



HAL
open science

Récepteur Wake-up radio basé sur des réseaux de neurones à impulsions pour la détection de séquence

Guillaume Marthe, Claire Goursaud, Laurent Clavier

► **To cite this version:**

Guillaume Marthe, Claire Goursaud, Laurent Clavier. Récepteur Wake-up radio basé sur des réseaux de neurones à impulsions pour la détection de séquence. CoRes 2023 - 8èmes Rencontres Francophones sur la Conception de protocoles, l'évaluation de performances et l'expérimentation de Réseaux de communication, May 2023, Cargese, France. pp.1-4. hal-04090633

HAL Id: hal-04090633

<https://hal.science/hal-04090633>

Submitted on 5 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Récepteur Wake-up radio basé sur des réseaux de neurones à impulsions pour la détection de séquence.

Guillaume Marthe¹, Claire Goursaud¹ et Laurent Clavier²

¹Univ Lyon, INSA Lyon, Inria, CITI, EA3720 69621 Villeurbanne prenom.nom@insa-lyon.fr

²CNRS, UMR 8520 - IEMN, IMT Nord Europe, Université de Lille 59000 Lille laurent.clavier@imt-nord-europe.fr

La consommation énergétique est un point critique dans le domaine de l'Internet des Objets. La Wake-Up Radio permet de laisser le plus possible les noeuds en mode veille et de les réveiller seulement si besoin. Malheureusement, les wake-up radios actuels utilisent des microcontrôleurs toujours trop énergivores. D'un autre côté, les réseaux de neurones à impulsions (SNN) émergent dans le domaine du traitement de données par exemple, grâce à leur faible consommation. Nous proposons ainsi, dans ce papier, d'utiliser ces réseaux de neurones comme Wake-up Radio. Le rôle du réseau de neurone sera de reconnaître la séquence d'activation du noeud concerné dans un flux de bits, afin de le réveiller. Nous présentons tout d'abord les hypothèses et modèles de neurones, de réseau et de signaux utilisés pour notre étude. Grâce à nos simulations, nous évaluons ensuite l'impact de paramètres sur les performances.

Mots-clés : Réseau de Neurones à Impulsions, Wake-Up Radio, Détection de séquence d'activation

1 Introduction

L'Internet des Objets (IoT) est devenu un sujet de recherche de premier plan grâce à ses usages dans un grand nombre de domaines [1]. Néanmoins, un de ses principaux défauts reste la consommation énergétique des appareils connectés. Le récepteur Wake-up Radio (WuR) est une des solutions permettant d'économiser de l'énergie. Le principe est d'utiliser un circuit frugal en énergie pour surveiller en permanence le canal [2], et réveiller le circuit principal quand nécessaire. Leur consommation, même si elle est réduite ($200\mu W$), reste toujours trop importante [3]. Ainsi dans cette étude, nous proposons un dispositif de WuR à base de SNN permettant de détecter une séquence d'activation. L'originalité de ce travail est de considérer une nouvelle génération de neurones basés sur une technologie standard CMOS 65nm, qui réduisent la consommation énergétique à $100pW$ grâce à l'absence de transistors. Dans ce papier, nous présentons notre système (signal + architecture) basé sur ces neurones.

2 Modèles

Nous considérons un réseau dans lequel la passerelle est en charge de N noeuds voisins. Du côté du noeud récepteur, l'objectif de la WuR est d'évaluer si sa signature est présente dans le flux entrant. Dans ce cas, le récepteur principal est réveillé afin de communiquer avec la passerelle. Le signal reçu est modélisé par :

$$y = \sum_{k=-\infty}^{\infty} \sum_{i=1}^N \delta_{ik} c_i(t - k \cdot T) + n \quad (1)$$

