International Colloquium on Signal, Automatic control and Telecommunications

10-12 juin 2020
Caen
France

Table des matières

Systèmes SSO basés sur OAuth: Vulnérabilités et attaques

BELFAIK Yousra
Laboratoire LIMATI,
FPBM, Université USMS
Beni Mellal, Maroc
Email: yousrabelfaik123@gmail.com

SADQI Yassine
Laboratoire LIMATI,
FPBM, Université USMS
Beni Mellal, Maroc
Email: y.sadqi@usms.com

SAFI Said
Laboratoire LIMATI,
FPBM, Université USMS
Beni Mellal, Maroc
Email: said.safi@gmail.com

***Abstract*—Dans nos jours, les utilisateurs accèdent quotidiennement à des services web et des applications Mobiles qui requièrent généralement une forme d'authentification pour mener à leurs activités professionnelles et personnelles, souvent à l'aide d'un couple (nom d'utilisateur / mot de passe). La grande prolifération de ces applications augmente également le nombre d'informations d'identification de chaque utilisateur et, par conséquent, la possibilité de les perdre ou de les oublier. L'authentification unique SSO (Single Sign-On) peut être utilisée pour résoudre de nombreux problèmes liés à l'authentification des utilisateurs Web. Le standard OAuth2.0 est l'un des protocoles d'autorisation les plus déployés pour les systèmes (SSO) afin de faciliter la gestion des mots de passe pour les utilisateurs. Les systèmes SSO basés sur OAuth sont largement déployés par de grandes entreprises de technologies telles que Facebook, Google et Microsoft. Dans cet article, nous proposons une analyse approfondie des problèmes de sécurité des systèmes SSO basés sur OAuth. En fait, les efforts antérieurs visaient soit à trouver des erreurs dans des implémentations spécifiques, soit à trouver des problèmes de sécurité dans la spécification elle-même. Dans ce travail, nous allons faire une analyse de sécurité globale combine tous les problèmes sécuritaires liés au OAuth au niveau de spécification et quelques implémentations spécifiques.**

Mots clés: Authentification Unique SSO - OAuth - Sécurité SSO - Web SSO - Authentification d'utilisateurs - Autorisation.

1. INTRODUCATION

OAuth 2.0 est un standard d'autorisation [1] permettant à un utilisateur d'accorder l'accès limité à ses ressources (données ou services) sur une application à une autre, sans avoir exposé ses informations d'identification. Par rapport à son prédécesseur OAuth1.0 et à d'autres protocoles existants tels que OpenID, Google AuthSub, Yahoo BBAuth et Microsoft Live ID, OAuth2.0 met l'accent sur la simplicité des développeurs clients tout en fournissant des flux d'autorisation spécifiques pour les sites Web et les applications fonctionnant sur le navigateur, mobile, bureau, ou appareils [2].

Depuis la publication de protocole OAuth2.0 à la fin de 2012, de nombreux sites Web à travers le monde l'ont adopté comme moyen de fournir des services d'authentification unique SSO (Single Sign-On). En utilisant OAuth2.0, les sites Web peuvent réduire le fardeau de la gestion des mots de passe pour leurs utilisateurs, ainsi que de sauver les utilisateurs de l'inconvénient de ressaisir les attributs qui sont à la place stockés par les fournisseurs d'identité (IdP) et fournis aux parties de confiance (RP) au besoin. Il existe une infrastructure riche de fournisseurs d'identité fournissant des services d'identité en utilisant OAuth2.0 tels que Google, Facebook, Microsoft, Instagram, etc. En effet, de nombreux sites Web fonctionnants en tant que fournisseurs d'identité SSO offrent également des fonctionnalités permettant aux utilisateurs de se connecter à ces sites web en utilisant d'autres services en tant que fournisseurs d'identité [3]. Les chercheurs Ghasemisharif et al.(2018) ont constaté que 52% des fournisseurs d'identité affichent un double comportement- servent à la fois des parties de confiance et des fournisseurs d'identité pour d'autres services.

Le but de ce travail est de fournir une analyse approfondie de la sécurité d'OAuth. La tâche d'analyse de la sécurité d'OAuth est très difficile, d'une part en raison des différents flux d(autorisation et options que fournit ce protocole, et d'autre part en raison de la complexité inhérente du Web. En outre, la grande utilisation d'OAuth pour les systèmes d'authentification unique SSO, rend la sécurité de ce protocole un défi majeur pour les chercheurs. Les informations protégées par ces systèmes peuvent être attrayantes pour les adversaires, puisque grâce à une exploitation réussite d'une faiblesse découverte dans la spécification ou les implémentations de ce protocole, un adversaire pourrait récolter les données privées des millions utilisateurs des systèmes SSO pour le vol d'identité, le profilage en ligne, le spam email à grande échelle, phishing et les campagnes drive-by-download [4].

La plupart des efforts d'analyse concernant la sécurité d'OAuth visaient soit à trouver des problèmes de sécurité dans des implémentations spécifiques, soit sur la spécification de ce protocole elle-même. Dans cet article, nous présentons une analyse sécuritaire globale combine les problèmes de sécurité d'OAuth au niveau de spécification et d'implémentation. Les principales contributions de cet

article sont les suivantes :

- Décrire en détails les flux d'autorisation OAuth2.0 et résumer les différences entre les flux de chaque scénario qui affectent la sécurité du protocole OAuth2.0.
- Examiner les problèmes de sécurité liés à la spécification OAuth2.0 et à l'implémentation dans l'environnement Web.

2. TRAVAUX CONNEXES

En 2016, les chercheurs Daniel Fett et al. [6] ont effectué la première analyse formelle approfondie de la norme OAuth2.0 dans un modèle Web expressif dans le but de fournir une analyse de sécurité approfondie d'OAuth. Le travail le plus proche de ce travail est celle proposé par Bansal et al. [12] ont analysé la sécurité d'OAuth en utilisant le pi-calcul appliqué et la bibliothèque WebSpi, ainsi que l'outil d'analyse de protocole ProVerif. L'objectif principal de leur travail est de découvrir des attaques contre OAuth, plutôt que de prouver la sécurité. La RFC 6749 [1] et la RFC 6819 [10] présentaient plusieurs considérations de sécurité des spécifications telles que les problèmes d'authentification des clients, la prédiction de jetons, la soumission de jetons sur une communication non sécurisée et les attaques par détournement de clics (clickjacking) et par injection. En 2018, Stefano Calzavara et al. [8] ont proposé un moniteur de sécurité côté navigateur pour les protocoles Web appelé*Web protocole Security Enforcer* (WPSE), et ils l'ont utilisé pour effectuer une évaluation expérimentale approfondie de la sécurité de OAuth 2.0 dans l'environnement Web. En 2020, J. Bradley et al. [11] ont décrit les meilleures pratiques de sécurité actuelles pour OAuth 2.0, ainsi les nouvelles menaces pertinentes de ce protocole. En 2019, Wanpeng Li et al. [13] ont développé un scanner de vulnérabilité et protecteur OAuth 2.0 et OpenID Connect appelé OAuthGuard, qui fonctionne avec les RPs utilisant les services de Google OAuth 2.0 et OpenID Connect. OAuthguard protège la sécurité et la vie privée des utilisateurs même lorsque les RPs ne mettent pas en œuvre Oauth 2.0 ou Openid Connect correctement. En 2014, Y. Zhou et al. [14] ont décrit la conception et l'implémentation de SSOScan, un vérificateur de vulnérabilité automatique pour les applications utilisant les APIs Facebook Single Sign-On(SSO), et ils l'ont utilisé pour étudier les vingt mille sites Web les mieux classés pour cinq vulnérabilités SSO (*Access token misuse*,*Signed request misuse*, *App secret leak* et *User OAuth credentials leak*). Sun S-T et Beznosov K. [4] ont examinés les implémentations de trois principaux fournisseurs d'identité OAuth (IdP) (Facebook, Microsoft et Google) et de 96 sites Web RP populaires qui prennent en charge l'utilisation de comptes Facebook pour la connexion, et ils sont arrivés à trouver plusieurs vulnérabilités critiques qui permettent à un attaquant d'obtenir un accès non autorisé au profil et au graphique social de l'utilisateur victime, et d'usurper l'identité de ce dernier sur le site Web (RP).

3. CONTEXTE

Dans cette section, nous définissons les différents rôles de protocole d'autorisation OAuth, son flux abstrait, ainsi que les paramètres nécessaires pour faire l'enregistrement d'un client OAuth. En outre, nous fournissons une description des différents flux (appelés aussi modes) d'autorisation d'OAuth ainsi que les cas d'utilisation de chaque flux. Et finalement, nous allons présenter les différents paramètres de sécurité d'OAuth pour atténuer les attaques et les problèmes de sécurité d'OAuth.

3.1. Rôles et flux abstrait du protocole OAuth

La spécification OAuth2.0 [RFC 6749] décrit un système qui permet à une application d'accéder à des ressources (généralement des informations personnelles) protégées par un serveur de ressources au nom du propriétaire de la ressource, par la consommation d'un jeton d'accès émis par un serveur d'autorisation. À l'appui de ce système, l'architecture OAuth 2.0 comporte les quatre rôles suivants:

- **Propriétaire de la ressource (Resource Owner):** Entité capable d'autoriser l'accès à une ressource protégée. Lorsque le propriétaire de la ressource est une personne, on parle d'utilisateur final (End-user).
- **Client:** Le client désigne l'application tierce qui demande l'accès à la ressource au nom de son propriétaire. Il peut être une application web, une application mobile, une application JavaScript, etc. (Le client est lui-même le RP lorsque OAuth 2.0 est utilisé pour SSO).
- **Serveur de ressource (Resource server):** Le serveur de ressource désigne le serveur qui héberge les ressources protégées.
- **Serveur d'autorisation (Authorization server):** Le serveur d'autorisation est le serveur qui délivre les jetons (tokens) au client. Ces tokens seront utilisés lors des requêtes du client vers le serveur de ressources. Le serveur d(autorisation peut être le même que le serveur de ressources (physiquement et applicativement), et c'est souvent le cas. Lorsqu'OAuth est utilisé pour SSO, cette entité et le serveur de ressource constituent conjointement l'IdP.

Le flux abstrait de protocole OAuth2.0 illustré à la figure 1, décrit l'interaction entre les quatre rôles et comprend les étapes suivantes :

(1) Le client demande une autorisation au propriétaire de la ressource. La demande d'autorisation peut être adressée directement au propriétaire de la ressource (comme indiqué dans la figure 1), ou de préférence indirectement via le serveur d'autorisation en tant qu'intermédiaire.

(2) Le client reçoit une Grant d'autorisation, qui représente l'autorisation du propriétaire de la ressource et qui l'utilise pour obtenir un jeton d'accès.

(3) Le client demande un jeton d'accès en s'authentifiant auprès du serveur d'autorisations puis en présentant le gant d'autorisation.

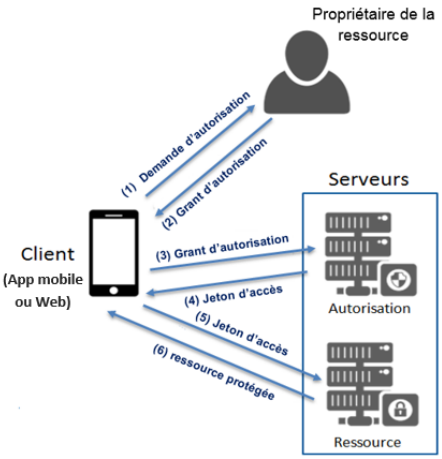

Figure 1. Flux abstrait de protocole OAuth2.0

(4) Le serveur d'autorisation authentifie le client et valide le grant d'autorisation, si il est valide, il émet un jeton d'accès.

(5) Le client demande la ressource protégée au serveur de ressources et s'authentifie en présentant le jeton d'accès.

(6) Le serveur de ressources valide le jeton d'accès et, s'il est valide, sert la demande.

3.2. Enregistrement de client OAuth2.0

Lorsqu'on veut accéder aux données d'un serveur de ressource utilisant OAuth2.0, le serveur d'autorisation doit connaitre les informations sur chaque client avant de pouvoir délivrer un token au client [9]. Le protocole OAuth2.0 définit les paramètres qui doivent être renseignés par le client lors de l'enregistrement et ceux qui doivent être retournés par le serveur d'autorisation. L'enregistrement de client nécessite principalement trois informations :

- **Un identifiant de client (client_id) :** est une chaîne de caractères unique générée d'une manière pseudo aléatoire. cet identifiant est utilisé par le serveur d'autorisation pour enregistrer le client.
- **Une clé secrète (client_secret):** est un secret généré lors de l'enregistrement du client auprès du serveur d'autorisations. Le RP peut utiliser le client_secret (s'il est émis) pour s'authentifier auprès de l'IdP [8].
- **Des URIs de redirection (redirect_uri):** le client doit indiquer au serveur d'autorisation les URIs où il est possible d'envoyer des utilisateurs une fois que le serveur d'autorisation a terminé son interaction. Ceci est important car le serveur d'autorisation ne doit jamais rediriger le navigateur d'une personne vers un URI qui n'a pas encore été enregistré. Si l'URI de redirection est une adresse Web, il doit toujours utiliser le schéma HTTPS pour protéger les données sensibles qui transitent entre le client et le serveur d'autorisation (token d'accès et éventuellement des identifiants et des mots de passe).

L'enregistrement d'un client OAuth est également un moment approprié pour informer le serveur d'autorisation de ses préférences, tels que les types d'algorithmes cryptographiques préférés, les champs d'application demandés et les mécanismes d'authentification par défaut. Toutes ces informations peuvent être fournies lors de l'enregistrement.

3.3. Flux d'autorisation OAuth2.0

Le grant d'autorisation est une information d'identification représentant l'autorisation du propriétaire de la ressource (pour accéder à ses ressources protégées) utilisée par le client pour obtenir un jeton d'accès. Dans OAuth 2.0, les interactions entre l'utilisateur et son navigateur, le RP, et l'IdP peuvent être effectuées en quatre flux différents, ou types de grant : grant de code d'autorisation, grant implicite, grant d'identifiants de propriétaire de ressource et grant d'identifiants de client.

L'organigramme montré dans la figure 2 représente le cas d'utilisation de chaque flux d'autorisation. En effet, si nous avons une communication machine-à-machine, nous utilisons le flux d'autorisation du type grant d'identifiants de client. Tandis que, dans le cas d'une communication avec un client, le flux d'autorisation diffère selon le type de ce dernier : Si le client est une application Web ou mobile, nous utilisons généralement le flux de code d'autorisation. Alors que dans le cas d'une application JavaScript ou basée sur un navigateur, nous utilisons le flux implicite. Et finalement, nous utilisons le flux de grant d'identifiants de propriétaire de ressource si et seulement si on a une relation de confiance absolue entre le propriétaire de ressource et le client.

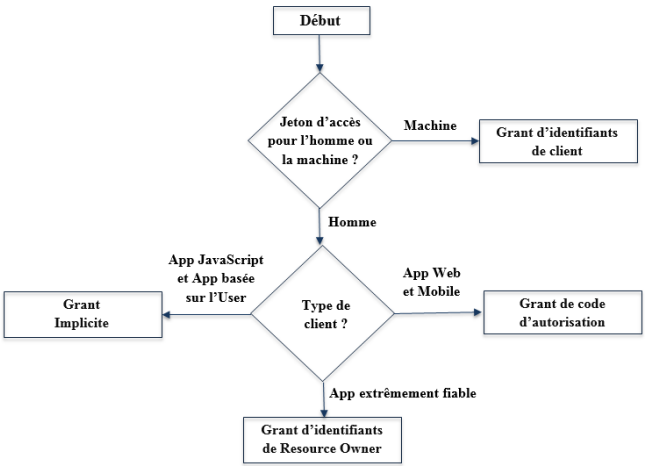

Figure 2. Organigramme de cas d'utilisation des Grants d'autorisations [9]

Nous fournissons ci-dessous une description du flux code d'autorisation avec une brève explication des trois autres flux. Pour la simplicité, nous devons préciser les étapes où ils diffèrent du flux de code d'autorisation.

Flux de code d'autorisation : Lorsque l'utilisateur tente d'autoriser un RP à accéder à ses données sur un IdP ou à se connecter à un RP, le RP redirige d'abord le navigateur (user–agent) de l'utilisateur vers l'IdP. L'utilisateur

s'authentifie ensuite auprès de l'IdP, par exemple en fournissant son nom d'utilisateur et son mot de passe, et est finalement redirigé vers le RP avec un code d'autorisation généré par l'IdP. Le RP peut maintenant communiquer avec l'IdP avec ce code d'autorisation et recevoir un jeton d'accès, que le RP peut à son tour utiliser comme justification pour accéder aux ressources protégées de l'utilisateur à l'IdP. Le diagramme de séquence illustré à la figure 3 présente les étapes du flux de code d'autorisation.

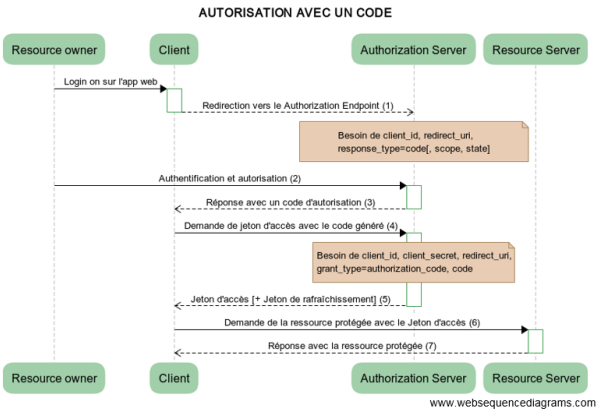

Figure 3. Diagramme de séquence de flux de code d'autorisation

(1) Le client redirige le propriétaire de la ressource (Utilisateur) via son user-agent (Navigateur web) vers l'Authorization Endpoint. Le client doit inclure son identifiant dans la requête de redirection et le niveau d'accès qu'il souhaite obtenir.

(2) Le propriétaire de la ressource s'authentifie auprès du serveur d'autorisation (IdP) et approuve ou non la requête du client.

(3) Si la requête est autorisée, le serveur d'autorisation redirige à nouveau le propriétaire de la ressource vers le client en utilisant l'URI de redirection fourni précédemment (dans la demande ou lors de l'enregistrement du client). L'URI de redirection comprend un code d'autorisation et le paramètre "state" fourni précédemment par le client.

(4) Le client demande un jeton d'accès au Token Endpoint du serveur d'autorisation en incluant le code d'autorisation reçu à l'étape précédente. Lors de la demande, le client s'authentifie auprès du serveur d'autorisation.

(5) Le serveur d'autorisation (IdP) authentifie le client, valide le code d'autorisation et s'assure que l'URI de redirection reçu correspond à l'URI utilisé pour rediriger le client à l'étape (3). S'il est valide, le serveur d'autorisation répond avec un jeton d'accès et, optionnellement, un jeton de rafraîchissement.

(6) Le client demande la ressource protégée au serveur de ressource (IdP) et s'authentifie en présentant le jeton d'accès.

(7) Le serveur de ressource valide le jeton d'accès, et s'il est valide, serve la demande.

Flux implicite : Le flux implicite est similaire au flux de code d'autorisation, mais au lieu d'émettre un code d'autorisation au RP, l'IdP (serveur d'autorisation) délivre directement un jeton d'accès au RP via le navigateur de l'utilisateur (le jeton d'accès est inclus dans le fragment de l'URI de redirection). Lors de l'émission de jeton d'accès, l'IdP n'authentifie pas le RP. Dans certains cas, l'identité de RP peut être vérifiée via l'URI de redirection.

