\theta}} \cdot \varepsilon_{a,b,c} \tag{18}$$

Los índices a, b y c, representan las distancias entre las coordenadas $\hat{x}$, $\hat{y}$ y $\hat{\theta}$, y se calculan con las expresiones mostradas en Ec.(19).

$$\begin{aligned} a &= (\hat{x} - i) \cdot mod(dim(\hat{x})) \\ b &= (\hat{y} - j) \cdot mod(dim(\hat{y})) \\ c &= (\hat{\theta} - k) \cdot mod(dim(\hat{\theta})) \end{aligned} \tag{19}$$

Obtenida la corriente de excitación, se calcula la corriente de inhibición mediante una matriz de pesos inhibitorios $\psi_{a,b,c}$ modelados de la misma forma que $\varepsilon_{a,b,c}$, pero con valores de pesos negativos, tal como se muestra en la Ec.(20).

$$\Delta i^{(inh)} = \sum_{i}^{n_{\hat{x}}} \sum_{j}^{n_{\hat{y}}} \sum_{k}^{n_{\hat{\theta}}} P^{(t)}_{\hat{x},\hat{y},\hat{\theta}} \cdot \psi_{a,b,c} \tag{20}$$

Con estos dos procesos completados, la corriente que se aplicará a cada neurona de la PCMS en el siguiente tiempo de muestreo es dada por la Ec.(21). Como consecuencia de aplicar una corriente a cada neurona, se obtiene el potencial de la membrana celular de cada una de ellas. Cada potencial representa la matriz de actividad de la PCMS. A esta se le debe restar un valor constante para realizar la inhibición global, y finalmente se la normaliza.

$$i^{(t+1)} = \Delta i^{(exc)} + \Delta i^{(inh)} \tag{21}$$

Hasta aquí, el proceso de actualización de la actividad en la PCMS ha completado los pasos de la dinámica del atractor 3D, sin tener en cuenta la percepción del entorno. Para incorporar esta información a la actualización se realizan dos pasos. El primero es *path integration*, y el segundo, *local view calibration*. El *path integration* proyecta la actividad de las celdas 3D a las cercanas desplazando el paquete de actividad en el plano ($\hat{x}$,$\hat{y}$) de acuerdo con la velocidad de translación lineal, y también se moverá en la en función de la velocidad angular para representar el rumbo ($\hat{\theta}$). En la Ec.(22) se muestra el cambio en la matriz de actividad debido a la percepción del entorno, donde $\delta_{\hat{x}_0}$, $\delta_{\hat{y}_0}$ y $\delta_{\hat{\theta}_0}$, se calculan de acuerdo a la Ec.(23), con $k_{\hat{x}}$, $k_{\hat{y}}$, y $k_{\hat{\theta}}$ representado los offset iniciales para cada eje, y γ se calcula de acuerdo con la Ec.(24), donde $\delta_{\hat{x}_f}$, $\delta_{\hat{y}_f}$ y $\delta_{\hat{\theta_f}}$ son los offset finales dados por la Ec.(25).

$$P^{(t+1)}_{l,m,n} = \sum_{x=\delta_{\hat{x}_0}}^{\delta_{\hat{x}_0}+1} \sum_{y=\delta_{\hat{y}_0}}^{\delta_{\hat{y}_0}+1} \sum_{\hat{\theta}=\delta_{\hat{\theta}_0}}^{\delta_{\hat{\theta}_0}+1} \gamma \cdot P^{(t+1)}_{(l+x),(m+y),(n+\theta)} \tag{22}$$

$$\begin{aligned} \delta_{\hat{x}_0} &= k_{\hat{x}} \cdot v \cdot cos(\hat{\theta}) \\ \delta_{\hat{y}_0} &= k_{\hat{y}} \cdot v \cdot sin(\hat{\theta}) \\ \delta_{\hat{\theta}_0} &= k_{\hat{\theta}} \cdot \omega \end{aligned} \tag{23}$$

$$\gamma = f(\delta_{\hat{x}_f}, \hat{x} - \delta_{\hat{x}_0}) f(\delta_{\hat{y}_f}, \hat{y} - \delta_{\hat{y}_0}) f(\delta_{\hat{\theta}_f}, \hat{\theta} - \delta_{\hat{\theta}_0}) \tag{24}$$

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

$$\begin{aligned} \delta_{\hat{x}_f} &= k_{\hat{x}} \cdot v \cdot cos(\hat{\theta}) - \delta_{\hat{x}_0} \\ \delta_{\hat{y}_f} &= k_{\hat{y}} \cdot v \cdot sin(\hat{\theta}) - \delta_{\hat{y}_0} \\ \delta_{\hat{\theta}_f} &= k_{\hat{\theta}} \cdot \omega - \delta_{\hat{\theta}_0} \end{aligned} \tag{25}$$