où δ_{ik} est l'indicateur d'activité (égal à 1 si le code i est transmis dans la k^{ieme} période, et nul sinon), c_i le code OOK spécifique à l'utilisateur i , et $n \sim \mathcal{N}(0, \sigma)$ un bruit blanc additif gaussien (AWGN). Nous proposons d'utiliser des neurones à impulsions pour traiter le signal d'entrée. Ils ne transmettent pas l'information de manière synchronisée, mais tirent lorsque leur potentiel de membrane atteint sa valeur de seuil v_{th} après avoir accumulé les stimuli électriques reçus. Après avoir atteint le seuil, le neurone envoie une impulsion et sa tension membranaire v redescend instantanément à sa valeur de repos v_{rest} . S'il n'atteint pas v_{th} , le potentiel de membrane va décroître vers v_{rest} . Le modèle le plus utilisé pour décrire la décroissance du potentiel de membrane est le modèle Leaky Integrate and Fire (LIF) [4] :

$$\tau_m \frac{dv}{dt} = (v_{rest} - v) + R_m I_m \quad (2)$$

où τ_m est la constante de temps de la membrane, R_m la résistance de membrane et I_m le courant arrivant en entrée du neurone. Les synapses sont les liens entre les différents neurones. Elles peuvent exciter (resp. inhiber) un neurone avec des poids positifs (resp. négatifs) qui demeurent constants. Elles pondèrent et propagent les impulsions depuis la sortie d'un neurone pré-synaptique (le neurone juste avant la synapse considérée) jusqu'à l'entrée du neurone post-synaptique (celui juste après la synapse considérée).

3 Proposition

Le signal radio reçu est transformé en un train d'impulsions en suivant une démodulation OOK. Pour un canal bruité, la démodulation peut fournir un flux erroné avec des impulsions manquantes, des impulsions en trop, ou les deux. Ainsi, l'entrée de notre réseau de neurones est modélisée par un train de spike :

$$s^{(f)}(t) = \sum_f \delta(t - t_f) \quad (3)$$

où t_f correspond à un vecteur contenant les différents temps d'envoi d'impulsion, et δ correspond à la distribution de Dirac. Cela représente le flux de 1 envoyés à un moment précis.

Une fois le signal transformé en train d'impulsions, il nous faut regarder en même temps l bits successifs, l étant la longueur du code, et générer une impulsion lorsque le code est reconnu dans le flux reçu (Figure 1a). Notre solution consiste en un réseau composé de trois couches de neurones, inter-connectés par des synapses de manière totalement feed-forward. La première couche (bleu) sert à paralléliser le train d'impulsion en entrée (Eq. 3), pour le fournir à la deuxième couche. La seconde couche (vert) contient $l - 1$ neurones qui renvoient le train d'impulsions avec un délai approprié tel que le neurone i reçoive $s^{(f)}(t - i \cdot t_{bit})$ avec t_{bit} le temps entre deux bits. Pour un motif de l bits, le neurone de la première couche et les $l - 1$ neurones de la deuxième couche vont simultanément envoyer les l derniers bits reçus en entrée à la dernière couche. Enfin, la couche de sortie (rouge) contient autant de neurones que de signatures différentes à reconnaître (m , que l'on fixera à $m = 1$ pour la suite) où chacun sera connecté aux l neurones définis précédemment. Chaque synapse excite ou inhibe les neurones de la troisième couche selon le code visé (Figure 1a). Si le chip attendu est un 1, le poids est positif pour augmenter la tension. Inversement, pour un 0, il est négatif : si le neurone précédent tire, la synapse inhibitrice envoie une impulsion négative au neurone de sortie pour diminuer sa tension membranaire et prévenir l'envoi d'une impulsion à la sortie du système. Le seuil de chaque neurone de sortie est choisi de manière à tirer s'il reçoit en même temps autant d'impulsions excitatrices que l'adresse contient de 1s, et aucune impulsion provenant de synapses inhibitrices.