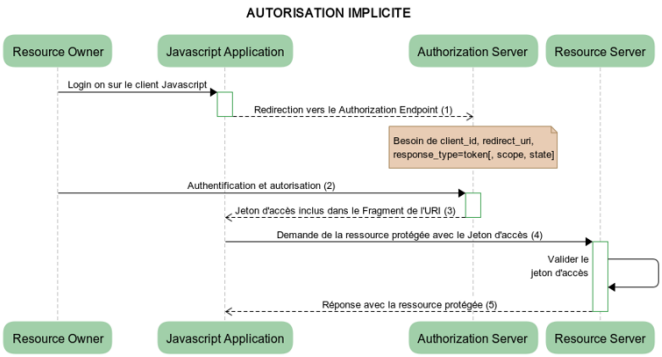

Figure 4. Diagramme de séquence de flux implicite

Flux d'autorisation avec les identifiants du propriétaire de la ressource : Dans ce flux, l'utilisateur (le propriétaire de la ressource) donne ses informations d'identification pour un IdP directement à un RP. Le RP peut ensuite s'authentifier auprès de l'IdP au nom de l'utilisateur et récupérer un jeton d'accès. Ce type d'autorisation est destiné aux RPs hautement fiables (le client et le propriétaire de ressource doivent y avoir une relation de confiance absolue entre eux).

Flux d'autorisation avec les identifiants du client: Ce flux est le plus simple parmi les quatre flux illustrés ci-dessus, car le client (RP) peut obtenir un jeton d'accès en fournissant seulement ses propres informations d'identification au IdP. Ce type d'autorisation est très utile dans les communications machine-à-machine où aucune personne n'est associée à une demande d'API, et il ne doit être utilisé que par des clients confidentiels.

3.4. Paramètres de sécurité OAuth2.0

OAuth2.0 a intégré quelques paramètres pour atténuer les attaques et les problèmes de sécurité des systèmes basés sur ce protocole [7]. En plus des paramètres "client_id", "client_secret" et "redirect_uri" présentés dans la sous-section 3.2, nous définissons trois autres paramètres: code d'autorisation, paramètre "state" et les jetons.

i. Code d'autorisation

Un code d'autorisation représente le résultat intermédiaire d'un processus d'autorisation d'utilisateur réussi et est utilisé par le client pour obtenir des jetons d'accès et de rafraîchissement. Le code d'autorisation est envoyé à l'URI de redirection du client au lieu de jetons, car il est plus simple d'authentifier les clients lors de la requête

directe entre le client et le serveur d'autorisation que dans le cadre de la requête d'autorisation indirecte (nécessite des signatures numériques).Un code d'autorisation est généré par un serveur d'autorisation, après une approbation valide de l'utilisateur final.

ii. Paramètre "state"

Le paramètre de sécurité "state" est utilisé par le client pour maintenir l'état entre la demande et le rappel (callback). Si le client envoie le paramètre "state" au serveur d'autorisation, le serveur d'autorisation gardera cette valeur inchangée et la renverra lors de la redirection de user-agent vers le client. Ce paramètre est utilisé pour empêcher les attaques CSRF.

iii. Jetons

La demande d'accès à une ressource protégée via OAuth se traduit par la délivrance d'un jeton au client. Le jeton représente une chaîne de caractère unique permettant d'identifier le client et les différentes informations utiles durant le processus d'autorisation. Les jetons peuvent être utilisés de deux façons pour invoquer les requêtes sur les serveurs de ressources :

- **Jeton porteur (Bearer Token) :** est un jeton de sécurité avec la propriété que toute partie en possession du jeton peut l'utiliser (la simple possession suffit pour utiliser le jeton). La communication entre les endpoints doit être sécurisée pour garantir que seuls les endpoints autorisés peuvent capturer le jeton.
- **Jeton de preuve (Proof token) :** est un jeton qui ne peut être utilisé que par un client particulier. Chaque utilisation de ce jeton nécessite que le client effectue une action qui prouve qu'il est l'utilisateur autorisé du jeton.

OAuth définit deux types de jetons: jeton d'accès et jeton de rafraîchissement.

- **Jeton d'accès (Access token):** ce jeton est utilisé par le client pour accéder à une ressource. Les jetons d'accès ont généralement une courte durée de vie (minutes ou heures) qui couvre la durée de vie typique de la session.
- **Jeton de rafraîchissement (Refresh token):** est un jeton à longue durée de vie utilisé uniquement pour obtenir de nouveaux jetons d'accès, sans l'interaction du propriétaire de la ressource (sans forcer l'utilisateur à accorder à nouveau l'autorisation), lorsque le jeton d'accès actuel devient invalide ou expire. Ce type de jeton n'est échangé qu'entre le client et le serveur d'autorisation.

4. OAuth2.0 : Vulnérabilités, attaques et impacts

Dans cette section, nous allons présenter les problèmes de sécurité liés à la spécification OAuth2.0 et quelques implémentations de ce protocole dans l'environnement Web.

4.1. Propriétés de sécurité OAuth 2.0

Pour assurer la sécurité des modèles actuels de OAuth2.0, il est nécessaire de se baser sur quatre propriétés de sécurité essentielles, à savoir l'autorisation, l'authentification, la confidentialité et l'intégrité de la session.

Autorisation: L'autorisation dans OAuth 2.0 signifie qu'un attaquant ne devrait pas être en mesure d'obtenir ou d'utiliser une ressource protégée disponible pour un RP honnête auprès d'un IdP pour un utilisateur, sauf si, en gros, le navigateur de l'utilisateur ou l'IdP est corrompu.

Authentification: L'authentification dans le contexte de Oauth2.0 signifie qu'un attaquant ne devrait pas être en mesure de se connecter à un client (honnête) sous l'identité d'un utilisateur à moins, en gros, le serveur d'autorisation impliqué ou le navigateur de l'utilisateur est corrompu.

Confidentialité: La confidentialité dans OAuth2.0 repose entièrement sur le transfert sécurisé des données entre les différents rôles de OAuth2.0 en utilisant le protocole *Transport Layer Security* (TLS). Tout manquement à l'exigence de confidentialité entraînera un compromis OAuth 2.0 complet.

Intégrité de la session [6]: L'intégrité de la session, pour l'autorisation, signifie que (a) un client OAuth ne devrait être autorisé à accéder à certaines ressources d'un utilisateur que lorsque l'utilisateur a exprimé le souhait de démarrer un flux OAuth auparavant, et (b) si un utilisateur a exprimé le souhait de démarrer un flux OAuth en utilisant un serveur d'autorisation honnête et une identité spécifique, le flux OAuth n'est jamais terminé avec une identité différente (dans la même session). De même, pour l'authentification, il signifie que (a) un utilisateur a démarré un flux OAuth en entrant une identité pour se connecter, et (b) si, l'IdP utilisé dans ce flux est honnête, alors l'utilisateur est connecté sous exactement la même identité pour laquelle le flux OAuth a été démarré par l'utilisateur.

4.2. Vulnérabilités, attaques et impacts

La sécurité des protocoles d'autorisation Web repose à la fois sur la conception des sites Web qui les déploient et sur des mécanismes basés sur un navigateur tels que les cookies et JavaScript. Par conséquent, en plus de l'attaquant traditionnel de réseau qui peut détourner les connexions HTTP non sécurisées, ces protocoles sont également exposés à une nouvelle classe d'attaquants Web qui peuvent exploiter les vulnérabilités de sites Web telles que Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Open Redirect et SQLI (SQL Injection) pour compromettre partiellement les scripts client ou serveur exécutant ces protocoles. Grâce à la grande application du protocole d'autorisation OAuth2.0 dans l'environnement web et son adoption par les grands fournisseurs de services web tels que Facebook, Google et Microsoft, la sécurité de OAuth 2.0 est devenu une nécessité. Plusieurs efforts ont été faits pour analyser OAuth2.0 et trouver les problèmes sécuritaires liés à la spécification et quelques implémentations de ce protocole. Par exemple,

Wanpeng Li et al.(2019) ont ont développé un scanner de vulnérabilité et protecteur OAuth 2.0 et OpenID Connect appelé *OAuthGuard*. En analysant le top 1000 sites Web prenant en charge la connexion à Google (Google Sign-in), *OAuthGuard* a détecté cinq vulnérabilités de sécurité et de confidentialité appelées *CSRF Attack Threat Detection, Impersonation, Authorization Flow Misuse, Unsafe Token Transfers et Privacy Leaks*. En outre, Y. Zhou et al. (2014) ont utilisé SSOScan, un vérificateur de vulnérabilité automatique pour les applications utilisant les APIs Facebook SSO, pour étudier le top 2000 sites web pour quatre vulnérabilités SSO: *Access token misuse, Signed request misuse, App secret leak et User OAuth credentials leak*. Et, ils ont trouvé que parmi les 1660 sites qui utilisent Facebook SSO, plus de 20% souffraient d'au moins une vulnérabilité grave parmi les quatre.

Le tableau 1 présent les cinq fameuses vulnérabilités de OAuth 2.0 dans lesquelles trois sont classés parmi le top dix principaux risques d'applications Web, par OWASP (Open Web Application Security Project) [17]. Pour chaque vulnérabilité, nous montrons l'ensemble des attaques exploitant cette vulnérabilité, le type de cette faiblesse (de spécification ou d'implémentation) et l'impact technique sur la sécurité du système.

5. Conclusion

Dans cet article, nous avons présenté une analyse sécuritaire approfondie de OAuth 2.0 au niveau de spécification et d'implémentation dans l'environnement Web. Notre analyse, qui a visé la norme elle-même, ainsi que des implémentations spécifiques d'OAuth, comprend tous les flux (modes) d'autorisation d'OAuth et les paramètres de sécurité intégrés dans ce dernier pour atténuer les attaques et les problèmes de sécurité. En outre, nous avons défini les quatre propriétés de sécurité de OAuth2.0 qui doivent être pris en considération dans un tel système OAuth2.0. Et finalement, nous avons fait une étude sur les vulnérabilités, les attaques et leurs impacts sur la sécurité des systèmes basés sur OAuth2.0.

References

[1] Hardt ¡dick.hardt@gmail.com¿ D. The OAuth 2.0 Authorization Framework [Internet]. Disponible sur: https://tools.ietf.org/html/rfc6749.

[2] OAuth 2.0 — OAuth [Internet]. Disponible sur: https://oauth.net/2/

[3] Ghasemisharif M, Ramesh A, Checkoway S, Kanich C, Polakis J. O Single Sign-Off, Where Art Thou? An Empirical Analysis of Single Sign-On Account Hijacking and Session Management on the Web. :19.

[4] Sun S-T, Beznosov K. The Devil is in the (Implementation) Details: An Empirical Analysis of OAuth SSO Systems.

[5] Schwartz M, Machulak M. Securing the Perimeter : Deploying Identity and Access Management with Free Open Source Software [Internet]. Berkeley, CA : Apress ; 2018. Disponible sur : http ://link.springer.com/10.1007/978-1-4842-2601-8.

[6] Fett D, Kuesters R, Schmitz G. A Comprehensive Formal Security Analysis of OAuth 2.0. ArXiv160101229 Cs [Internet]. 6 janv 2016 ; Disponible sur: http://arxiv.org/abs/1601.01229

[7] Lodderstedt T, McGloin M, Hunt P. OAuth 2.0 Threat Model and Security Considerations [Internet]. RFC Editor; 2013 janv. Report No.: RFC6819. Disponible sur: https://www.rfc-editor.org/info/rfc6819

[8] Calzavara S, Focardi R, Maffei M, Wien T, Squarcina M, Tempesta M. WPSE: Fortifying Web Protocols via Browser-Side Security Monitoring.

[9] Schwartz M, Machulak M. Securing the Perimeter: Deploying Identity and Access Management with Free Open Source Software [Internet]. Berkeley, CA: Apress; 2018. Disponible sur: http://link.springer.com/10.1007/978-1-4842-2601-8.

[10] RFC 6819 - OAuth 2.0 Threat Model and Security Considerations: https://tools.ietf.org/html/rfc6819. Accessed: 2020-01-06.

[11] Bradley J, Labunets A, Lodderstedt T, Fett D. OAuth 2.0 Security Best Current Practice [Internet]. Disponible sur: https://tools.ietf.org/html/draft-ietf-oauth-security-topics-14

[12] C. Bansal, K. Bhargavan, A. Delignat-Lavaud, and S. Maffeis. Discovering Concrete Attacks on Website Authorization by Formal Analysis. Journal of Computer Security, 22(4):601–657, 2014. IOS Press, 2014.

[13] Wanpeng Li, Chris J Mitchell, & Thomas Chen (2019). OAuthGuard: Protecting User Security and Privacy with OAuth 2.0 and OpenID Connect. arXiv preprint arXiv:1901.08960

[14] Zhou Y, Evans D. SSOScan: Automated Testing of Web Applications for Single Sign-On Vulnerabilities. In 23rd USENIX Security Symposium. August 20–22, 2014.

[15] Rodriguez G, Torres J, Flores P, Benavides E. Cross-Site Scripting (XSS) Attacks And Mitigation: A Survey. Comput Netw. 1 nov 2019.

[16] SQL Injection— OWASP. Disponible sur: https://owasp.org/www-community/attacks/SQL_Injection

[17] The Ten Most Critical Web Application Security Risks, Top OWASP 10,Toronto, ON, Canada, 2013.

Vulnérabilités	Attaques	Type	Impact
Cross Site Scripting (XSS) [12], [4], [15], [17]	• Attaque CSRF • Attaque SQL injection • Attaques persistant XSS • Attaques non-persistant XSS • Attaques DOM XSS	Implémentation	Authentification, autorisation et confidentialité
Cross Site Request Forgery (CSRF) [12], [13], [4], [11], [17]	• Attaque Social login CSRF • Attaque Form CSRF • Attaque Automatic login CSRF • Attaque Social sharing CSRF • Attaque Session swapping • Attaque access token theft • Attaque cookie theft	Spécification et Implémentation	Authentification, autorisation et confidentialité
Impersonation [12], [11], [4]	• Attaque Clickjacking • Attaque State leak • Attaque Naive RP Session Integrity • Attaque CSRF • Session swapping	Spécification et Implémentation	Intégrité de la session
Open redirect [12], [11]	• Attaque Phishing • Attaque Resource theft by access token redirection • Attaque unauthorized login by code redirection • Attaque 307 redirect	Spécification et Implémentation	Authentification, autorisation et Intégrité de la session
SQL Injection [12], [17]	• Attaque SQL Injection Bypassing WAF • Attaque Blind SQL Injection • Attaque Code Injection • Attaque Token Injection • Attaque Double Encoding	Implémentation	Authentification, autorisation, confidentialité et intégrité de la session

TABLE 1. LES VULNÉRABILITÉS DE SPÉCIFICATION ET D'IMPLÉMENTATION CONTRE OAUTH2.0

Application of Artificial Neural Networks on Growth Prediction of E. Coli in Drinking Water.

A. EL HATIMY
LIMATI Laboratory
Polydisciplinary Faculty
Sultan Moulay Slimane University
Email:a.elhatimy@gmail.com

S. SAFI
LIMATI Laboratory
Polydisciplinary Faculty
Sultan Moulay Slimane University
Email: safi.said@gmail.com

A. BOUMEZZOUGH
EPANT Laboratory
Polydisciplinary Faculty
Sultan Moulay Slimane University
Email: ahmed.boumezzough@gmail.com

***Abstract*—Cell growth rate prediction remains a challenging task because of the extreme variation of physicochemical variables that are responsible of how the bacteria are reproducing. In this paper, we have investigated and compared tow approaches used to automate the process of E. Coli growth prediction with the use of artificial neural networks: Radial Basis Function Networks (RBFN) and Recurrent Neural Networks (RNN). We have used dataset that is constituted of 5 columns and 956 rows where temperature (10 to 39C), pH (6.42 to 9.96), electrical potential (0,02 to 0,98 mV) and electrical conductivity (0,8 to 17 mS) are feeded as inputs of the networks and logarithm of the cell number as the output.**

I. Introduction

Escherichia coli (E. coli) represents 80% of the bacteria in our digestive tract [1]. It is harmless in most cases, but some strains are found to be pathogenic [1][2]. The pathogenic types are dangerous for the body and responsible for diseases like gastroenteritis, urinary tract infections, meningitis ...[3]. Bacteria growth rate prediction is very important in many healthcare disciplines and food security. If the conditions are favorable for growth, the evolution of the microbial population always follows the same profile [1][3]. Food microbiologists are constantly concerned to determine, experimentally, the extent to which microbial growth depends on factors relating to ecology or processing modes [4].

The so called traditional techniques that are most widely used to define relationships between combinations of factors and growth parameters are based on three main types: the Arrhenius equation, the square root model (Blehrdek model) and the Response Surface Model[5]. These methods use the variations of physicochemical parameters accompanying bacterial metabolism, for example: pH, electrical conductivity, electrical potential, the optical density, the oxygen ... etc[5]. The measurements of one of these parameters can be considered as an indicator of the presence or absence of this type of bacteria[5]. in the last decade, a lot of new and advanced models were introduced, most of them were based on artificial neural networks (ANNs) technology. ANNs are a highly connected layers of neurons that try somehow to simulate the behavior of human neuro-biological system [6], they are known for there ability to perform complex and non linearly separable problems [6]. Hence, ANNs directly extract the information contained in the dataset to adjust its parameters, as the input output patterns are repeatedly presented to the network.

In this paper, authors propose a comparative study of tow types of ANNs architectures: recurrent neural networks (RNN) [7] and radial basis function network (RBFN) [8] applied to E. coli growth prediction. For RBFNs, physicochemical variables (Temperature, pH, Electrical potential and Electrical conductivity) feeded as inputs and logarithm of bacterial number as output, in the other hand, these physicochemical variable are used in combination with previous outputs (Logarithm of bacterial number) to regenerate new outputs of RNNs. The numerical simulation results, including comparison illustrations, are presented to describe the behavior of each approach.

II. Data preprocessing

In order to compar the result of the two algorithms, the use of one and only one dataset is required. Thus, the dataset, obtained from [9], is constituted of 5 columns (temperature, pH, electrical potential, electrical conductivity, and logarithm of bacterial number) and 956 rows, 80% of the data is used for training and parameters regularization purpose and 20% to test the model performance.

The features were scaled to the rank $[0, 1]$ due to their different measurement ranges and to avoid saturation problems in the activation function in the network. The MinMax scaler is used in our case [10] :

$$\hat{x} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where $\hat{x}$ is the scaled value and $max(x)$, $min(x)$ are respectively the maximum and the minimum values of x.

The following tables show the data before and after MinMax transform :

TABLE I
Before MinMax transform.

Y	T(C)	pH	Pot. (mV)	Cond (mS)
4.80	15	8.45	0.85	3.01
4.81	22	8.45	0.85	3.01
4.84	18	8.44	0.84	3.02

TABLE II
AFTER MINMAX TRANSFORM.

Y	T(C)	pH	Pot. (mV)	Cond (mS)
0.254	0.172	0.573	0.864	0.138
0.256	0.413	0.573	0.864	0.138
0.260	0.275	0.570	0.854	0.1389

III. RADIAL BASIS FUNCTION MODEL.

RBFNs are a powerful models used, in general, to approximate functions and curve fitting [6]. One of their main particularities is the speed of training process due to simplicity of their architecture, another distinguishing feature of this kind of ANNs is the activation function used in the intermediate layer, witch is known as radial basis function such as the Gaussian function [6][11]:

$$\varphi_j(x) = \exp(-\frac{1}{2\sigma_j^2} \parallel x - x_j \parallel^2) \quad j = 1, 2, ..., N \qquad (2)$$

where σ_j is a measure of the width of the jth Gaussian function with center (inputs) x_j, all the Gaussian hidden units are assigned a common width σ.
The architecture of RBFNs consist of input layer, RBF layer and an output layer. In our case, the input layer is constituted of 4 neurons witch represent the input data (four features), the intermediate layer consist of 15 neurons as an optimal number in our case, considering the optimal testing results, and only one neuron for the output layer [6][11]. The following figure (Fig.1) illustrates a typical configuration of an RBFN :

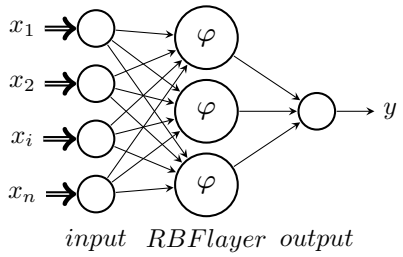

Fig. 1. RBFN architecture.