Por otra parte, la calibración de vista local o *local view calibration* es un proceso cuyo objetivo es restablecer los errores acumulativos de la integración del camino. Para ello, las *local view cell* están asociadas con la PCMS de modo que cuando el robot se posiciona en un lugar ya visitado, la *local view cell* asociada a esta ubicación activa, por medio de un enlace de excitación, las mismas PCMS que se activaron cuando esta ubicación se visitó por primera vez. Para aprender la relación entre el vector de las *local view cell* y la actividad en la PCMS, la matriz ψ almacena sus interconexiones sinápticas, utilizando una versión modificada de la ley de Hebb para el aprendizaje. La conexión entre $P_{\hat{x},\hat{y},\hat{\theta}}$ y V_i se obtiene mediante la Ec.(26), donde ξ es el learning rate y V_i es el vector de actividad de las *local view cell*.

$$\psi^{t+1}_{i,\hat{x},\hat{y},\hat{\theta}} = max(\psi^{t}_{i,\hat{x},\hat{y},\hat{\theta}}, \xi V_i P_{\hat{x},\hat{y},\hat{\theta}}) \tag{26}$$

El mapa de experiencias o *experience map* es un mapa topológico que se compone de muchas experiencias individuales, e, cada una de ellas conectadas por transiciones, t. Una experiencia individual e_i, dada por la Ec.(27), contiene la posición de las celdas activas en la PCMS ($P^{i}_{\hat{x},\hat{y},\hat{\theta}}$), el vector de actividad de la *local view cell* (V^i), la posición espacial ($\mathbf{p}^i$) estimada a partir de las medidas de odometría. La primera experiencia se crea en un punto de inicio arbitrario, y las experiencias subsiguientes se construyen a partir de la primera a través de transiciones.

$$e_i = \{P^{i}_{\hat{x},\hat{y},\hat{\theta}}, V^i, \mathbf{p}^i\} \tag{27}$$

Cuando la métrica de comparación de la actividad en los dos módulos anteriores supera un umbral S_{max}, se crea una nueva experiencia, con una transición asociada. La métrica S se obtiene con la Ec.(28), donde μ_p y μ_v son constantes que ponderan las respectivas contribuciones de la PCMS y la *local view cell* a la puntuación.

$$S = \mu_p |P^{i}_{\hat{x},\hat{y},\hat{\theta}} - P_{\hat{x},\hat{y},\hat{\theta}}| + \mu_v |V^i - V| \tag{28}$$

La transición t_{ij} almacena el cambio de posición medido a partir de la integración de las velocidades proveniente de la odometría. Como la velocidad de traslación debe estar descompuesta en el plano, es necesario el rumbo brindado por la odometría para llevar a cabo esta operación. En la Ec.(29) se muestra las transiciones almacenadas, donde $\Delta\mathbf{p}^{ij}$ es el cambio en la pose del vehículo estimada. El vínculo entre la experiencia previa e_i y la nueva experiencia e_j se forma según la Ec.(30).

$$t_{ij} = \{\Delta\mathbf{p}^{ij}\} \tag{29}$$

$$e_i = \{P^{j}_{\hat{x},\hat{y},\hat{\theta}}, V^j, \mathbf{p}^i + \Delta\mathbf{p}^{ij}\} \tag{30}$$

En este módulo, el cierre de bucle ocurre cuando la actividad en la PCMS y *local view cell* coinciden lo suficiente con una experiencia almacenada. Cuando esto ocurre, es muy poco probable que el cambio sumado en la posición de las transiciones conduzca a la experiencia del cierre de bucle a coincidir con la almacenada previamente para misma posición. Para lograr esta coincidencia, las posiciones de todas las experiencias se actualizan usando la Ec.(31), donde α es una de tasa de corrección constante, N_f es el número de enlaces desde la experiencia e_i hacia otras experiencias, y

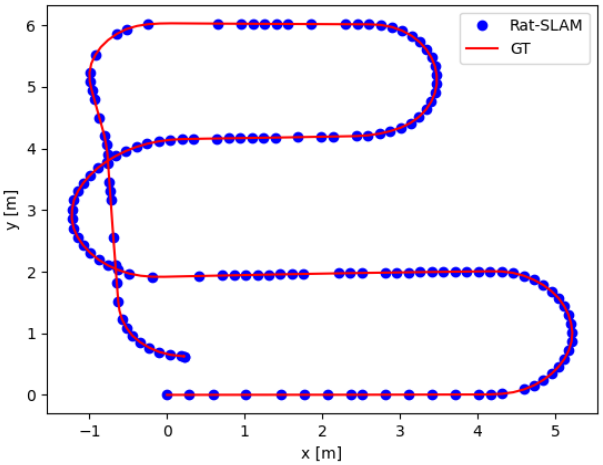

Fig. 3. Trayectoria de prueba con forma de zig-zag.