4 Résultats

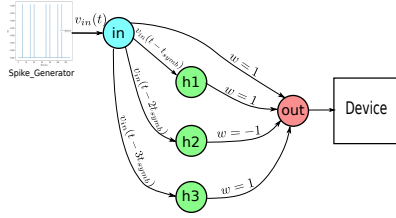
Les simulations ont été réalisées avec Brian2 [5], un simulateur pour SNN basé sur Python. À partir de la taille de la signature l et du nombre de sorties que l'on souhaite reconnaître m , nous créons le réseau comme décrit en Section 2. Notre réseau contient donc $n_{neu} = l + m$ neurones et $n_{syn} = l(m + 1)$ synapses.

4.1 Modèle de transmission

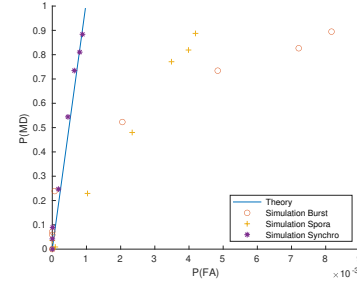
Nous évaluons les performances pour trois types de motif de séquences d'activation. Dans le premier, appelé *Burst*, les séquences sont envoyées de manière continue, reproduisant le cas où il y aurait tout le temps un noeud à réveiller. Le deuxième est appelé *Sporadic* et correspond au cas où les noeuds sont rarement réveillés. Les séquences sont envoyées de manière éparse, avec un temps de silence entre deux motifs consécutifs durant au moins le temps d'un code. Pour le troisième cas, appelé *Synchro*, on suppose que la transmission est slottée et que le récepteur est synchronisé. Le bruit peut créer des erreurs sur la génération du train d'impulsions, ce qui peut entraîner des Fausse Alarmes (FA) et des Détections Manquées (MD). La probabilité de MD est donnée par $\frac{FN}{FN+TP}$, tandis que celle de FA correspond à $\frac{FP}{FP+TN}$; où FN est le nombre de Faux Négatifs, FP celui de Faux Positifs et TP (resp. TN) pour les vrais positifs (resp. négatifs).

4.2 Configuration initiale : Codes libres

Dans un premier temps, on considère que n'importe quelle séquence de l bits peut être utilisée comme code. Les performances obtenues avec σ variant de 0 à 60 pour $l = 10$ sont reportées en Figure 1b. Tout d'abord, les MD semblent être plus importantes que les FA. En effet, les MD se produisent quand il y a au moins une erreur sur la séquence. Pour obtenir une FA, une séquence quasi similaire doit être envoyée et les bits différents doivent être inversés, ce qui rend ce cas moins probable. De plus, les performances sont bien



(a) Parcours du signal dans le réseau.



(b) Impact du bruit sur FA et MD pour un code libre

meilleures dans le cas Synchro que dans les autres scénarios. Avec la méthode Burst, ceci est dû à la fenêtre glissante qui peut reconnaître la séquence dans la juxtaposition de deux séquences. Pour le cas sporadique, ceci est dû au fait que le réseau de neurones ne peut pas faire la différence entre l'absence d'impulsion due à l'absence de transmission, ou à l'envoi d'un 0 dans la séquence OOK transmise. De ce fait, les cas de FA ont plus de chances d'apparaître en Sporadic lorsque le code recherché contient des 0 au début ou à la fin. Dans le cas Synchro, ces FA disparaissent. Néanmoins, la synchronisation est trop consommatrice en énergie. Nous proposons donc de forcer la séquence à commencer et terminer par un 1.