Thus, the output of each neuron j of the intermediate layer is expressed by [11]:

$$\varphi_j(x) = \exp(-\frac{((\sum_{i=1}^{n} x_i w_{ij}) - x_j)^2}{2\sigma_j^2}) \qquad (3)$$

Where $i = 1, 2, ..., n$ correspond to the input neurons, j is for the hidden neurons and w_{ji} are the weights.

Adam optimizer [12] is used, in our case, to adjust the network parameters. The error (eq.4) is propagated backward and forward in order to adjust the weights of the network [7]:

$$MSE = \sum_{i=1}^{n} \frac{(y_i^{(actual)} - y_i^{(predicted)})^2}{n} \qquad (4)$$

IV. RECURRENT NEURAL NETWORKS BASED APPROACH.

Recurrent neural networks (RNNs) are a type of neural networks where certain neurons have connections (loops) from the output to input or to the inputs of neurons of the previous layers, in the sense that they keep information in memory: they can take into account at time t a certain number of past states[11]. To sum up, the values calculated by the neural network can be fed back to its input layer or any previous layers.
In this paper, the input layer consists of 6 elements: temperature, pH, electrical conductivity, electrical potential and tow neurons for the previous output ($Y_{t-\Delta t}$ and $Y_{t-2*\Delta t}$), 20 neurones in the hidden layer was determined as the best structure and an output of one element. The hyperbolic tangent was chosen as an activation function for each neurone. as for the MLP, the weights of the neural connections, initially chosen randomly, are adjusted by Adam optimizer [12]. The following figure (fig.2) illustrates the architecture of the network used in our case:

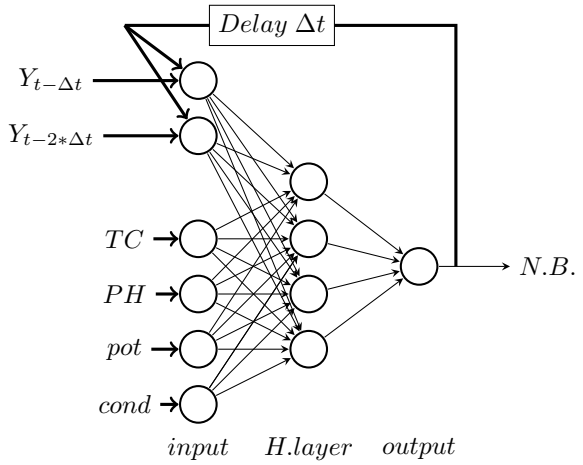

Fig. 2. RNN architecture.

V. RESULTS AND DISCUSSION.

A priori, there are no rules for the choice of the hidden structure, it is determined empirically: the structure that gives the best results is chosen. The optimum neurone number of the hidden layer and the learning rate were iteratively determined by developing several neural networks and simultaneously observing the change in the mean square of the output error. The result exhibited that learning parameter at 0.8 for RBFN and 0.9 for RNN gave the minimum error (table.3) after running program for 30 cycles

TABLE III
THE MINIMUM TESTING ERROR.

	RNN	RBFN
MSE	$6,1.10^{-4}$	$1.5.10^{-3}$

Two different types of networks were designed: radial basis function network (RBFN) and recurrent neural network (RNN), they were set up to predict the logarithm of bacterial numbers values. Figure (fig.3) and (fig.4) shows, respectively, the training error and the validation error of each model. For the training process, as it is shown in the figure (fig.3), RBFN's curve converge faster than RNN's due to the network architecture witch is less deeper(low number of neurons in the input and hidden layers compared to RNN). However, it is clear that after 30 iterations RNN gives the lowest mean squared error (MSE).

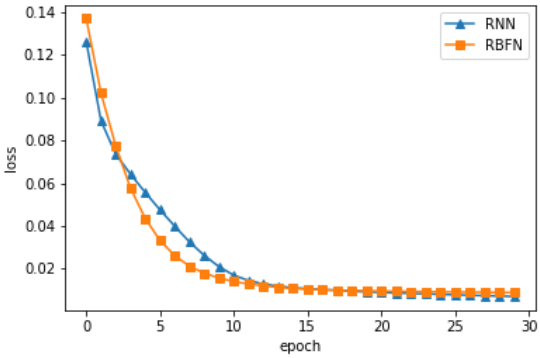

Fig. 3. Training error.

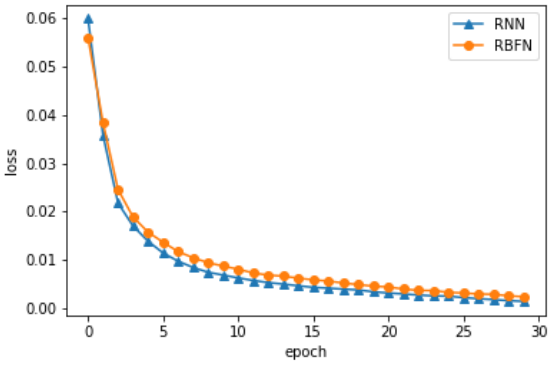

Fig. 4. Validation error.

From table.3 it is clear that RNN model has the lowest testing mean squared error (MSE) wish means the best testing performance, as it can be clearly seen in the table.4, RNN gave satisfactory results compared to RBFN witch is more clarified in the figure (fig.5).

All of this demonstrates that RNN has good generalization ability when it comes to accurately predicting the growth of E.Coli, and that's because we deploy the previous stocked outputs to generate more informations about datasets in a way that we combine between the flexibility of simple multilayered perceptron and capability of time series on predicting in a temporal space.

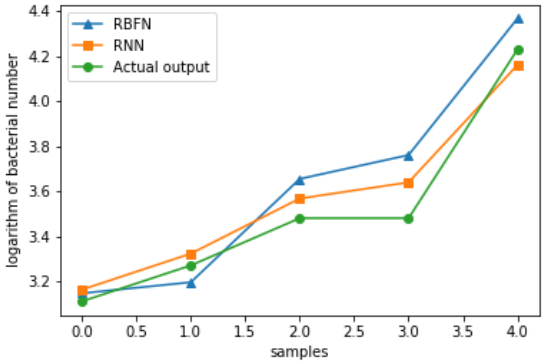

Fig. 5. the predicted output of each model versus the experimented output.

TABLE IV
PREDICTED VALUES OF LOGARITHM OF BACTERIAL NUMBER VS THE ACTUAL VALUES

TC	pH	Pot.	Cond	Y	Y_RNN	Y_RBFN
26	6,48	0,35	11,12	5,3	5,32	5,41
15	7,96	0,32	2,18	3,27	3,27	3,27
12	8,06	0,39	2,14	3,2	3,2	3,29
21	7,93	0,55	3,51	5,75	5,74	5,75
20	7,43	0,54	15,31	9,64	9,42	9,19
24	8,35	0,79	3,13	5,13	5,09	5,14
23	7,58	0,12	2,47	3,46	3,45	3,59
25	7,83	0,48	3,6	5,85	5,92	5,89
23	7,51	0,11	2,51	3,49	3,53	3,62
28	7,31	0,25	3,91	6,37	6,38	6,39
18	7,02	0,92	12,95	7,68	7,65	7,69
24	7,34	0,28	3,89	6,33	6,31	6,36
12	7,34	0,23	2,62	4,89	4,91	4,77
21	6,88	0,73	11,96	6,81	6,84	6,98
36	7,38	0,61	14,99	9,45	9,29	9,06
26	7,94	0,3	2,21	3,28	3,25	3,20
14	7,12	0,88	13,75	8,27	8,22	8,21
39	7,53	0,11	2,5	3,48	3,54	3,63
19	7,91	0,28	2,24	3,3	3,28	3,25
24	6,98	0,04	4,32	6,9	6,82	6,83
12	7,26	0,73	14,49	8,88	8,89	8,83
29	7,31	0,24	3,92	6,38	6,41	6,41
14	7,02	0,87	12,88	7,65	7,63	7,79

VI. CONCLUSION.

Clinical microbiology is a wide, varied and complex field of investigation. what we proposed in this work is a comparative study between tow different artificial neural networks based approaches, recurrent neural networks (RNN) and radial basis function (RBFN). In term of performance, we proved that RNNs are more efficient and we proved that they give the lowest training and testing errors. For a future work, we are going to apply a convolutional neural networks to microscopic images dataset, in order to predict the colony number in a certain image.

REFERENCES

[1] Christopher JAlteri and Harry LTMobley. *Escherichia coli physiology and metabolism dictates adaptation to diverse host microenvironments*, Current Opinion in Microbiology Volume 15, Issue 1, February 2012, Pages 3-9

[2] BGregory L Armstrong, Jill Hollingsworth and J. Glenn Morris, Jr. *Emerging Foodborne Pathogens: Escherichia coli O157:H7 as a Model of Entry of a New Pathogen into the Food Supply of the Developed World*, Vol. 18, No. 1, 1996.

[3] Thomas A.Russo and James R; Johnson *Medical and economic impact of extraintestinal infections due to Escherichia coli: focus on an increasingly important endemic problem*, Microbes and Infection,Volume 5, Issue 5, April 2003, Pages 449-456.

[4] Hajmeer M, Basheer I, Najjar Y. *Computational neural networks for predictive microbiology II. Application to microbial growth*, Int J Food Microbiol 34:51-66,1997.

[5] Whiting R, Buchanan R. *Microbial modeling.* , Food Technol 48:113-20, 1994.

[6] Haykin, S. *Neural Networks. A Comprehensive Foundation.*, MacMillan College Publishing Company, 1994.

[7] M. Cheroutre-Vialette, A. Lebert. *Application of recurrent neural network to predict bacterial growth in dynamic conditions*. International Journal of Food Microbiology, 73 (2002) 107 118.

[8] Francisco Fernandez-Navarroa,Csar Hervs-Martnez,Cruz-Ramrez, Pedro Antonio Gutirrez and Antonio Valero. *Evolutionary q-Gaussian Radial Basis Function Neural Network to determine the microbial growth/no growth interface of Staphylococcus aureus*. Applied Soft Computing 11 (2011) 30123020.

[9] B. Saddek *Dtection automatique par des techniques de lintelligence artificielle des indicateurs de contamination microbiologique dans les eaux de consommation*. UFAS (ALGERIE), 2018.

[10] R.M. GARCA-GIMENO, C. HERVS-MARTNEZ, E. BARCO-ALCAL, G. ZURERA-COSANO, AND E. SANZ-TAPIA *An Artificial Neural Network Approach to Escherichia Coli O157:H7 Growth Estimation*. Journal of Food Science March 2003

[11] I. N. Da Selva, Danilo Hernane and other *Artificial Neural Networks*. Springer International Publishing Switzerland 2017.

[12] Diederik P. Kingma and Jimmy Ba, *Adam: A Method for Stochastic Optimization*. the 3rd International Conference for Learning Representations, San Diego, 2015.

Analyse, caractérisation et conception d'un système radar anticollision embarqué dans les véhicules

Hassan FACOITI
Faculté Polydisciplinaire
Université Sultan Moulay Slimane
Email : hassan.facoiti@gmail.com

Said SAFI
Faculté Polydisciplinaire
Université Sultan Moulay Slimane
Email : safi.said@gmail.com

Ahmed BOUMEZZOUGH
Faculté Polydisciplinaire
Université Sultan Moulay Slimane
Email : ahmed.boumezzough@gmail.com

***Résumé*—Dans cet article, nous nous concentrons sur la conception et la production d'un prototype de système radar anticollision embarqué dans les véhicules afin de devenir autonome. Basé sur la carte Arduino et grâce à des capteurs à ultrasons, ce système électronique peut détecter simultanément plusieurs cibles dans l'environnement du véhicule. Il affiche également les distances de détection d'obstacles et gère le contrôle intelligent de la manoeuvre sans intervention humaine. Sur la base d'une étude théorique et en comparant les différentes solutions susceptibles de résoudre le problème du projet, nous avons réussi à réaliser un produit qui répond aux besoins mentionnés avec la possibilité d'une optimisation supplémentaire.**

Mots clés : Radar anticollision, Voiture autonome, Robotique, Détection d'obstacles.

I. Introduction

La recherche sur la sécurité des véhicules est liée à un radar anticollision et à un système d'aide au conducteur. Le but du système radar est essentiellement la mesure de la distance entre deux véhicules ou entre un véhicule et un objet. Il est utilisé pour faciliter le stationnement même lorsque la visibilité arrière est nulle et mesure la vitesse d'approche d'un obstacle afin que le conducteur puisse être informé du risque de collision. Eventuellement aussi pour gérer les dispositifs de freinage dans les situations dangereuses [1].

Les mesures de distances et de vitesses relatives des véhicules sont obtenues généralement par l'effet Doppler. C'est la différence de fréquence entre les ondes émises par un radar et les ondes reçues après la réflexion sur les obstacles. Le problème général est de compléter l'emplacement des obstacles mobiles ou stables par rapport au véhicule, et de déterminer leur direction de déplacement [2].

Le but de cette étude est de mettre en place un système radar anticollision capable de mesurer la distance entre deux véhicules ou entre un véhicule et un objet, ainsi que de détecter simultanément plusieurs obstacles dans l'environnement du véhicule. Afin de rendre le véhicule autonome. Dans cet article, nous présentons une étude sur les systèmes radar et leurs méthodes de fonctionnement. Puis nous présentons la conception et l'étude théorique de notre système robot (radar anticollision, voiture autonome), et les différentes parties constituant la chaîne de l'information et la chaîne d'énergie du robot. Dans la section suivante, nous présentons les modèles expérimentaux de notre système radar qui nous permettent d'effectuer des mesures de distance, et la mise en œuvre du système qui est installé sur une voiture télécommandée, en détectant la distance d'une cible et en gérant l'intelligent contrôle de la manœuvre. Par la suite, nous étudions la précision des mesures et les performances du système ainsi que leurs propriétés et limites, puis l'influence du milieu de propagation sur sa fiabilité.

II. Types de systèmes radar anticollision

Les types de systèmes radar classées selon les formes d'onde : les radars à impulsion et les radars à onde continue (FMCW , Duplex). Dans un premier temps, nous commençons par l'effet Doppler qui permet de calculer la vitesse de la cible (dans le champ radar, l'obstacle est appelé cible).

A. Effet Doppler

L'effet Doppler est le décalage de fréquence d'une onde (acoustique, électromagnétique, ...) observé entre les mesures à l'émission et à la réception des ondes, grâce à la variation de la distance en fonction du temps entre l'émetteur et le récepteur. Le nom "Effet Doppler-Fizeau" est réservé aux ondes électromagnétiques. Plusieurs cas peuvent être considérés de la variation de distance entre l'émetteur et le récepteur. Dans chaque cas, il peut être envisagé que l'émetteur et le récepteur, s'éloignent ou se rapprochent l'un de l'autre [7]. Pour calculer la vitesse, il faut mesurer la fréquence du récepteur et extraire la vitesse à partir des formules des cas de mouvement de l'émetteur et le récepteur :

1) Emetteur en mouvement et récepteur immobile:
De la (Fig.1), il y a trois cas de fréquence perçue par le récepteur :

⁕ 1^{re} cas : La voiture s'approche d'un récepteur immobile.

$$f = \frac{c}{c - V_e} . f_e \tag{1}$$

⁕ 2^{me}cas : La voiture s'éloigne d'un récepteur immobile.

$$f = \frac{c}{c + V_e} . f_e \tag{2}$$

⁕ 3^{me} cas : La voiture est stationnaire, et émet une fréquence f_e. Le récepteur perçoit la fréquence f, tel que :

$$f = f_e \tag{3}$$

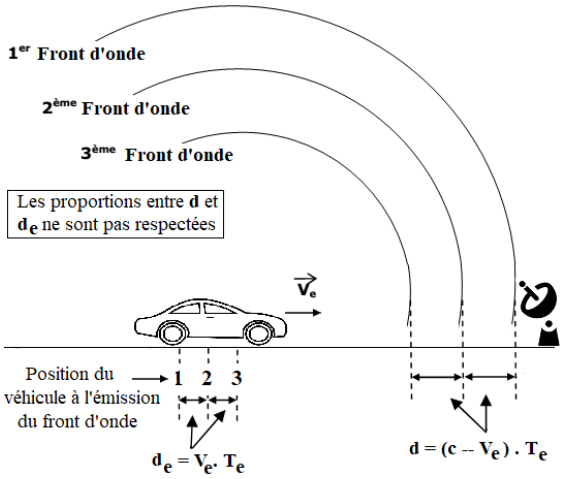

FIGURE 1. Émetteur en mouvement et Récepteur immobile.

2) Émetteur immobile et récepteur en mouvement:

⁎ 1^{re} cas : L'émetteur immobile émet une fréquence f_e (Fig.2).

$$f = \frac{c + V_r}{c} . f_e \tag{4}$$

⁎ 2^{me} cas : Le récepteur s'éloigne d'un émetteur stationnaire.

$$f = \frac{c - V_r}{c} . f_e \tag{5}$$

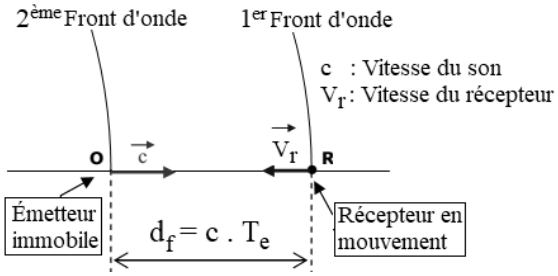

FIGURE 2. Émetteur immobile et Récepteur en mouvement

3) Émetteur et récepteur tous deux en mouvement:

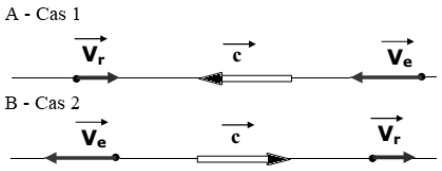

FIGURE 3. (A)-Cas 1 : l'émetteur et le récepteur se rapproche l'un de l'autre. (B)-Cas 2 : l'émetteur et le récepteur après le croisement.

⁎ 1^{re} cas : l'émetteur et le récepteur se déplacent en sens inverse :

• Cas 1 : (Fig.3(A))

$$f = \frac{c + V_r}{c - V_e} . f_e \tag{6}$$

• Cas 2 : (Fig.3(B))

$$f = \frac{c - V_r}{c + V_e} . f_e \tag{7}$$

⁎ 2^{me} cas : l'émetteur et le récepteur se déplacent dans le même sens :

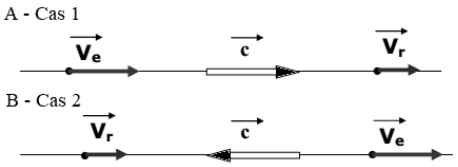

FIGURE 4. (A)-Cas 1 : l'émetteur et le récepteur avant le dépassement. (B)-Cas 2 : l'émetteur et le récepteur après le dépassement.

• Cas 1 : (Fig.4(A))

$$f = \frac{c - V_r}{c - V_e} . f_e \tag{8}$$

• Cas 2 : (Fig.4(B))

$$f = \frac{c + V_r}{c + V_e} . f_e \tag{9}$$

B. Radars à impulsions

Le radar est une combinaison d'un émetteur, d'un récepteur et d'un système d'exploitation. Il utilise la propriété des ondes électromagnétiques ou acoustiques pour détecter tout obstacle, grâce à une onde de retour détectable par un récepteur adapté à ce signal. Ce principe peut facilement être démontré lorsque le signal transmis est une séquence d'impulsions.