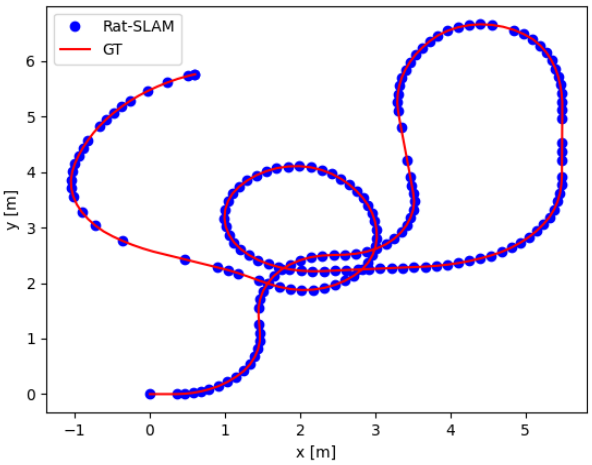

Fig. 4. Trayectoria de prueba con forma compleja.

N_t es el número de enlaces desde otras experiencias hacia la experiencia e_i.

$$\Delta \mathbf{p}^i = \alpha \left[\sum_{j=1}^{N_f} \left(\mathbf{p}^j - \mathbf{p}^i - \Delta \mathbf{p}^{ij} \right) + \sum_{k=1}^{N_t} \left(\mathbf{p}^k - \mathbf{p}^i - \Delta \mathbf{p}^{ki} \right) \right] \tag{31}$$

V. Resultados Experimentales

En esta sección presentamos los resultados obtenidos para trayectorias de prueba realizadas en simulación por computadora. En este trabajo se utilizó Robot Operating System (ROS) [12], ya que dispone de un simulador incorporado llamado Gazebo. El robot que se utilizó es el TurtleBot3 modelo *waffle_pi*, disponible en https://wiki.ros.org/turtlebot3.

Las trayectorias de prueba que se ensayaron son algunas de las más utilizadas en las evaluaciones del funcionamiento de aplicaciones robóticas. La primera tiene forma similar a un zig-zag, que es un tipo de trayectoria útil para tareas de prospección y monitorización de zonas, mientras que la segunda, es de mayor complejidad y una combinación de las anteriores. Los resultados de estas evaluaciones se muestran en las Fig. 3 y Fig. 4, cuyo error cuadrático medio (RMSE) igual a 0.14 m y 0.11 m., respectivamente. El error fue calculado con la diferencia entre las predicciones realizadas por nuestra propuesta de Rat-SLAM y los valores medidos por el odómetro.

VI. Conclusiones

De todas estas evaluaciones, se puede concluir que las estimaciones realizadas por el algoritmo propuesto en este trabajo se aproximan cualitativa y cuantitativamente, con un bajo grado de error como se mencionó, con las lecturas obtenidas del odómetro del robot. Estos resultados validan que la PCMS puede almacenar las poses del robot en las frecuencias o tiempos de los impulsos o *spikes*, tal como ocurre en la actividad cerebral de los mamíferos. Además, valida la posibilidad de aplicar CN para abordar el problema de SLAM, permitiendo sustituir la *pose cell* del algoritmo original implementada en *software* por una implementada en *hardware*. Este pasaje a *hardware* acerca la propuesta a los requisitos de tiempo real y potencia de cómputo a bordo de un RMA. Por otra parte, el uso de los memristores permite que el modelo neuronal implementado disponga, por un lado, de la capacidad intrínseca de memoria, gracias a la cual es posible relacionar una corriente de excitación con una pose, y por otro, disponer de mínimo consumo de energía, comparado con otras soluciones de *hardware* e incluso emulaciones digitales. Finalmente, constatamos que aplicar la idea de RNS hace posible codificar las distintas poses del robot en los tiempos o frecuencias de los impulsos o *spikes*.

Referencias

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

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

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

[4] 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.

[5] Marianne Fyhn, Sturla Molden, Menno P Witter, Edvard I Moser, and May-Britt Moser. Spatial representation in the entorhinal cortex. *Science*, 305(5688):1258–1264, 2004.

[6] John O'Keefe and Dulcie H Conway. Hippocampal place units in the freely moving rat: why they fire where they fire. *Experimental brain research*, 31:573–590, 1978.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

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