4.3 Code amélioré

On peut voir en Figure 2a qu'ajouter les 1s (violet) permet d'atteindre de meilleures performances à nombre de bits utiles constant (vert), mais de moins bonnes si l'on compare à longueur de signature constante (bleu clair). Cela entraîne une augmentation de la taille de la signature, et donc de la consommation énergétique. La partie intéressante de cette figure est la courbe pour une signature entourée par les 1s (violet). On peut remarquer que pour $\sigma = 30mV$ la probabilité de MD augmente, mais celle de FA commence à diminuer jusqu'à atteindre sa valeur limite (autour de 10^{-3}). Les premier et dernier bits des codes émis et recherchés étant toujours 1, si une erreur est présente sur un de ces bits, il n'est plus possible que l'on ait une FA. Or pour de grandes valeurs de sigma, la probabilité que les bits aux extrémités s'inversent croît plus vite que celle que les autres bits s'inversent pour devenir la signature à reconnaître.

4.4 Analyse du seuil

De manière à protéger les 1s de synchronisation, on propose de modifier le seuil du neurone d'entrée. Pour une valeur quelconque de v_{th} , on obtient deux équations pour FA et MD :

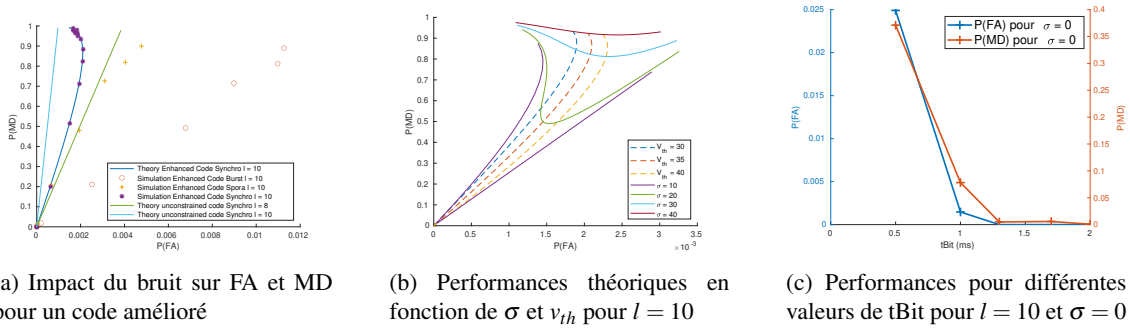
$$P(FA \cap N) = (1 - P(E|1))^{2o} \sum_{k=1}^{l-2o} \left(\frac{1}{2}\right)^{l-2o} \binom{l-2o}{k} \times \sum_{m=0}^k \left(\left(\frac{1}{2}\right)^k \binom{k}{m}\right) P(E|0)^m P(E|1)^{k-m} \quad (4)$$

$$\times \sum_{n=0}^{l-2o-k} \left(\frac{1}{2}\right)^{l-2o-k} \binom{l-2o-k}{n} \times (1 - P(E|0))^n (1 - P(E|1))^{l-2o-k-n}$$

$$P(MD) = 1 - \frac{1}{2^{l-2o}} \sum_{k=1}^{l-2o} \binom{l-2o}{k} (1 - P(E|0))^k (1 - P(E|1))^{l-k} \quad (5)$$

$$P(E|0) = P(n > (v_{th} - v_{rest})) \quad \text{et} \quad P(E|1) = P(n < (v_{th} - v_{spike})) \quad (6)$$

La Figure 2b présente les probabilités théoriques de FA et MD dans deux cas. Tout d'abord, pour une valeur spécifique de sigma, en faisant varier v_{th} (traits pleins). Cela permet de définir, pour un groupe précis de paramètres, le seuil qui donne les meilleures performances. On peut observer que chacune de ces courbes a une valeur optimale, que nous essayons de trouver. Pour cela, nous avons tracé les courbes pour un sigma qui varie tout en fixant la valeur de v_{th} (lignes hachurées). Si notre séquence inclue en moyenne autant de 0 que de 1, on s'attend à ce que le seuil optimal soit placé autour de $v_{th} = -35mV$, soit $30mV$ au dessus du potentiel de repos, avec $v_{spike} = 60mV$ la hauteur d'une impulsion. Comme les bits aux extrémités sont fixés à 1, il y a en moyenne plus de 1s que de 0s dans la signature. Le seuil optimal est donc décalé autour de $v_{th} = -30mV$ pour $l = 10$. Ce phénomène disparaît progressivement quand on augmente la taille de la signature du fait que le nombre moyen de 1 tend vers $\frac{l}{2}$.