1) Radars à impulsions électromagnétiques:

Ce type de radar utilise les propriétés réfléchissantes d'une onde électromagnétique sur un obstacle en créant une onde réfléchie captée par le dispositif de réception. La forme d'onde la plus simple pour étudier ce principe est une série d'impulsions. Une impulsion a une courte durée T_i. Pour de nombreux radars, la durée d'une impulsion est de l'ordre de la microseconde. L'existence de la cible aide à refléter une partie de l'impulsion. Les caractéristiques temporelles sont conservées, mais la réflexion conduit à perdre une partie de l'amplitude. C'est pourquoi la mesure de la puissance reçue est peu exploitée pour une évaluation précise de la distance. En mesurant le retard τ entre l'émission et la réception on peut calculer la distance entre le radar et la cible par l'équation suivante [6] :

$$d = c.\frac{\tau}{2} \tag{10}$$

La durée d'une impulsion T_i et la fréquence de répétition f_r sont les paramètres les plus importants pour calculer la plus grande distance mesurable et la résolution du radar (Fig.5). Pour éviter l'ambigüité dans la mesure de la distance, il faut que l'écho de la cible soit reçu par le radar avant que

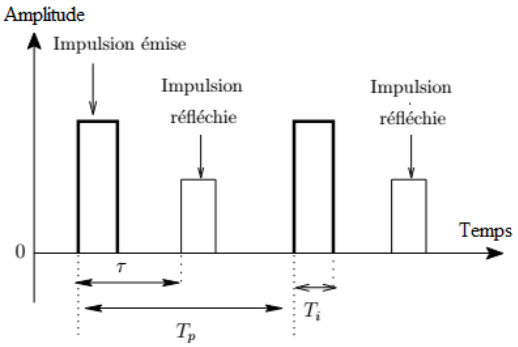

FIGURE 5. Radar à impulsions.

l'impulsion suivante soit émise. Donc, le temps d'un aller-retour (retard τ) doit être inférieur à la période des impulsions :

$$T_p = \frac{1}{f_r} \quad et : \quad T_p = \frac{2d_{max}}{c} \tag{11}$$

Ce qui nous donne une distance maximale de :

$$d_{max} = \frac{cT_p}{2} \tag{12}$$

La plus petite variation de distance qu'on peut trouver est déterminée par :

$$\Delta d = \frac{cT_i}{2} \tag{13}$$

2) Radars à impulsions acoustiques:
Le radar acoustique fonctionne sur le même principe qu'un radar à impulsions électromagnétiques, mais il n'utilise pas le même type d'onde. Il utilise le principe de propagation des ondes ultrasonores dans l'air qui se réfléchissent lorsqu'ils rencontrent un obstacle. Ce type de radar basé sur un capteur à ultrasons. Dans l'industrie automobile on trouve ce type de radar acoustique appelé (Radar de recul). Les types de véhicules équipés d'un radar de recul, voire plusieurs, bénéficient des équipements appelés "assistance au stationnement". Le plus souvent placé sur le pare-chocs arrière.

3) Capteur à ultrasons:
Ce capteur a été choisi dans l'industrie automobile car tous les matériaux qui reflètent le son peuvent être détectés, quelle que soit leur couleur. Même les matériaux transparents ou les feuilles minces ne représentent aucun problème pour un capteur à ultrasons, sauf les objets absorbent les ondes sonores. Le signal ultrasonique n'est pas influencé par la poussière et les environnements brumeux, même les dépôts minces sur le diaphragme du capteur.

Les ondes ultrasonores sont générées par l'effet piézoélectrique. Certains matériaux, comme le quartz, ont la propriété de vibrer lorsqu'une tension est appliquée sur ses bornes, cet effet est réversible : lorsque le quartz vibre à cause des ultrasons, une tension apparaît à ses bornes. Lorsque le détecteur est sous tension, l'élément piézoélectrique commence à vibrer, le cas du radar de recul se traduit par l'apparition d'une fréquence de l'ordre de 40 kHz. La vibration est transmise à l'air ambiant grâce à la face avant du capteur qui est composé de céramique.

III. VOITURES AUTONOMES

Une voiture est dite autonome si elle est équipée d'un système de conduite automatique qui lui permet de circuler et de maintenir la sécurité routière sans intervention humaine dans des conditions de circulation réelles. En outre, elle peut être capable de gérer les tâches suivantes : Maintenir une distance de sécurité par rapport aux autres véhicules, adapter la vitesse du véhicule en fonction des conditions de circulation et des caractéristiques de la route, détecter les piétons, changer de voie pour éviter les obstacles et les collisions, parking automatique en créneau des voitures [11] [12].

Les voitures autonomes doivent être équipées de capteurs (LIDAR, antenne GPS, capteur à ultrasons, radars, caméra vidéo) collectant des informations brutes sur leur environnement. Ces informations sont transmises à des unités de traitement informatique, contiennent des logiciels capables d'adapter la vitesse et la trajectoire du véhicule en estimant les mouvements d'autres véhicules ou piétons.

IV. CONCEPTION ET ÉTUDE THÉORIQUE

La robotique et l'automatisation industrielle sont des domaines qui englobent toutes les réalisations scientifiques et sont les domaines scientifiques les plus prometteurs pour la sécurité routière, en particulier pour rendre les voitures autonomes. L'automatisation industrielle est souvent associée à la robotisation, c'est-à-dire à l'utilisation d'une technique qui assure le fonctionnement d'une machine ou d'un groupe de machines sans intervention humaine. L'automatisation utilise des outils numériques et des automates programmables ou des microcontrôleurs pour guider et donner des informations aux machines. Donc on fait appel à des systèmes électroniques qui regroupent toute la hiérarchie de contrôle-commande (Fig.6).

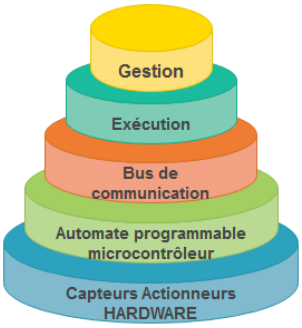

FIGURE 6. Hiérarchie de contrôle−commande

V. ANALYSE FONCTIONNELLE

Avant de commencer la partie réalisation une approche théorique est très nécessaire pour comprendre le fonctionnement de notre système. Cette étude vise à faciliter le concept d'un robot autonome (une voiture autonome basée sur un Radar anticollision), à étudier de plus près les composants et à faire une analyse des phénomènes physiques. Pour que nous établissons une analyse fonctionnelle a priori afin d'associer à chaque besoin la solution constructive appropriée, on peut répartir

toutes les solutions selon le diagramme chaine information et chaine énergie (Fig7). La (Fig8) illustre une modélisation 3D de la vue globale de notre système installé sur un châssis.

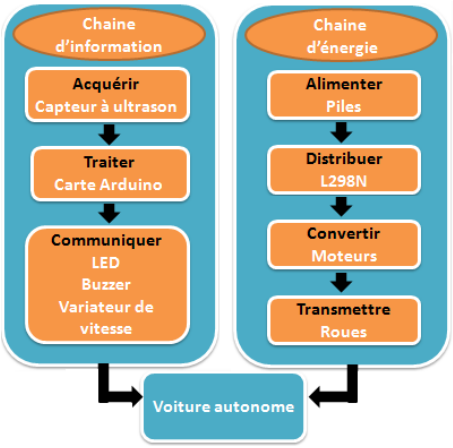

FIGURE 7. Diagramme chaine d'information et chaine d'énergie

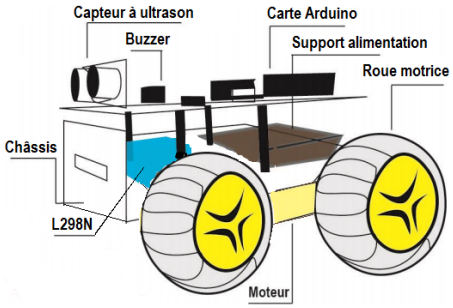

FIGURE 8. Vue globale du robot

VI. MODÈLES EXPÉRIMENTAUX

A. Premier modèle expérimental

Ce modèle repose sur un capteur à ultrasons connecté aux broches numériques de l'Arduino. Un buzzer et deux LEDs sont activés en présence d'un obstacle entre 0 et 30 cm, et un écran LCD pour afficher la distance entre le capteur et l'obstacle. Nous testons l'utilisation de ce système sur une voiture programmée pour un freinage forcé si un obstacle est détecté à 30 cm de distance pour éviter une collision.

B. Deuxième modèle expérimental

Ce modèle scan une sphère de détection à un angle de 120° (30°- 150°). Nous recevons les valeurs de l'angle et de la distance mesurées par le capteur de la carte Arduino dans l'IDE de traitement (Fig.9). Les lignes en verts indiquent l'absence d'obstacles à une distance inférieur à 40 cm. Si le radar détecte un objet ou plus, les lignes deviennent rouges et la coloration commence avec la position de l'objet. Nous utilisons ce radar pour concevoir le deuxième modèle d'une voiture autonome qui détecte les obstacles et prenait d'autres chemins.

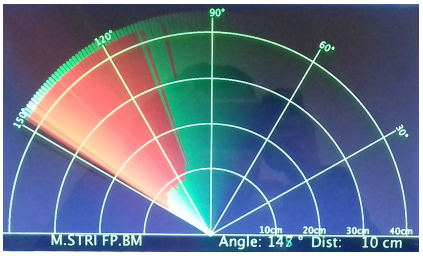

FIGURE 9. Radar de vision à un angle de 120°.

C. Troisième modèle expérimental

Le 2^{me} modèle est basé sur la rotation du servo moteur pour scanner l'environnement de la voiture, ce qui donne un retard grâce à cette rotation, pour cela nous avons développé le modèle par l'ajout de trois capteurs fixes, et chacun effectue la détection à sa direction. Ce modèle est programmé pour fournir la distance minimale détectée par les capteurs.

VII. PERFORMANCE DU SYSTÈME

A. Précision du système

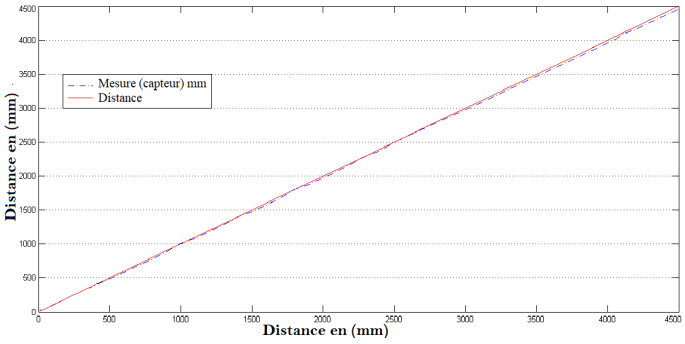

FIGURE 10. Graphique de linéarité du système sur la plage 10 mm−4.5 m.

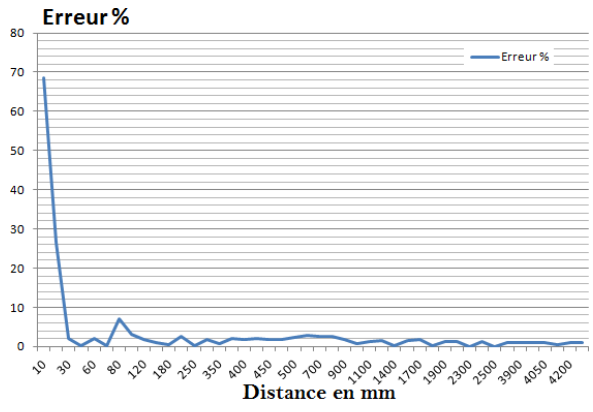

FIGURE 11. Graphique d'erreurs du système sur la plage 10 mm−4.5 m.

La figure (Fig.10) illustre les résultats des mesures de test de précision, la courbe rouge représentant la théorie et la courbe pointillée bleue représente la pratique. Les deux sont

presque identiques, Il y a une légère dérivé avec un taux d'erreur de $\sim$ 3.705 % en moyenne. Nous remarquons dans les 3 cm premiers que le taux d'erreur est important (Fig.11), c'est la zone aveugle du capteur qui correspond à la distance de détection minimale pour que notre système fonctionne correctement. De plus, Nous obtenons une précision de $\pm$ 1.38 cm qui pourrait suffire à notre radar. Et la distance maximale que nous pourrions mesurer avec notre système est 4.445 m.

B. Influence du milieu de propagation

1) Milieu chaud:

La figure (Fig.12) illustre le système radar détecte un obstacle à une distance fixe de 70,74 cm (la courbe pointillée bleue). Ensuite, la distance augmente par rapport à l'augmentation de la température du milieu, cette variation due à la variation de la vitesse de propagation des ultrasons. Pour cela nous ajoutons un capteur de température à l'ensemble, pour adapter l'émission des ultrasons à une vitesse dépendant de la température du milieu, selon l'équation (14) en utilisant les résultats de Wiliam M.Haynes (Fig.13) [13].

$$V = 331.5 + 0.607.\theta \tag{14}$$

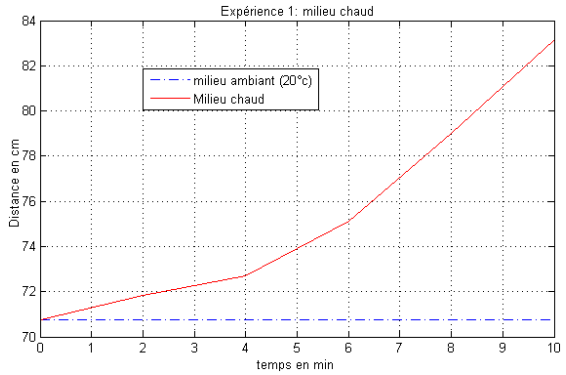

FIGURE 12. Graphique du résultat obtenu dans le milieu chaud

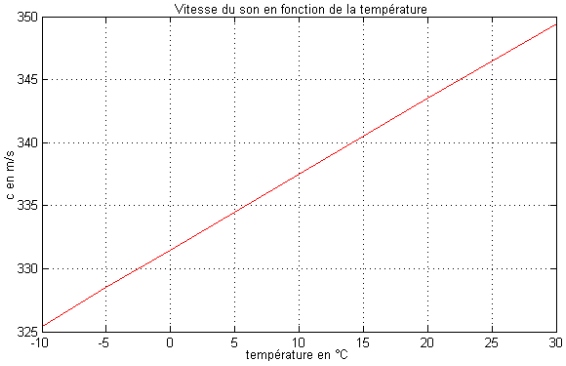

FIGURE 13. Vitesse du son en fonction de la température

2) Milieu pollué:

On met un obstacle à une distance fixe (la courbe pointillée bleue), puis on simule un milieu pollué par la poussière du sol et de la combustion du charbon. Nous notons la variation de la distance à mesurer en fonction du temps. Nous obtenons le résultat dans la (Fig.14). Les résultats obtenus montrent que la variation de distance est très faible au millimètre près. Par conséquent, le milieu pollué n'influence pas sur notre système.

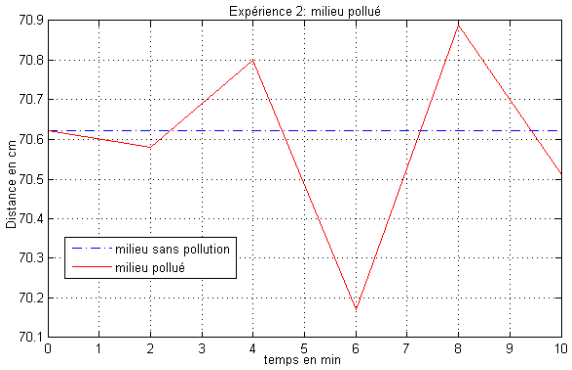

FIGURE 14. Graphique du résultat obtenu dans le milieu pollué

3) Milieu pluvieux:

Nous plaçons le système devant un obstacle fixe à une distance de 115.84 cm et nous simulons un environnement pluvieux de 70 à 115 cm. Nous notons que le système détecte la pluie (Fig.15), cela est considéré comme une limite de celui-ci.

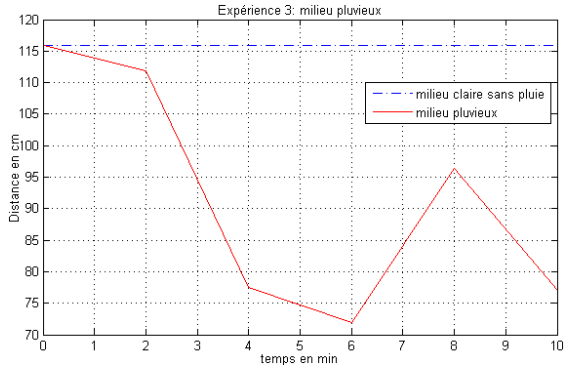

FIGURE 15. Graphique du résultat obtenu dans le milieu pluvieux

VIII. RÉSULTATS

L'étude de la performance et de l'influence du milieu de propagation, nous a permis de connaître les limites de notre système, et d'exploiter les résultats pour améliorer autant que possible. Au final, nous avons réussi à fabriquer deux cartes électroniques d'un radar de l'avant et de l'arrière valables d'embarqué sur le véhicule (Fig.16 et (Fig.17)).

Notre robot (Fig.18) basé sur le radar anticollision capable de détecter les obstacles grâce à 6 capteurs à ultrasons, 3 placés devant le châssis et 3 à l'arrière. L'Arduino traite les signaux acquis par ses ports connectés aux différents capteurs puis commande 4 moteurs à courant continu via le module L298N

capable d'alimenter ces 4 moteurs avec une alimentation externe adaptée et ainsi inverser le sens de rotation, faire varier la vitesse et forcer le freinage selon à l'organigramme de la manœuvre de la figure (Fig.19).

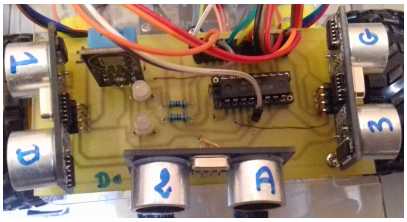

FIGURE 16. Carte électronique : Radar-avant

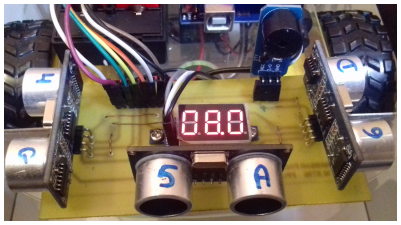

FIGURE 17. Carte électronique : Radar-arriére

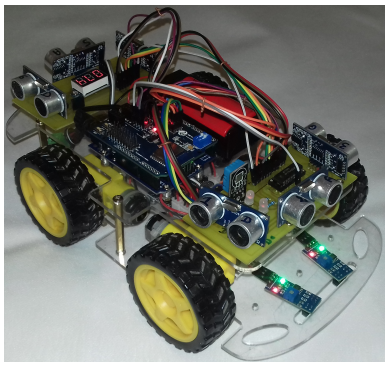

FIGURE 18. Robot : Assemblage et câblage des cartes ensemble sur le châssis. Construction du prototype d'une voiture autonome

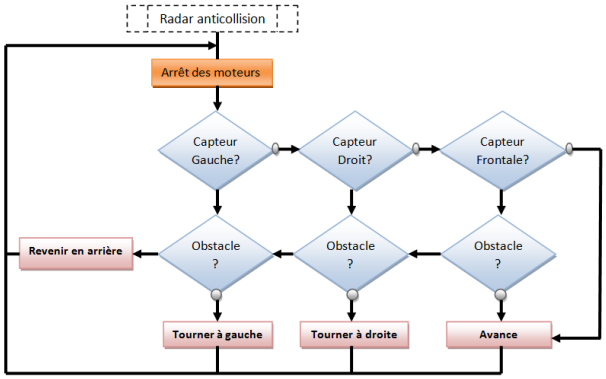

FIGURE 19. Organigramme de la fonction du robot

IX. CONCLUSION

Comme tout produit industriel, le système réalisé a naturellement des bornes de performances. Ainsi, il est destiné à être développé au fil du temps soit en optimisant le programme de contrôle, soit en ajoutant d'autres fonctionnalités dans la perspective d'un système plus complet. A l'aide de plusieurs simulations, nous avons réalisé des modèles expérimentaux des radars à impulsions qui sont les plus efficaces du point de vue de la précision des détections des cibles. Nous avons adapté ces modèles sur un robot mobile pour pouvoir traiter simultanément plusieurs cibles de l'environnement du robot, ajouté également des améliorations : le contrôle du système par une application Android via un module Wifi, et pour assurer des véhicules autoguidés, nous avons équipé ce système par un sous-système basé sur des capteurs optoélectroniques capables de différencier la nuance de la couleur pour détecter et suivre une ligne.

RÉFÉRENCES

[1] Charaf-Eddine Souria. *Conception d'interfaces boitiers innovantes pour le radar automobile 77 GHz : Application à la conception optimisée d'une chaine de réception radar en boitier*. Université Paul Sabatier-Toulouse III, 2017. Français.

[2] Beatriz Amante Garcia. *Conception d'un radar d'aide à la conduite automobile utilisant un système discriminateur de fréquence type "six-port*. Thèse de doctorat, Décembre 2002.

[3] Yassine Ruichek. *Perception de l'environnement par stéréovision, Application à la sécurité dans les systèmes de transports terrestres*. Thèse de doctorat, Université Lille1 - Sciences et Technologies, 2005.

[4] Abderrazak El Harti. *Notions Fondamentales de Télédétection spatiale*. Faculté des Sciences et Techniques de Béni Mellal, Maroc.

[5] *LiDAR : Etat de l'art, application en perception de l'environnement pour le véhicule autonome*. lidar Rapport P6 2017 03, INSA Rouen.

[6] Ali Bazzi. *Contribution à la définition de forme d'ondes pour les radars d'aide à la conduite automobile*. Thèse de Doctorat, 2010 telb0137.

[7] Jean-François RÉCOCHÉ. *Radars et effet Doppler*. Dossier thématique n 5.

[8] Mohamed Riad KRATTOU. *Étude De La Détection Radar Dans Un Milieu Homogéne*. Mémoire, Université Abou Bekr Belkaid, Tlemcen Algérie, 2013.

[9] Franck Chebila. *Lecteur radar pour capteurs passifs à transduction radio fréquence*. Micro et nanotechnologies/Microélectronique, Institut National Polytechnique de Toulouse, 31 Mars 2011.

[10] Tobias Otto. *Principle of FMCW Radars*. Université de technologie de Delft, 2012.

[11] Vincent Lemonde. *Stéréovision embarquée sur véhicule : de l'auto-calibrage à la détection d'obstacles*. Institut National Des Sciences Appliquées de Toulouse, 10 novembre 2005.

[12] Mohamed ZAYED. *Véhicules Intelligents : Etude et développement d'un capteur intelligent de vision pour l'attelage virtuel*, Thèse de Doctorat, Université des Sciences et Technologies de Lille, 12 juillet 2005.

[13] William M.Haynes, CRC Handbook of Chemistry and Physics, vol.97, CRC Press/Taylor and Francis, 2016, 2652p. (ISBN 1498754287), *Speed of sound in dry air*, p.2433 (14-48).

[14] Simon Landrault (Eskimon) et Hippolyte Weisslinger (olyte). *PREMIERS PAS EN INFORMATIQUE EMBARQUÉE* . livre, Édition du 19 février 2016.

[15] Marcial Carrobles Maeso, Félix Rodriguez Garcia, Angel Martin Hernandez. *Automates et Robotique III*. Manuel de mécanique industrielle, Edition 2004.

[16] Julien Dell. *Radar de recul*. Article, Autonews, 29 Octobre 2007.

Identification des canaux BRAN avec la méthode du noyau et les méthodes adaptatives linéaires

R. FATEH
Laboratoire LIMATI
Faculté Polydisciplinaire
Université Sultan Moulay Sliman
Beni Mellal, Maroc
Email :Fateh.smi@gmail.com

S. SAFI
Laboratoire LIMATI
Faculté Polydisciplinaire
Université Sultan Moulay Sliman
Beni Mellal, Maroc
Email : safi.said@gmail.com

A. DARIF
Laboratoire LIMATI
Faculté Polydisciplinaire
Université Sultan Moulay Sliman
Beni Mellal, Maroc
Email : anouar.darif@gmail.com

***Résumé*—Dans cet article, nous présentons une étude comparative entre la méthode du noyau dans espace d'Hilbert à noyau reproduisant, et les algorithmes adaptatifs linéaires, à savoir l'algorithme du gradient stochastique (LMS), l'algorithme du gradient stochastique normalisé (NLMS) et l'algorithme récursive des moindres carrés (RLS), nous montrons que l'algorithme de filtrage adaptatif du noyau réduit l'erreur quadratique moyenne vers le bas, cela en adoptant les canaux à évanouissement très rapide appelé BRAN (Broadband radio access Network).**

***Mots-clés*—Identification, Canaux BRAN, Noyaux définis positifs, systèmes Hammerstein, RKHS, LMS, NLMS, RLS.**

I. INTRODUCTION

Les méthodes classiques d'identification séquentielle, sont une classe de techniques qui permettent de suivre les variations lentes dans le temps et d'estimer en ligne un modèle du procédé qui reproduit asymptotiquement le comportement d'entrées-sorties du système. Ces systèmes linéaires sont souvent utilisés. Ils ont trouvé une variété d'applications dans tous les domaines de la vie. Cependant, ces modèles sont limités en raison de cette hypothèse de linéarité qui est improbable dans la plupart des problèmes réels [1]. Les méthodes à noyaux constituent une classe de modèles qui étend intelligemment les méthodes linéaires au cas non linéaire.

Ces méthodes à noyau sont des techniques d'apprentissage automatique puissant qui présentent une architecture moins complexe et fournissent des solutions non linéaires avec un coût calculatoire [2], [3].

On va s'intéresser dans ce papier à une étude comparative de la méthode noyau et les méthodes adaptatives linéaire.

II. L'ARCHITECTURE DU SYSTÈME

Comme le montre la figure 1, la structure d'Hammerstein est utilisée pour modéliser des systèmes où l'entrée statique du système est non linéaire.

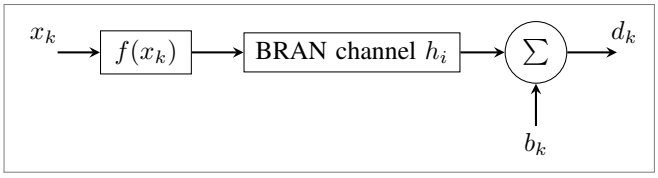

Fig. 1. Système Hammerstein [12]

Le système est décrit par l'équation entrée-sortie :

$$d_k = \sum_{i=0}^{N} h_i f(x_{k-i}) + b_k \tag{1}$$

Avec :

- x_k est le symbole émis par la source à l'instant k ;
- la réponse impulsionnelle $h_i, i = 1, 2, ...$, qui est d'une longueur finie. Nous supposons qu'il n'ya pas de retard dans le système, c'est-à-dire $h_0 \neq 0$;
- Le bruit additif b_k est gaussien et indépendant de l'entrée x_k et d_k ;

III. FILTRES ADAPTATIFS LINÉAIRES

L'objectif des filtres adaptatifs linéaires est d'identifier le modèle à partir des mesures bruyantes d'un système linéaire basé sur les critères de correction d'erreur.

A. Algorithme LMS

LMS [4], [7], est l'un des algorithmes les plus populaires pour le calcul des coefficients d'un filtre à réponse impulsionnelle finie. Cet algorithme a la propriété d'ajuster les coefficients d'un filtre pour réduire l'erreur entre le signal souhaité et sortie du filtre. Il est utilisé pour mettre à jour les poids du filtre adaptatif à chaque itération :

$$\theta(n+1) = \theta(n) + 2\mu_{LMS} e(n) x(n) \tag{2}$$

où $x(n) = [x(n), x(n-1), ..., x(n-M+1)]^\top$ est le vecteur de signal de référence de longueur M à l'instant n,

$\theta(n) = [\theta_0(n), \theta_1(n)...\theta_{M-1}(n)]^\top$ est le vecteur de poids, $e(n) = d(n) - y(n)$ est l'erreur entre le signal souhaité et la sortie du filtre et μ_{LMS} est appelé, le pas de convergence, sa valeur a un impact sur la performance de l'algorithme LMS. Afin d'assurer la convergence du vecteur de pondération, il est nécessaire que le pas de convergence soit compris dans l'intervalle ci-dessous :

$$0 < \mu_{LMS} < \frac{2}{\lambda}$$

avec λ représentant l'auto-corrélation des observations $x(n)$.

B. Algorithme NLMS

Le pas de convergence utilisé dans l'algorithme LMS est normalisé dans le cas des gradients stochastiques normalisé. Dans NLMS [5] la valeur de taille de pas pour le vecteur d'entrée est calculée :

$$\mu(n) = \frac{\beta}{\|x(n)\| + \epsilon} \tag{3}$$

Avec ϵ est le paramètre de contrôle et β est un pas normalisé $0 < \beta < 2$.

C. Algorithme RLS

L'algorithme récursive des moindres carrés converge plus rapidement mais est plus complexe que les algorithmes LMS et NLMS.
Dans cet algorithme [17], nous considérons ce qui suit :

$$\theta(n+1) = \theta(n) + G * e(n) \tag{4}$$

$$G(n) = \frac{\lambda^{-1}P(n-1)x(n)}{1 + \lambda^{-1}x^\top(i)P(n-1)x(n)} \tag{5}$$

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}G(n)x^\top(n)P(n-1) \tag{6}$$

où λ est le facteur d'oubli compris entre 0 et 1, $G(n)$ est le gain et $P(n)$ la matrice de covariance est initialisée de la manière suivante :

$$P(0) = \beta^{-1}I \tag{7}$$

Avec I est la matrice identité et β une constante positive très faible.

Le principal inconvénient de ces algorithmes est la dégradation de leurs performances lors de la résolution de problèmes non linéaires.

IV. Rappels du cadre théorique des noyaux

Dans le cadre standard d'apprentissage en ligne, chaque itération comprend plusieurs étapes, comme indiqué dans la figure 2. Au cours de la n−ime itération, un nouveau vecteur d'entrée se présente à l'entrée du modèle. Ensuite, il calcule la sortie estimée y_n correspondant à cette donnée, dans la mesure où le résultat final d_n est mis à disposition, ce qui permet à l'algorithme de calculer $e_n = d_n - y_n$, afin de mettre à jour sa solution.

Dans cette section, nous décrivons brièvement les principes de base des méthodes à noyau. Nous considérons l'espace des observations $\mathcal{X}$, auquel est associe le produit scalaire $\langle ., . \rangle$. Les détails sont disponibles dans des ouvrages plus spécialisés

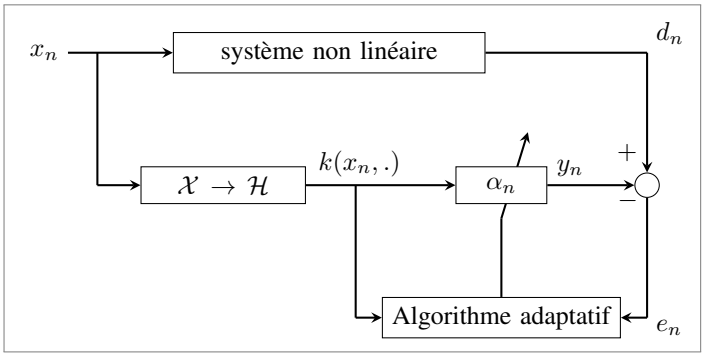

Fig. 2. Identification de système adaptatif basé sur le noyau [14].

que nous citons [9]–[12]. Un noyau est une fonction symétrique $k(x_i, x_j)$ de $\mathcal{X} \times \mathcal{X}$ dans $\mathbb{R}$. On rappelle la définition d'un noyau défini positif.

A. Noyau défini positif

Un noyau est appelé défini positif si, pour chaque point de données d'entrée $\{x_i\}_{i=1}^N \in \mathcal{X}$ satisfait à la condition suivante :

$$\sum_{i,j=1}^{N} a_i a_j k(x_i, x_j) \geq 0, \tag{8}$$

pour tout $N \in \mathbb{N}$, $x_1, ..., x_N \in \mathcal{X}$ et $a_1, ..., a_N \in \mathbb{R}$

B. Espace de Hilbert à noyau reproduisant (EHNR)

Soit $(\mathcal{H}, \langle ., . \rangle_\mathcal{H})$ un espace de Hilbert constitue par des fonctions de $\mathcal{X}$ dans $\mathbb{R}$. La fonction $k(x_i, x_j)$ de $\mathcal{X} \times \mathcal{X}$ dans $\mathbb{R}$ est le noyau reproduisant de $\mathcal{H}$, sous resserve que celui-ci en admette un, si et seulement si : la fonction $k(x, .) : x_i \longrightarrow k(x, x_j)$ appartient à $\mathcal{H}$, quel que soit $x \in \mathcal{X}$. Avec le but de construire un tel espace des fonctions, nous commencerons par définir la cartographie de fonction à partir de $\mathcal{X}$ dans l'espace des fonctions $\mathbb{R}$, pour un noyau défini positif k.

$$\begin{aligned} \phi \quad &: \mathcal{X} \longrightarrow \mathcal{H} \\ &x \longrightarrow k(x, .) \end{aligned} \tag{9}$$

La fonction $\phi(.) = k(., x)$ affecte la valeur $k(x, x')$ au point

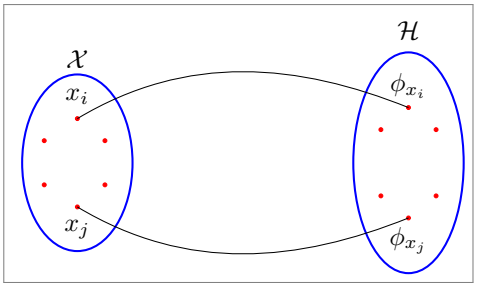

Fig. 3. Espace des données $\mathcal{X}$ et espace $\mathcal{H}$ induit par le noyau reproduisant k [13].

d'entrée x'. En interprétant la fonction du noyau comme une fonction de similarité, cette cartographie représente chaque point d'entrée x par sa similarité $k(x,.)$ à tous les autres points sur le domaine $\mathcal{X}$. Afin de construire un espace de caractéristiques associé à ϕ, l'image de ϕ doit être transformée en un espace vectoriel et dotée d'un produit scalaire [10]. Un espace vectoriel peut être défini en prenant des combinaisons linéaires de la forme :

$$f(.) = \sum_{i=1}^{N} \alpha_i k(x_i, .) \tag{10}$$

C. Théorème de représentation

Soient un espace non vide $\mathcal{X}$, K un noyau défini positif sur $\mathcal{X} \times \mathcal{X}$, un ensemble d'apprentissage donné $(x_1, y_1), ..., (x_N, y_N)$ avec $x_i \in \mathcal{X}$ l'ensemble des données et $y_i \in \mathbb{R}$ l'ensemble des sorties désirées, une fonction cout arbitraire $L : (\mathbb{R} \times \mathbb{R})^N \longrightarrow \mathbb{R}$ et une fonction réelle ϕ strictement monotone croissante sur $[0, \infty[$. Toute fonction $f \in \mathbb{R}$ minimisant la fonctionnelle de risque régularisée.

$$\underset{f \in \mathcal{H}}{min} J(f) = \frac{1}{N} \sum_{i=1}^{N} L((x_i, y_i, f(x_i)) + \phi(\|f\|_{\mathcal{H}}) \tag{11}$$

admet une représentation de la forme suivant :

$$f(x_k) = \sum_{i=1}^{N} \alpha_i k(x_i, x_k) \tag{12}$$

Dans la pratique, la plupart du temps, nous allons utiliser une ponctuelle moyenne fonction de perte carré L pour lesquels le problème de minimisation (11) écrit comme suit :

$$\underset{f \in \mathcal{H}}{min} J(f) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \phi(\|f\|_{\mathcal{H}}) \tag{13}$$

L'importance de ce théorème réside dans l'existence d'une solution unique à une fonctionnelle de coût régularisée, celle-ci pouvant s'exprimer comme un développement en série fini de fonctions noyau. La minimisation de cette fonction coût (11) se ramène à un problème d'optimisation à n dimensions, celui de la détermination des coefficients optimaux $\alpha_i \in \mathbb{R}$.

D. Algorithme d'identification basé sur le noyau

Cet algorithme appartient à la catégorie des algorithmes basés sur la régression du noyau. L'idée essentielle de la méthode du noyau réside dans la transformation de l'espace d'observation en un espace plus pertinent, le noyau gaussien étant le noyau le plus utilisé [10] :

$$k(x, y) = \exp(-\frac{\|x - y\|^2}{2\sigma^2}) \tag{14}$$

où σ est la largeur du noyau.
Le problème original d'optimisation s'écrit [20] :

$$J = \sum_{n=1}^{N} |d(n) - \phi(x(n))^\top \omega|^2 + c\omega^\top \omega, \tag{15}$$

Où c est une constante de régularisation de Tikhonov, le théorème représentatif IV-C stipule que ω peut être exprimé sous la forme d'une combinaison linéaire de données d'apprentissage transformées $\phi(x(n))$. Pour réduire la complexité de la méthode, nous utiliserons une séquence d'apprentissage des données d'entraînement pour représenter ω [20].

$$\omega = \sum_{i=1}^{M} \alpha_i \phi(x_i^j), \tag{16}$$

La combinaison des équations (16) et (15) conduit au problème matricielle suivant :

$$J = \|d - K\alpha\|^2 + c\alpha^\top K_j \alpha, \tag{17}$$

où K est la matrice du noyau, $d = [d_1, d_2, ..., d_N]^\top$ et $\alpha = (K^\top K + cK_j)^{-1} K^T d$
Comme indiqué à la section II, la sortie $d(n)$ du système hammerstein devient :

$$d(n) = h(n) * f(x(n)) + b(n) \tag{18}$$

Afin d'identifier les deux parties du système hammerstein ($h(n)$, $f(.)$), nous proposons de minimiser le coût suivant :

$$J = \|d - h * K\alpha\|^2 + c_\alpha \alpha^\top K_j \alpha + c_h h^\top h, \tag{19}$$

Avec c_α et c_h sont des constantes. Néanmoins, si une estimation des coefficients linéaires, $\hat{h}$, était disponible, il serait possible d'obtenir les coefficients correspondants $\hat{\alpha}$. Le terme $c_h h^\top h$ peut être éliminé par ce qu'il n'a aucune incidence sur la minimisation de l'équation (19). En faisant cela, la fonction du coût peut écrite comme, suit :

$$J_\alpha = \|d - K_h \hat{\alpha}\|^2 + c_\alpha \hat{\alpha}^\top K_j \hat{\alpha}, \tag{20}$$

De manière itérative, nous pouvons assurer la convergence de l'algorithme d'identification basée sur le noyau des systèmes de hammerstein, car il minimise à chaque fois la fonction de coût et qu'il alterne pour rechercher l'approximation de h en se concentrant sur les mises à jour des coefficients. L'algorithme du noyau qui peut identifier les paramètres du canal est le suivant [20] :

Algorithme 1 Identification basée sur le noyau

Début
Initialiser $\hat{h} = (\mathcal{X}^\top \mathcal{X} + c_h I)^{-1} \mathcal{X} d$
Tant que la fonction de coût J n'est pas convergé
  Mise à jour de :
  $K_h = \hat{h} * K$ et $\hat{\alpha} = (K_h^\top K_h + c_\alpha K_j)^{-1} K_h^\top d$
  $K_\alpha = \hat{\alpha} * K$ et $\hat{h} = (K_\alpha^\top K_\alpha + c_h I)^{-1} K_\alpha d$
Fin Tant que
Retourner $\hat{h}$.
Fin

tel que :

$$K_\alpha = \begin{pmatrix} k_\alpha(1) & 0 & \dots & 0 \\ k_\alpha(2) & k_\alpha(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ k_\alpha(N) & k_\alpha(N-1) & \dots & k_\alpha(N-P+1)) \end{pmatrix}$$

V. RÉSULTATS DE LA SIMULATION

La simulation va nous permettre d'étudier les résultats et les performances de algorithme de filtrage adaptatif de noyau. L'erreur quadratique moyenne sera utilisée pour mesurer la précision des valeurs estimées comme indiqué dans l'équation (21).

$$EQM(h,\hat{h}) = \frac{1}{p}\sum_{i=1}^{p}\left(\frac{h(i)-\hat{h}(i)}{h(i)}\right)^2 \tag{21}$$

Avec :

- $h(i)$: la réponse impulsionnelle réelle.
- $\hat{h}(i)$: la réponse impulsionnelle estimée.
- L'ordre p du modèle qui représente la longueur de la réponse impulsionnelle.