4.5 Impact de la contrainte temporelle

Pour finir, nous étudions une contrainte matérielle des neurones. Leur comportement dépend de leur propre constante de temps. En particulier, la décroissance de la tension n'est pas instantanée. La tension peut ainsi ne pas avoir le temps de retourner à sa valeur de repos, ce qui favorise une alarme pour la séquence suivante. Nous évaluons donc les performances tout en faisant varier la durée d'un bit, $tBit$. Cette étude a été réalisée en mode Synchro avec une signature entourée par les 1. On peut voir sur la Figure 2c que laisser plus de temps aux neurones entre la réception de deux bits consécutifs améliore la précision en terme de FA et de MD, car il y a ainsi assez de temps pour que son potentiel décroisse jusqu'à sa tension de repos. On peut également remarquer que pour notre modèle et notre jeu de paramètres, il n'y a plus d'impact sur la précision à partir de $1.25ms$. Le délai entre deux symboles OOK est donc une contrainte et la vitesse d'envoi du code a une limite haute. Cependant, nous pouvons adapter τ_m , la hauteur des impulsions, ou les caractéristiques internes du neurone pour tendre plus rapidement vers v_{rest} et outrepasser cette limitation.

5 Conclusion

Dans cet article, nous avons proposé une architecture basée sur des neurones à impulsions pour la reconnaissance de motif, ainsi que la définition de la séquence appropriée. Nous avons montré qu'ajouter les 1s à cette séquence nous permettait de nous rapprocher de la précision obtenue en mode Synchro, sans circuit de synchronisation. Cela implique d'adapter le seuil en fonction de la séquence d'activation de manière à améliorer à la fois FA et MD. Nous avons également observé que nous ne pouvons pas recevoir nos bits trop rapidement à cause de la dynamique du neurone, ce qui nous place face à une limite de la vitesse du flux entrant si l'on veut une détection optimale. Cependant, cette limitation n'en est pas réellement une car on veut seulement réveiller un noeud, et non transmettre des données. Côté consommation énergétique notre schéma contient 11 neurones or la puissance des architectures neuromorphiques actuelles se situe autour de $100pW$ par neurone [6]. La puissance du réseau serait de l'ordre de $1nW$. Cela reste très inférieur aux pics de puissance de l'ordre de $200\mu W$ des WuR actuels. Même avec une séquence de 100 bits, la puissance serait réduite d'un facteur 20000. Cette approche est donc prometteuse.

Références

- [1] R. Mehta, J. Sahni, and K. Khanna, "Internet of things : Vision, applications and challenges," *Procedia Computer Science*, vol. 132, pp. 1263–1269, 2018.
- [2] I. Demirkol, C. Ersoy, and E. Onur, "Wake-up receivers for wireless sensor networks : benefits and challenges," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 88–96, 2009.
- [3] N. S. Mazloum and O. Edfors, "Performance analysis and energy optimization of wake-up receiver schemes for wireless low-power applications," *IEEE Transactions on Wireless Communications*, vol. 13, no. 12, pp. 7050–7061, 2014.
- [4] E. Hunsberger and C. Eliasmith, "Spiking deep networks with LIF neurons," *CoRR*, vol. abs/1510.08829, 2015.
- [5] M. Stimberg, R. Brette, and D. Goodman, "Brian 2, an intuitive and efficient neural simulator," *eLife*, vol. 8, 08 2019.
- [6] F. Danneville, C. Loyez, K. Carpentier, I. Sourikopoulos, E. Mercier, and A. Cappy, "A sub-35 pw axon-hillock artificial neuron circuit," *Solid-State Electronics*, vol. 153, pp. 88–92, 2019.