A. Canaux radio mobiles

Dans cette section, nous considérons des modèles pratiques de canaux BRAN (Broadband radio access Network). Ces modèles sont des canaux à évanouissement très rapide qui ont été normalisés par l'Institut européen de normalisation des télécommunications (ETSI [15]) dans le cadre du projet BRAN [6], [8]. Chaque modèle est composé de 18 trajets dont l'amplitude des retards suit une décroissance exponentielle.
L'équation (22) décrit la réponse impulsionnelle $h(n)$ du canal radio BRAN.

$$h(n) = \sum_{i=0}^{p} A_i \delta(n-\tau_i) \tag{22}$$

Où $\delta(n)$ est la fonction de Dirac, τ_i est le retard temporel du trajet i, A_i est l'amplitude du trajet i,$A_i \in N(0,1)$, $i = 1, ..., p$ et $p = 18$ est le nombre de trajets.
Dans les tableaux suivants, nous rapportons les valeurs de réponse impulsionnelle correspondant aux canaux radio mobiles BRAN.

TABLE I: Le modèle de canal BRAN A de l'ETSI.

Retard $\tau_i[ns]$	Amp. $A_i[dB]$	Retard $\tau_i[ns]$	Amp. $A_i[dB]$
0	0	90	-7.8
10	-0.9	110	-4.7
20	-1.7	140	-7.3
30	-2.6	170	-9.9
40	-3.5	200	-12.5
50	-4.3	240	-13.7
60	-5.2	290	-18
70	-6.1	340	-22.4
80	-6.9	390	-26.7

TABLE II: Le modèle de canal BRAN B de l'ETSI.

Retard $\tau_i[ns]$	Amp. $A_i[dB]$	Retard $\tau_i[ns]$	Amp. $A_i[dB]$
0	-2.6	230	-5.6
10	-3.0	280	-7.7
20	-3.5	330	-9.9
30	-3.9	380	-12.1
50	0	430	-14.3
80	-1.3	490	-15.4
110	-2.6	560	-18.4
140	-3.9	640	-20.7
180	-3.4	730	-24.6

TABLE III: Le modèle de canal BRAN C de l'ETSI.

Retard $\tau_i[ns]$	Amp. $A_i[dB]$	Retard $\tau_i[ns]$	Amp. $A_i[dB]$
0	-3.3	230	-3.0
10	-3.6	280	-4.4
20	-3.9	330	-5.9
30	-4.2	400	-5.3
50	0	490	-7.9
80	-0.9	600	-9.7
110	-1.7	730	-13.2
140	-2.6	880	-16.3
180	-1.5	1050	-21.2

TABLE IV: Le modèle de canal BRAN D de l'ETSI.

Retard $\tau_i[ns]$	Amp. $A_i[dB]$	Retard $\tau_i[ns]$	Amp. $A_i[dB]$
0	0.0	230	-9.4
10	-10.0	280	-10.8
20	-10.3	330	-12.3
30	-10.6	400	-11.7
50	-6.4	490	-14.3
80	-7.2	600	-15.8
110	-8.1	730	-19.6
140	-9.0	880	-22.7
180	-7.9	1050	-27.6

TABLE V: Le modèle de canal BRAN E de l'ETSI.

Retard $\tau_i[ns]$	Amp. $A_i[dB]$	Retard $\tau_i[ns]$	Amp. $A_i[dB]$
0	-4.9	320	0
10	-5.1	430	-1.9
20	-5.2	560	-2.8
40	-0.8	710	-5.4
70	-1.3	880	-7.3
100	-1.9	1070	-10.6
140	-0.3	1280	-13.4
190	-1.2	1510	-17.4
240	-2.1	1760	-20.9

B. Identification des méthodes :

Dans cette section, nous allons comparer les performances de l'algorithme d'identification basée sur le noyau (la régularisation de Tikhonov $c_a = 10^{-2}$ et $c_a = 10^{-2}$), à celles des algorithmes LMS (le paramètre de taille de pas $\mu = 0.03$), NLMS (le facteur de régularisation $\epsilon = 10^{-4}$, le paramètre de taille de pas $\mu = 0.05$) et RLS(le facteur d'oubli $\lambda = 0.95$, facteur de régularisation $\beta = 10^{-3}$), ceci en identifiant des canaux radio mobiles BRAN. Nous utilisons l'estimation de la fonction de référence $f(x) = \tanh(x)$, qui est souvent utilisée pour les applications de régression non linéaire.

C. Résultats de simulation : Canal BRAN A

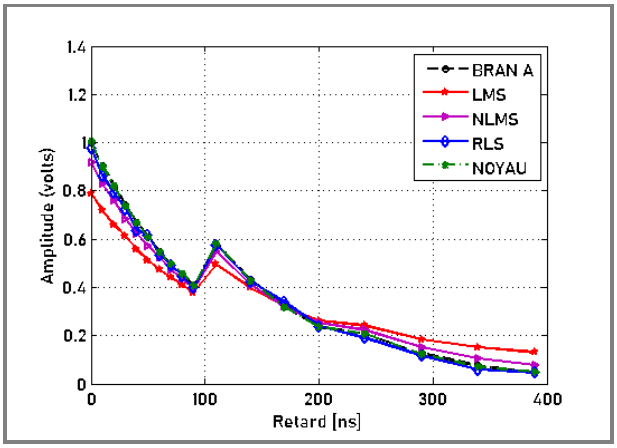

Fig. 4. Estimation des paramètres du canal BRAN A, en fonction des délais des trajets pour $N = 1000$, $SNR = 10dB$ et 50 itération de Monte Carlo.

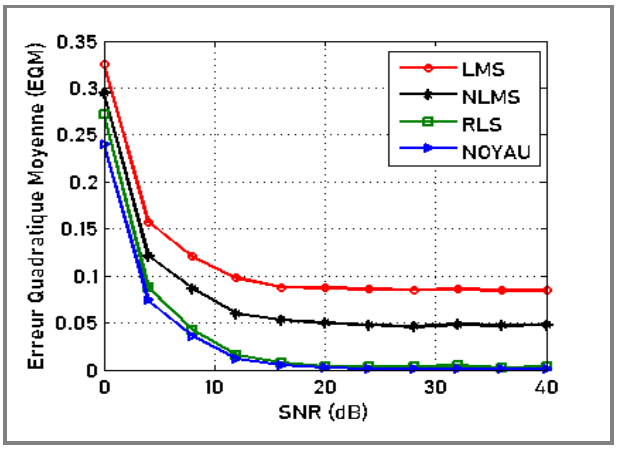

Fig. 5. EQM du canal BRAN A en fonction de SNR, pour $N = 1000$ et 50 itération de Monte Carlo.

1) Interprétation des résultats de simulation BRAN A: La figure 4 montre l'estimation des paramètres de la réponse impulsionnelle, du canal BRAN A, en fonction des délais des trajets, en utilisant les quatre algorithmes, pour un nombre d'échantillons $N = 1000$ et pour un $SNR = 10dB$. Nous relevons que les algorithmes (NOYAU) et (RLS) donne les meilleurs performances.

À partir de ce résultat, nous observons que les allures de l'amplitude estimées, via les algorithmes (NOYAU) et (RLS), ont la même forme que celles des données mesurées. Comparativement à l'algorithme (LMS), nous remarquons une grande différence entre l'allure de l'amplitude estimée et celles mesurées et avec une légère différence pour l'algorithme NLMS. Dans la figure 5, les valeurs de l'erreur quadratique moyenne pour les quatre algorithmes sont représentées, pour 1000 échantillons et différents SNR. Comme prévu, l'algorithme du noyau a les meilleures performances, suivies par RLS, NLMS et LMS.

D. Résultats de simulation : Canal BRAN B

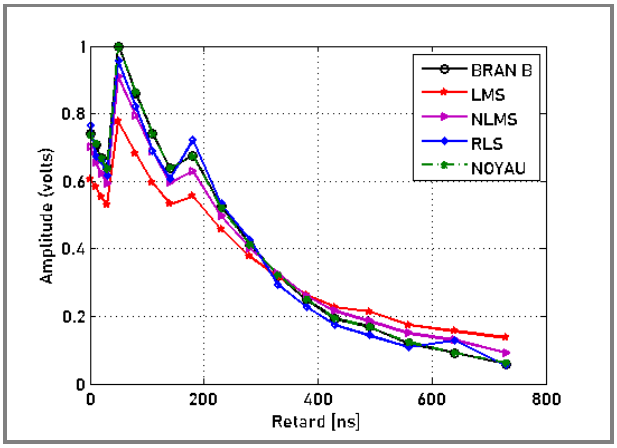

Fig. 6. Estimation des paramètres du canal BRAN B, en fonction des délais des trajets pour $N = 1000$, $SNR = 10dB$ et 50 itération de Monte Carlo.

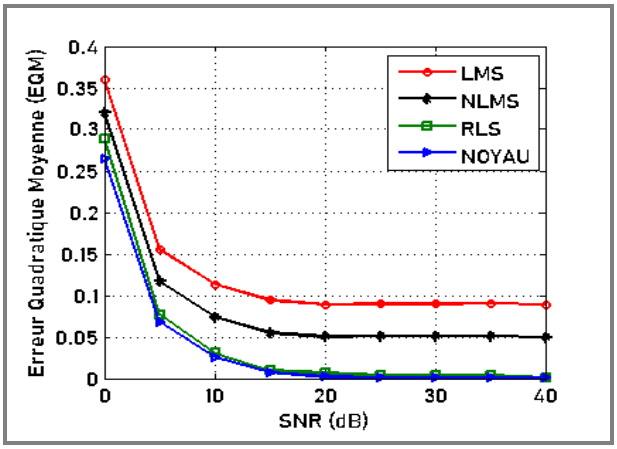

Fig. 7. Comparions des algorithmes en terme de l'EQM pour le BRAN B, $N = 1000$ et 50 itération de Monte Carlo.

1) Interprétation des résultats de simulation BRAN B: Sur la figure 6 nous représentons l'estimation de l'amplitude de la réponse impulsionnel du canal BRAN B, en utilisant les quatre algorithmes, pour un nombre d'échantillons $N = 1000$, un $SNR = 15db$ et 50 itérations de monte Carlo. Nous remarquons que la réponse estimée via l'algorithme du noyau possèdent la même allure que les valeurs réelles, alors que pour les autres algorithmes, nous avons une grande différence entre la réponse impulsionnelle estimée et mesurée.

À partir de la figure 7, On peut observer que la méthode du noyau donne des erreurs quadratiques moyennes assez faibles par rapport eux autres algorithmes.

E. Résultats de simulation : Canal BRAN C

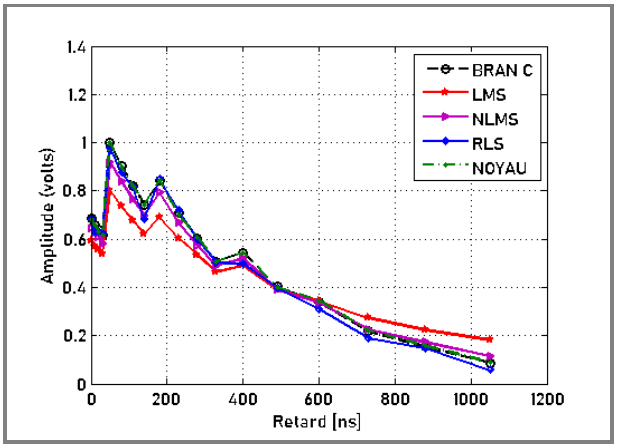

Fig. 8. Estimation des paramètres du canal BRAN C, en fonction des délais des trajets pour $N = 1000$, $SNR = 10dB$ et 50 itération de Monte Carlo.

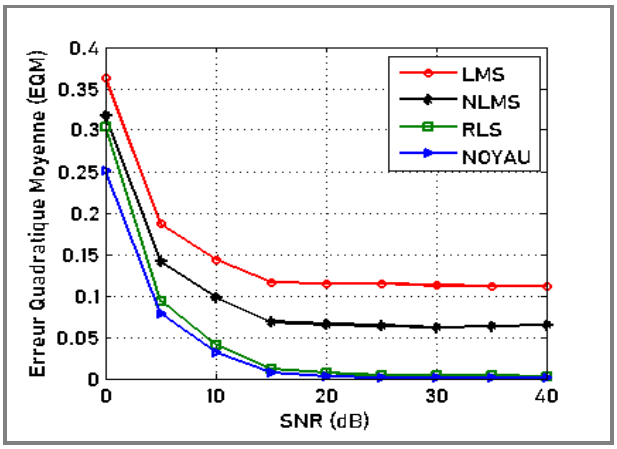

Fig. 9. Comparions des algorithmes en terme de l'EQM pour le BRAN B, $N = 1000$ et 50 itération de Monte Carlo.

1) Interprétation des résultats de simulation BRAN C: La figure 8 représente l'estimation de l'amplitude du canal BRAN C, en fonction des délais des trajets, en utilisant les quatre algorithmes, dans le cas bruité ($SNR = 10dB$), pour un nombre d'échantillons $N = 1000$. À partir de la figure 8, nous pouvons conclure que l'algorithme du noyau donne une bonne estimation de tous les paramètres de la réponse impulsionnelle du canal BRAN C. Si nous observons les valeurs estimées de la réponse impulsionnelle BRAN C, en utilisant l'algorithme RLS, nous remarquons, approximativement, les mêmes résultats donnés par l'algorithme du noyau sauf les quatre derniers paramètres. Concernant l'estimation de la réponse impulsionnelle du canal, à l'aide de l'algorithme NLMS, nous avons une différence mineure entre les valeurs estimées et les valeurs mesurées, et une différence apparente lors de l'utilisation de l'algorithme LMS.

La figure 9 montre les courbes de convergence (en termes de l'EQM) des quatre algorithmes.

F. Résultats de simulation : Canal BRAN D

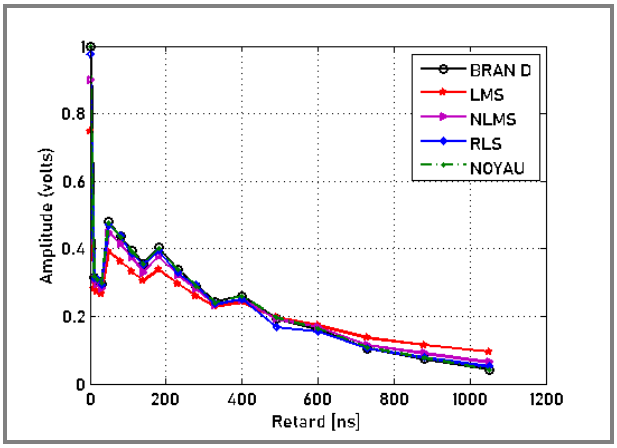

Fig. 10. Estimation des paramètres du canal BRAN D, en fonction des délais des trajets pour $N = 1000$, $SNR = 10dB$ et 50 itération de Monte Carlo.

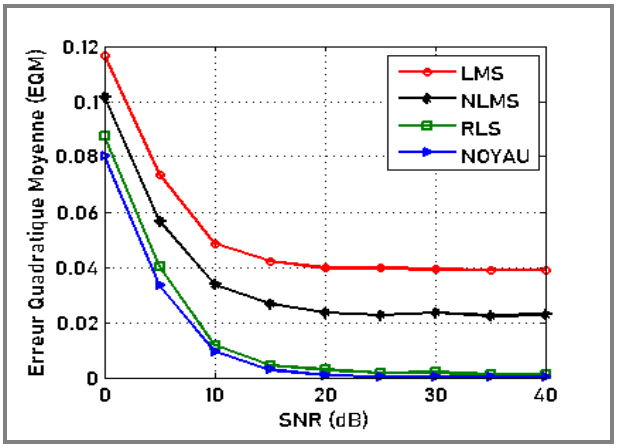

Fig. 11. Comparions des algorithmes en terme de l'EQM pour le BRAN B, $N = 1000$ et 50 itération de Monte Carlo.

1) Interprétation des résultats de simulation BRAN D: Les figures 10 et 11 présentent respectivement l'estimation des paramètres de la réponse impulsionnelle, du canal BRAN D, en fonction des délais des trajets, et les valeurs de l'erreur quadratique moyenne, pour un nombre d'échantillons $N = 1000$ et pour différents SNR. Il convient de noter que, les paramètres de la réponse impulsionnelle sont estimés avec une bonne précision à l'aide des algorithmes NOYAU, RLS et NLMS. Pour l'algorithme LMS nous avons une différence sur quelques trajets.

À partir de la figure 11, les valeurs du EQM fournies par l'algorithme du noyau sont très faibles, par rapport aux valeurs données par les algorithmes (LMS, RLS, NLMS), pour les

différents SNR, ce qui implique que les paramètres estimés sont très proche des valeurs réelles.

G. Résultats de simulation : Canal BRAN E

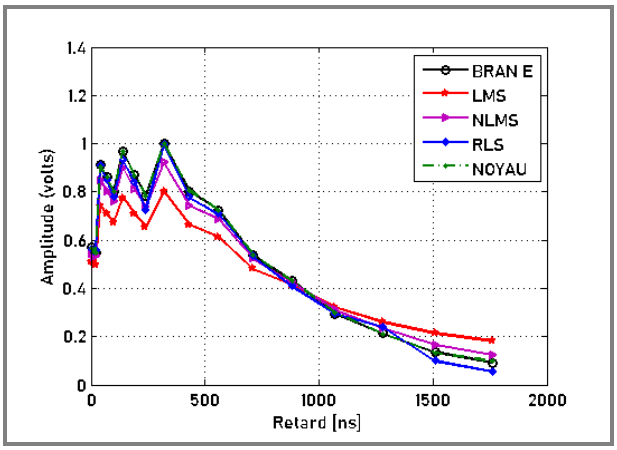

Fig. 12. Estimation des paramètres du canal BRAN E, en fonction des délais des trajets pour $N = 1000$, $SNR = 10dB$ et 50 itération de Monte Carlo.

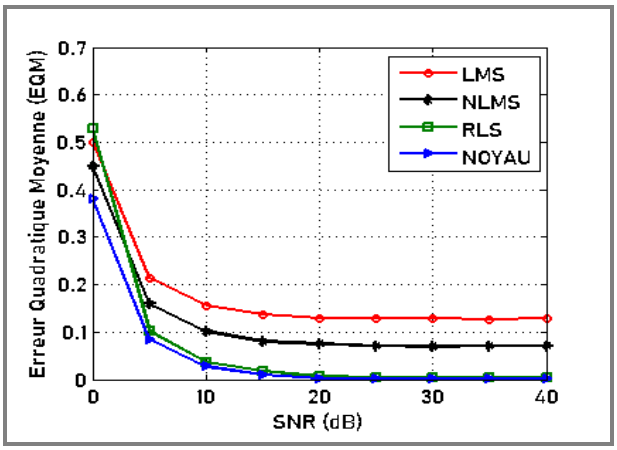

Fig. 13. Comparions des algorithmes en terme de l'EQM pour le BRAN B, $N = 1000$ et 50 itération de Monte Carlo.

1) Interprétation des résultats de simulation BRAN E: L'estimation de l'amplitude du canal BRAN E, à l'aide des quatre algorithmes est présentée sur la figure 12, pour une taille déchantions $N = 1000$ et pour un $SNR = 10dB$. Nous notons que la réponse estimée pour l'algorithme du noyau ressemble à celle des vrais, mais elle représente une certaine différence pour les algorithmes RLS, NLMS et LMS. D'après la figure 13, nous pouvons conclure que la méthode non linéaire surpasse considérablement les méthodes linéaires RLS, NLMS et LMS, quelle que soit le type de canal utilisé.

H. Influence des paramètres

Le but de cette section est d'analyser l'influence des paramètres N et SNR sur la fiabilité des estimations algorithmiques.

1) Effet du nombre d'échantillons N: Dans la figure 14 nous représentons la moyenne des paramètres estimés du canal BRAN E, en utilisant l'algorithme d'identification basée sur le noyau de système Hammerstein. Nous estimons les paramètres pour différents nombres d'échantillons ($N = 500$, $N = 1000$ et $N = 2000$), et pour un rapport $SNR = 10dB$.

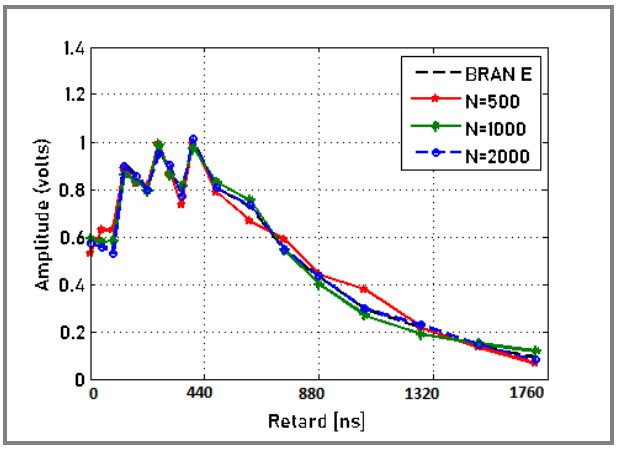

Fig. 14. Identification du canal BRAN E, SNR = 10dB et pour différents nombres d'échantillons N.

D'après la figure 14, on observe l'influence du nombre d'échantillons N sur la qualité de l'estimation. L'algorithme nécessite donc un grand nombre d'échantillons pour mieux s'approcher des valeurs réelles.

2) Influence du bruit: Pour savoir l'effet de SNR sur l'estimation du canal BRAN E, on a effectué un test avec SNR = 0, 08 et 30dB en fixant la taille d'échantillons à $N = 1000$. La figure suivant montre les valeurs moyennes des paramètres estimés, en utilisant l'algorithme du noyau.

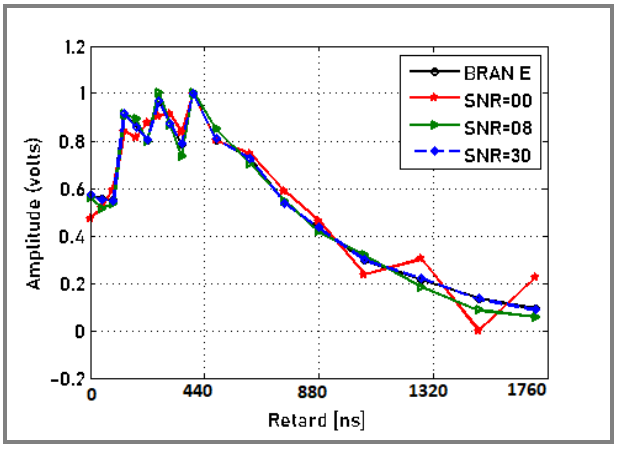

Fig. 15. Identification du canal BRAN E, N = 1000 et pour différents SNR.

De la figure 15, nous remarquons que, dans un environnement très bruyant $SNR = 0dB$, le bruit gaussien influencé les paramètres estimés du modèle, et une légère influence du bruit dans l'estimation des paramètres de réponse impulsionnelle

principalement si le $SNR < 08dB$, mais si le $SNR > 08dB$ les paramètres estimés sont très proches de la mesure.

VI. Conclusion

Dans cet article, nous avons étudié les performances de l'algorithme de filtrage adaptatif du noyau par rapport aux solutions adaptatives linéaires, tel que la méthode des moindres carrés récursive (RLS), la méthode du gradient stochastique normalisé (NLMS) et la méthode du gradient stochastique (LMS). Ces algorithmes ont été utilisés pour estimer les paramètres des canaux mesurés dans différents scénarios (BRAN A, B, C et E). En effet, l'algorithme de noyau devra donner une précision remarquable de l'estimation de l'amplitude des canaux BRAN comparativement aux algorithmes RLS, NLMS et LMS.

Références

[1] WJ.Jemaii, K.Abderrahim and M.Fouazi. *"Comparaison de deux méthodes adaptatives LMS & RLS du modèle de Volterra"*, Sciences and Techniques of Automatic control STA'2006, December 17-19, 2006, Tunisia.

[2] J. Shawe-Taylor and N. Cristianini.*"Kernel Methods for Pattern Analysis"*. Cambridge University Press, 2004.

[3] Maya Kallas.*"Méthodes à noyaux en reconnaissance de formes, prédiction et classification. Applications aux biosignaux"*. Sciences de l'ingénieur. Université de Technologie de Troyes, 2012.

[4] Simon Haykin :*"Filter Theory"*, Prentice Hall, 2002.

[5] Behrouz Farhang-Boroujeny : *"ADAPTIVE FILTERS THEORY AND APPLICATIONS"*, Second Edition, University of Utah USA, 2013.

[6] ETSI. Broadband radio acces network (bran) ; high performance radio local area network (hyperlan] type 2 ; physical layer. Technical report, European Telecommunications Standards Institute, December 2001.

[7] S. Safi, M. Frikel, M. Pouliquen, I. Badi, Y. Khmou and M. Boutalline. *"MC-CDMA System Identification and Equalization Using the LMS Algorithm and Takagi-Sugeno Fuzzy System"*. August 2014.

[8] ETSI, Broadband Radio Acces Network (BRAN) ; High PErformance Radio Local Area Network (HYPERLAN] Type 2 ; Requirements and archtiectures for wireless broadband access. Janvier 1999.

[9] N. Aronszajn. *"Theory of reproducing kernels"*. Transactions of the American Mathematical Society, 1950.

[10] B. Scholkopf and A.J. Smola. *"Learning with Kernels"* : Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive computation and machine learning. MIT Press, 2002.

[11] Bernhard Schölkopf, Ralf Herbrich and Alexander J. Smola. *"A generalized representer theorem"*. In Computational learning theory, 2001.

[12] Steven Van Vaerenbergh. *"Kernel Methods for Nonlinear Identification, Equalization and Separation of Signals"*, Universidad de Cantabria, 2009.

[13] Pantelis Bouboulis. *"Online learning in Reproducing Kernel Hilbert Spaces"*. 1 mai, 2012.

[14] Wemerson D. Parreira, Jose Carlos M. Bermudez, Cédric Richard, and Jean-Yves Tourneret. *"Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm"*. IEEE Transactions on Signal Processing, 2012.

[15] ETSI European Telecommunications Standards Institude

[16] Arnaud Massiani.*"Prototyping of high-rate systems based on the combination of multi-carrier modulations, spread spectrum technique and multiple antennas modulations"*. Theses, INSA de Rennes, Nov 2005.

[17] A.H. Abdullah, M.I. Yusof, and S.R.M. Baki, *"Adaptive noise cancellation : a practical study of the least-mean square (LMS) over recursive least-square (RLS) algorithm"*, 2002 Student Corference on Rescarch and Developmcnt Proccedings. Shali Alam. Malaysia.

[18] S. Safi, M. Frikel, M. M'Saad and A. Zeroual, *"Blind Impulse Response Identification of frequency Radio Channels : Application to Bran A Channel"*, World Academy of Science, Engineering and Technology 9 2007.

[19] Said Safi, Miloud Frikel, Boubekeur Targui, Estelle Cherrier, Mathieu Pouliquen, Mohammed M'Saad, *"Identification and equalization of MC-CDMA system driven by stochastic chaotic code"*,18th IEEE Mediterranean Conference on Control and Automation, MED'10, Jun 2010, Marrakech, Morocco. 2010.

[20] Mohammed Boutalline, Belaid Bouikhalene, and Said Safi. *"Channel identification and equalization based on kernel methods for downlink multicarrier-cdma systems"*. Journal of Electronic Commerce in Organizations, 2015.

Un nouvel algorithme d'extraction de motifs à partir d'une série de données temporelles

Abdelhak Goudjil, Mathieu Pouliquen, Eric Pigeon, Olivier Gehan
University of Caen - 14050 Caen Cedex, France

Résumé: Le problème de la découverte de motifs à partir d'une série de données temporelles est un problème classique de la fouille des données. Il s'agit d'un processus d'exploration et d'analyse dont le but est de trouver des structures originales et d'extraire des informations significatives. Dans ce papier, nous présentons un algorithme d'extraction de motifs à partir de données temporelles basé sur des techniques de classification non supervisée en justifiant certains choix techniques. L'algorithme permet d'extraire des motifs et de construire une bibliothèque de motifs sans connaissance a priori sur la nature et le nombre de motifs.

Mots clés: Motif, série temporelle, classification non supervisée, distance, classe.

I. INTRODUCTION

Le problème de l'extraction de motifs a reçu une attention remarquable ces dernières années et suscite actuellement une importante activité de recherche. L'extraction de motifs peut être réalisée sur des séries temporelles ([32]), des documents texte ([34], [14]), des graphes [18] ou des séquences vidéo [22]. Ses domaines d'application sont très variés, e.g., la médecine, la robotique, le traitement d'image, l'analyse de données ou encore la détection d'anomalies ([31], [4], [15] [23], [9]). L'objectif de ce papier est de présenter un algorithme d'extraction de motifs à partir d'une série de données temporelles.

Une série temporelle représente un phénomène ou une série d'évènements; elle résulte d'un processus de collecte d'informations ou de mesures. Nous nous focalisons sur les séries temporelles monovariables à valeur réelle pour une question de simplicité d'écriture et de présentation. Dans la suite nous utilisons la définition suivante:

Définition 1: Une série temporelle X est une suite ordonnée de N variables réelles:

$$X = \{x_1, \cdots, x_N\} \qquad x_k \in \mathbb{R} \tag{1}$$

Des données météorologiques, financières, de consommation domestique, de suivi de populations sont des exemples des séries temporelles considérées ici.

Dans le cadre de la recherche de motifs au sein d'une série temporelle nous ne nous intéressons pas aux propriétés globales de la série mais plutôt aux relations entre les sous-séquences de la série.

Définition 2: Soit X une série temporelle de taille N. Une sous-séquence S de X est une suite de taille $m \leq N$ constituée d'éléments de X à des instants de temps contigus

$$S = \{x_{k-m+1}, \cdots, x_k\}$$

avec $1+m \leq k \leq N$.
L'extraction de motifs à partir d'une série temporelle consiste à trouver toutes les sous-séquences qui apparaissent de manières récurrente à des instants distincts sur la série temporelle. Plus formellement, un motif peut être défini comme suit:

Définition 3: Soit une série temporelle X. Toute sous-séquence de X qui se produit à plusieurs reprises dans la série temporelle correspond à un motif.

Définition 4: Soit $S^{(1)}$, $S^{(2)}$, ...$S^{(p)}$, p sous-séquences de X correspondant à un même motif. La classe de motif est désignée par

$$C = \{S^{(1)}, \cdots, S^{(p)}\}$$

Du fait de la présence de bruit lors de la collecte d'informations, ou tout simplement parce que le phénomène observé présente certaines irrégularités, il est probable que des sous-séquences correspondant à un même motif ne soient pas identiques. Afin de prendre en compte ces imperfections lors de l'extraction de motifs il est nécessaire d'utiliser les notions de distance entre sous-séquences et de distance entre classes de motif.

- La distance entre deux sous-séquences $S^{(1)}$ et $S^{(2)}$ est notée $d_{S^{(1)},S^{(2)}}$. Le calcul de la distance entre deux sous-séquences permet d'estimer le degré de similarité entre les deux sous-séquences. Si la distance $d_{S^{(1)},S^{(2)}}$ est faible alors les deux sous-séquences $S^{(1)}$ et $S^{(2)}$ sont considérées comme similaires. Elles représentent alors le même motif et peuvent être agrégées au sein de la même classe.
- La distance entre deux classes $C^{(1)}$ et $C^{(2)}$ est notée $D_{C^{(1)},C^{(2)}}$. Le calcul de la distance entre deux classes permet d'estimer le degré de similarité entre les deux classes. Si la distance $D_{C^{(1)},C^{(2)}}$ est faible alors les deux classes représentent le même motif et peuvent être agrégées au sein de la même classe.

Différentes distances, entre sous-séquences et entre classes, sont proposées dans la littérature. Le choix de telle ou telle distance conditionne le résultat de la recherche de motif. Le choix de la distance entre classe oriente par exemple la morphologie des classes résultantes de l'analyse.

Différents algorithmes sont disponibles dans la littérature pour l'extraction de motifs. Nous nous focalisons sur les méthodes de recherche de motifs à partir d'un ensemble de sous-séquences extraites via une fenêtre glissante à partir d'une seule série temporelle (Subsequence time series clustering). Trois aspects permettent de caractériser ces méthodes: le choix d'une mesure de similarité, le choix d'un modèle de classe et le choix d'un algorithme de classification.

Le choix d'une mesure de similarité conditionne d'une manière importante les performances des algorithmes d'extraction de motifs. Un choix approprié dépend de la caractéristique de la série temporelle, la longueur de la série et bien sûr de l'objectif attendu. Différentes mesures de similarité peuvent être utilisées [2],[8]. Les plus utilisées sont: la distance euclidienne et la distance DTW (Dynamic Time Warping). La distance Euclidienne est largement utilisée comme un outil de mesure de similarité entre des sous-séquences de même longueur dans les algorithmes d'extraction de motifs ([28], [6] [21], [24]). La distance DTW est utilisée dans plusieurs algorithmes comme un outil de mesure de similarité entre des sous-séquences de longueurs différentes ([33], [29], [11]).

Un autre aspect qui permet de différencier les algorithmes d'extraction de motifs entre eux est le choix d'un modèle pour une classe de sous-séquences. Trouver un modèle de classe ou un représentant de classe est une étape essentielle dans les algorithmes d'extraction de motifs. En effet, la qualité des classes dépend de la qualité des modèles de classes. Plusieurs approches sont utilisées pour définir les modèles des classes: certains algorithmes utilisent la moyenne de toutes les sous-séquences de la classe comme étant le modèle de la classe ([5] et [25]). D'autres méthodes considèrent la medoïde de la classe comme étant le modèle de la classe ([10] et [13]). La medoïde d'une classe est la sous-séquence de la classe présentant la distance minimale avec l'ensemble des autres sous-séquences de la classe.

Outre la mesure de similarité et le modèle des classes, la majorité des algorithmes d'extraction de motifs reposent sur des méthodes de classification afin de regrouper des sous-séquences similaires au sein des classes de telle manière que les sous-séquences d'une même classe soient aussi proches les unes des autres que possible. Deux types de classification sont disponibles: la classification supervisée et la classification non supervisée. Il ne s'agit pas ici d'être exhaustif mais de présenter différents éléments permettant de nous orienter dans les choix techniques.

- L'objectif de la classification supervisée est de trouver des règles permettant de classer des sous-séquences dans des classes déjà identifiées et connues. Par la suite, ces règles permettent de répartir de nouvelles sous-séquences au sein des classes. Parmi les algorithmes d'extraction de motifs basés sur des méthodes de classification supervisée nous pouvons citer: [17], [19], [16]. Ces algorithmes supposent la connaissance a priori du nombre de motifs et de sous-séquences appartenant à ces motifs.
- La classification non supervisée a pour objectif de structurer et regrouper au sein d'une même classe des sous-séquences qui présentent des caractéristiques communes. L'idée est de grouper les sous-séquences dans des classes homogènes selon certains critères tout en favorisant l'hétérogénéité entre ces différentes classes. Plusieurs méthodes de classification non supervisée sont disponibles dans la littérature. Les plus utilisées sont: les méthodes de partitionnement [12] et les méthodes hiérarchiques [13]. Parmi les algorithmes d'extraction de motifs reposant sur des techniques de classification non supervisée, nous citons: [7], [20], [30], [26], [11]. Certains de ces algorithmes supposent la connaissance a priori du nombre de classes, d'autres non.

Pour un état de l'art détaillé sur les algorithmes d'extraction de motifs à partir des séries temporelles, le lecteur est invité se référer à ([35] et [1]).

Dans cet article, nous présentons un algorithme d'extraction de motifs à partir de données temporelles basé sur des techniques de classification non supervisée en justifiant certains choix techniques. L'algorithme permet d'extraire des motifs et de construire une bibliothèque de motifs sans connaissance a priori sur la nature et le nombre de motifs.

II. Objectifs et contraintes

Nos objectifs en terme d'extraction de motifs dans ce papier sont les suivants:

- déceler la présence éventuelle de motifs sur une série temporelle;
- réaliser la construction des classes de motif;
- modéliser le motif pour chaque classe;
- regrouper ces modèles dans une bibliothèque de motifs;

Les difficultés posées par ces objectifs sont multiples. Tout d'abord, le nombre de motifs présent sur la série temporelle est inconnu, il est même possible qu'aucun motif n'apparaisse. Il va donc être nécessaire d'estimer le nombre de motifs. De même, toutes les sous-séquences de la série temporelle ne correspondent pas forcément à un motif ou ne peuvent être inclues dans une classe de motifs. Il est probable que de nombreuses sous-séquences ne ressemblent à aucune autre sous-séquence. Ensuite, si au moins un motif est présent sur la série temporelle, ses instants d'occurence et sa longueur sont inconnus. Enfin, les séries temporelles doivent être analysées sans connaissance a priori sur la nature des informations qu'elles représentent. Il n'est donc pas possible d'utiliser de connaissance a priori (fréquence d'apparition, régularité des occurences, etc.) pour la réalisation des tâches précédentes. Rajoutons que ces différentes tâches doivent être mise en œuvre sous forme de briques logicielles écrites en C/C++ afin de pouvoir assurer un maximum de portabilités.

Nous proposons dans ce papier un algorithme, reposant sur des techniques de classification non supervisée, pour la recherche, l'identification et la localisation des motifs apparaissant sur une série de données temporelles. Il s'agit

aussi de regrouper ces motifs identifiés dans une bibliothèque de motifs.

III. Description de l'algorithme

A. Structure de l'algorithme

La recherche et la localisation des motifs se réalisent en mode batch. Il est nécessaire de récupérer l'ensemble des données avant de réaliser le traitement. L'algorithme nécessite en outre la disposition de données échantillonnées de manière régulière.

Les principaux points saillants de l'algorithme sont:

- une transformation de la série de données sous forme de sous-séquences de longueur m;
- une répartition des sous-séquences en classes de motifs;
- une validation des classes de motifs.

Les paragraphes ci-dessous détaillent les choix réalisés sur ces différents points.

1) Transformation de la série temporelle sous forme de sous-séquences de longueur m: Les instants d'occurrence des motifs sont inconnus, par conséquent toute sous-séquence peut a priori faire partie d'une classe de motif. La première étape de l'algorithme consiste donc à récupérer toutes les sous-séquences possibles, pour une longueur m fixée, en utilisant une fenêtre glissante. Ceci est illustré sur la figure 1. Il convient de noter qu'entre deux sous-séquences consécutives il y a chevauchement de $m-1$ échantillons, ceci devra être pris en compte lors de la formation des classes.

La longueur des motifs est inconnu a priori. Il est par conséquent nécessaire de réaliser l'analyse pour une gamme de longueur $m \in [m_{min}; m_{max}]$. Le choix de la longueur maximale m_{max} et la longueur minimale m_{min} dépend de la nature des données mais aussi des résultats attendus. Il est conseillé de chercher dans un premier temps les motifs de longue durée, c'est à dire de réaliser l'extraction de motif pour $m = m_{max}$ et ensuite de décrémenter la valeur de m jusqu'à $m = m_{min}$.

Pour chaque longueur m, une fois les classes de motif créées, il peut être nécessaire de supprimer de la série temporelle les sous-séquences appartenant à ces classes.

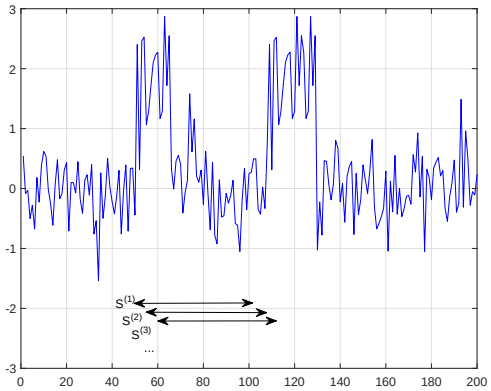

Fig. 1. Transformation de la série temporelle sous forme de sous-séquences

2) Répartition des sous-séquences en classes de motif: La répartition des sous-séquences en classes de motif est réalisée via une méthode de classification non-supervisée. Les méthodes de classification par partitionnement ne sont pas appropriées ici du fait de la nécessité pour ces méthodes de connaître a priori le nombre de classes de motifs.

L'utilisation des méthodes de classification hiérarchiques semble ici plus adaptée. L'idée est d'estimer dans un premier temps la similitude entre les différentes sous-séquences et de former l'arbre hiérarchique (figure 2). Cet arbre hiérarchique va nous permettre d'une part de localiser les "paquets" de sous-séquences similaires et d'autre part d'isoler les sous-séquences ne ressemblant à aucune autre sous-séquence.

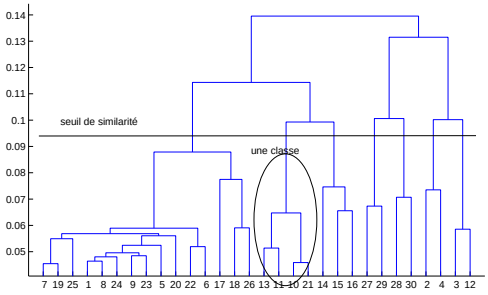

Fig. 2. Construction de l'arbre hiérarchique

Nous proposons de fixer un seuil de similarité afin de couper l'arbre hiérarchique d'une manière automatique. Ceci va assurer la création de classes constituées de sous-séquences similaires. Ceci va aussi permettre de ne pas avoir à construire nécessairement l'intégralité de l'arbre, d'où un gain de temps en terme de mise en œuvre.

Le choix de la classification hiérarchique nous contraint à deux choix pour la mise en œuvre: le premier sur le choix de la distance entre sous-séquences, le second sur le choix de la distance entre classes.

Choix de la distance entre sous-séquences

Étant donné qu'une des contraintes pour l'extraction de motifs est une mise en œuvre automatique, nous avons choisi une distance normalisée. Ceci nous permet de fixer par avance le seuil de similarité et non d'avoir à choisir de seuil au cas par cas, suivant l'amplitude des échantillons de la série temporelle.

Pour ce qui est du choix de la distance normalisée, nous proposons de favoriser l'usage d'une distance pour laquelle la charge de calcul est réduite. Les distances par cosinus et par corrélation permettent d'analyser des relations linéaires entre les sous-séquences et présentent des performances proches. La différence entre les deux réside dans le fait que la distance de corrélation est insensible à la présence d'un offset (centrage des données).

Choix de la distance entre classes

Le choix de la distance entre classes doit permettre de former des classes naturelles et homogènes. Les distances

entre classes les plus communes sont les suivantes: (single linkage, complete linkage, average linkage, weighted linkage, ward linkage, ...[3]). Étant donné que le critère de saut minimum ("single linkage") peut générer un effet de chaînage et qu'il est probable que certaines sous-séquences des classes formées soient éloignées les unes des autres, nous excluons ce critère des choix possibles. D'autre part, comme l'efficacité de différents critères d'agrégation dépend du volume et de la nature des données, les différents critères d'agrégation pour construire l'arbre hiérarchique doivent être évalués par le coefficient de corrélation cophenetic [27] via des simulations numériques sur les données de la série temporelle. Le critère qui donne un coefficient de corrélation le plus proche de 1 sera sélectionné. Ceci est illustré dans la prochaine section.

3) Validation des classes de motif: Chaque classe formée à l'étape précédente contient probablement des sous-séquences avec chevauchement. Il est aussi possible que certaines classes ne soient qu'artificielles dans le sens que les sous-séquences les constituants ne soient similaires les unes des autres que par coïncidence. Ainsi, il s'agit ici de prendre en compte uniquement les sous-séquences sans chevauchement dans une classe et d'exclure les classes dont le cardinal n'est pas significatif. La procédure est la suivante:

- Création d'un modèle pour chaque classe obtenue à l'étape précédente.
- Pour chaque classe, recherche sur la série temporelle des sous-séquences qui ont une forte similarité avec le modèle. Sont considérées comme similaires au modèle les sous-séquences dont la distance avec le modèle est inférieure au seuil utilisé dans l'étape de classification.
- Si deux sous-séquences se chevauchent, on ne garde que la sous-séquence qui présente la distance minimale avec le modèle.
- Élimination des classes dont le cardinal est inférieur à un seuil préfixé.

Le modèle d'une classe est une sous-séquence représentant au mieux la classe. Plusieurs options sont possibles pour la construction de ce modèle. Il peut être la moyenne de toutes les sous-séquences de la classe, la médiane ou encore la sous-séquences de la classe présentant la distance minimale avec l'ensemble des autres sous-séquences de la classe.

Le cardinal de la classe représente le nombre d'occurrences du motif sur la longueur de la série temporelle. Le seuil sur le cardinal d'une classe peut être lié à une fréquence minimale d'apparition du motif sur un intervalle de temps. Si le nombre de sous-séquences constituant une classe est supérieur à la fréquence d'apparition minimale f_{min}, on sauvegarde le motif dans la bibliothèque de motifs. Une fois qu'on a sauvegardé le motif, on supprime toutes les sous-séquences correspondantes à ce motif de la série de données. Si le nombre de sous-séquences sauvegardées pour une classe est inférieur à la fréquence d'apparition minimale f_{min}, le motif sera ignoré.

- ***Entrées: série de données*** T***,*** $seuil_{distance}$***,*** f_{min}***,*** m_{max} ***et*** m_{min}
- ***Sorties: Une bibliothèque de motifs***
- ***Début***
- ***(1): pour*** m ***décroissant*** $m_{min} \leq m \leq m_{max}$
- ***(2): Pour chaque longueur*** m***:***
  - ***(2-1): Transformer la série de données sous forme de sous-séquences en utilisant une fenêtre glissante.***
  - ***(2-2): Mesurer la similarité entre les sous-séquences en utilisant une distance normalisée".***
  - ***(2-3): Former les classes en utilisant la classification hiérarchique.***
  - ***(2-4): Les sous-séquences qui ont une corrélation très élevée entre elles (distance inférieure au*** $seuil_{distance}$***) sont réunies dans la même classe. Les classes constituées de peu d'individus ne sont pas prises en compte.***
  - ***(2-5): Un modèle de chaque classe est calculé.***
  - ***(2-6): Pour chaque classe, garder uniquement les sous-séquences sans chevauchement.***
  - ***(2-7): -Si le nombre de sous-séquences détectées d'un motif*** $\geq f_{min}$***:***
    - ***Sauvegarder le motif(modèle-longueur*** m ***- fréquence d'apparition - instant d'apparition)***
    - ***Supprimer toutes les sous-séquences du motif***
    - ***Aller chercher d'autres motifs avec d'autres fréquences d'apparition (s'il y en a) ou d'autres longueurs*** m***.***
  - ***(2-7): -Si le nombre de séquences détectées d'un motif*** $< f_{min}$
    - ***Ignorer la classe formée et sa modèle***
    - ***Aller chercher d'autres motifs avec d'autres fréquences d'apparition ou d'autres longueurs*** m
- ***Fin***

TABLE I
LE PSEUDO CODE DE L'ALGORITHME PROPOSÉ

B. Résumé de l'algorithme

Le résumé descriptif de l'algorithme est présenté dans le tableau I. On note que l'algorithme a des paramètres de synthèses qui doivent être spécifiés convenablement par l'utilisateur. Le choix de ces paramètres conditionne d'une manière importante les performances de l'algorithme. Ces paramètres de synthèses sont:

- La longueur maximale m_{max} et la longueur minimale m_{min} des motifs recherchés: Le choix de la longueur maximale m_{max} et la longueur minimale m_{min} dépend essentiellement de la nature et de l'amplitude des données, et dépend aussi des objectifs souhaités. L'algorithme cherche à identifier des motifs de longueur m telle que $m_{min} \leq m \leq m_{max}$.
- La fréquence d'apparition minimale f_{min}. La fréquence d'apparition minimale f_{min} permet d'élaguer les motifs non fréquents et par conséquent, éviter la création de motifs correspondant à des coïncidences.
- Le seuil de distance $seuil_{distance}$. Si le seuil distance fixé par l'utilisateur est élevé, des sous-séquences ne se ressemblant pas risque d'être inclues dans une même classe, c'est-à -dire assignées au même motif qui ne sera pas, représentatif d'un évènement. Si ce seuil est fixé trop proche de zéro, des sous-séquences représentant le même évènement, à un bruit de mesure prêt, risque d'être assignées à deux motifs différents. Ceci va générer un nombre de classes important avec un cardinal faible.

IV. Exemples de mise en œuvre

Afin d'illustrer certains aspects de la solution proposée, nous analysons les données de consommation d'eau d'un foyer sur plusieurs semaines. La figure 3 présente les données analysées sur une période de 10 semaines avec une période d'échantillonnage égale à 1 minute. Le nombre de données disponibles est alors $N = 100800$. Cette série de données correspond au cumul de différentes activités régulières (douche, bain, chasse d'eau, lave-linge, lave-vaisselle, etc). L'objectif est de permettre l'extraction automatique de ces activités.

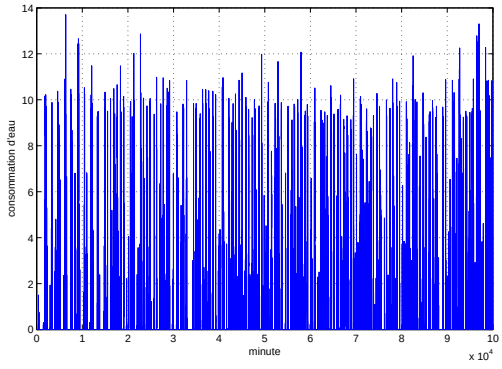

Fig. 3. Consommation en eau d'un foyer sur plusieurs semaines

Dans un premier temps, l'objectif est de choisir la meilleure distance entre sous-séquences, la meilleure distance entre classes et d'étudier l'évolution du temps de calcul (CPU time) de l'algorithme en fonction de nombre de données disponibles N. Deux distances entre sous-séquences peuvent être utilisées (distance par cosinus et distance par corrélation) et plusieurs distances entre classes peuvent être utilisées. Afin de sélectionner la meilleure distance entre sous-séquences et la meilleure distance entre classes, nous procédons à une simulation numérique en utilisant les différentes distances possibles. La qualité de ces distances est évaluée par le coefficient de corrélation cophénétique. Nous appliquons notre algorithme sur une période de 3 semaines ($N = 30240$) avec le paramétrage suivant:

- Fréquence d'apparition minimale $f_{min} = 3$.
- $m_{min} = 10$ minutes, $m_{max} = 10$ minutes
- $seuil_{distance} = 0.05$.

La longueur maximale m_{max} est choisi volontairement égale à la longueur minimale m_{min} car l'objectif de cette première simulation n'est pas la recherche de motifs, mais plutôt la sélection de la meilleure distance entre sous-séquences et la meilleure distance entre classes. La fréquence d'apparition minimale est choisie égale à $f_{min} = 3$, ceci signifie qu'on cherche des motifs qui apparaissent au moins une fois par semaine. Le calcul du coefficient de corrélation cophénétique et le temps nécessaire à l'exécution de l'algorithme pour les différentes distances possibles sont données dans le tableau II.

Il apparaît qu'avec les deux distances entres sous-séquences (distance par cosinus et distance par corrélation),

Distances entre sous-séquences	Distances entre classes	Corrélation cophénétique	Temps de calcul (sec)
Distance par cosinus	Average	0.7322	4.6251
	Centroid	0.6946	4.5788
	Complete	0.6029	4.4951
	Median	0.6273	4.5807
	Ward	0.6919	4.5212
	Weighted	0.6921	4.4819
Distance par corrélation	Average	0.7295	4.6962
	Centroid	0.6939	4.5436
	Complete	0.7022	4.5313
	Median	0.6130	4.5925
	Ward	0.7006	4.7127
	Weighted	0.6433	4.5116

TABLE II

Coefficient de corrélation "cophenetic" c et le temps de calcul pour les différents choix possibles

la distance entre classes "average" donne les meilleurs résultats. La qualité des résultats avec la distance par corrélation n'est pas trop dégradée par rapport à la qualité des résultats avec la distance par cosinus. Par contre, le temps nécessaire à l'exécution de l'algorithme avec la distance par corrélation est plus grand que le temps nécessaire à l'exécution de l'algorithme avec la distance par cosinus. Étant donné que les algorithmes hiérarchiques souffrent déjà de l'augmentation de la complexité de calcul d'une manière cubique en fonction du nombre de données, un choix d'une mesure de distance par corrélation est un handicap lors du traitement de séries de données de grand volume. Ceci nous oriente à choisir la distance par cosinus comme distance entre sous-séquences et le critère d'agrégation "average" comme distance entre classes.

Afin d'illustrer l'évolution du temps de calcul nécessaire à l'exécution de l'algorithme en fonction du nombre de données, nous procédons à plusieurs expériences pour différentes valeurs de N. Le temps d'exécution de l'algorithme pour différentes valeurs de N est donné dans le tableau III.

N	Temps d'exécution (sec)
$N = 20160$	2.1744
$N = 50400$	6.7369
$N = 70560$	11.2510
$N = 100800$	24.7646

TABLE III

Evolution du temps d'exécution en fonction de N

Selon les résultats donnés dans le tableau III, on voit clairement que le temps de calcul nécessaire à l'exécution de l'algorithme augmente d'une manière exponentielle avec l'augmentation du nombre de données. Ceci est dû à l'utilisation de la classification hiérarchique comme cœur de l'algorithme. A noter que le temps d'exécution de l'algorithme proposé dépend aussi de l'étendue de l'intervalle $[m_{min}, m_{max}]$. Plus l'étendue de l'intervalle est large, plus le temps d'exécution de l'algorithme est important.

Dans un second temps, l'objectif est d'extraire les motifs présents sur la série de données. Nous appliquons notre

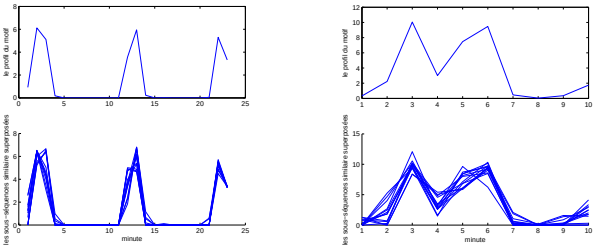

Fig. 4. Le modèle et les sous-séquences superposées de deux motifs identifiés

algorithme sur la série de données ($N = 100800$) avec le paramétrage suivant:

- Fréquence d'apparition minimale $f_{min} = 10$.
- $m_{min} = 10$ minutes, $m_{max} = 50$ minutes.
- $seuil_{distance} = 0.05$.

Nous cherchons des motifs de longueurs entre $[10\ ,\ 50]$ minutes avec une apparition au moins une fois par semaine. Figure 4 illustre un exemple de deux motifs identifiés par l'algorithme. L'observation a posteriori des instants d'apparition, le volume et la longueur de ces deux motifs révèlent qu'elles correspondent probablement à un lave-linge (motif de 25 minutes) et une douche (motif de 10 minutes).

V. CONCLUSION

Nous avons présenté un algorithme d'extraction de motifs à partir d'une série de données temporelles qui permet de rechercher, localiser des motifs et les retourner sous forme d'une bibliothèque de motifs. L'algorithme repose sur l'utilisation des techniques de classification non supervisée. Nous avons testé notre algorithme d'extraction de motifs sur des données réelles. Les résultats illustrent que l'algorithme permet bien d'extraire des motifs interprétables présents sur les séries de données.

REFERENCES

[1] S. Aghabozorgi, A. Shirkhorshidi, and T. Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015.

[2] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *International Conference on Foundations of Data Organization and Algorithms, Chicago, USA*, 1993.

[3] P. Berkhin. A survey of clustering data mining techniques. *Grouping multidimensional data*, pages 25–71, 2006.

[4] G. Bode, T. Schreiber, and M. Baranski. A time series clustering approach for building automation and control systems. *Applied energy*, 238:1337–1345, 2019.

[5] A. Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, Canada*, 2001.

[6] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[7] J. Ernst, G. Nau, and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21(suppl 1):i159–i168, 2005.

[8] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. *Fast subsequence matching in time-series databases*, volume 23. ACM, 1994.

[9] J. Gu, Z. Jiang, W. Fan, J. Wu, and J. Chen. Real-time passenger flow anomaly detection considering typical time series clustered characteristics at metro stations. *Journal of Transportation Engineering, Part A: Systems*, 146(4):04020015, 2020.

[10] V. Hautamaki, P. Nykanen, and P. Franti. Time-series clustering by approximate prototypes. In *19th International Conference on Pattern Recognition, Florida, USA*, 2008.

[11] H. Izakian, W. Pedrycz, and I. Jamal. Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 39:235–244, 2015.

[12] A. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[13] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[14] K. Kim, K. Jung, and J. Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1631–1639, 2003.

[15] N. Krishnan and D. Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.

[16] N. Krishnan and D. Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.

[17] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Journal of bioinformatics and computational biology*, 3(03):527–550, 2005.

[18] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings IEEE International Conference on Data Mining, San Jose, USA*, 2001.

[19] O. Lara and M. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[20] T. Liao. Clustering of time series dataa survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[21] E. Maharaj and P. DUrso. Fuzzy clustering of time series in the frequency domain. *Information Sciences*, 181(7):1187–1211, 2011.

[22] S. Mashtalir, M. Stolbovoi, and S. Yakovlev. Hybrid approach to clustering various lengths video. *Journal of Automation and Information Sciences*, 51(3), 2019.

[23] A. Mueen and N. Chavoshi. Enumeration of time series motifs of all lengths. *Knowledge and Information Systems*, 45(1):105–132, 2015.

[24] S. Nanda, B. Mahanty, and M. Tiwari. Clustering indian stock market data for portfolio management. *Expert Systems with Applications*, 37(12):8793–8798, 2010.

[25] F. Petitjean, A. Ketterlin, and P. Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[26] P. Rodrigues, J. Gama, and J. Pedroso. Hierarchical clustering of time-series data streams. *IEEE transactions on knowledge and data engineering*, 20(5):615–627, 2008.

[27] R. Sokal and F. J. Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40, 1962.

[28] M. Vlachos, J. Lin, E. Keogh, and D. Gunopulos. A wavelet-based anytime algorithm for k-means clustering of time series. In *proceedings Workshop on Clustering High Dimensionality Data and Its Applications, San Francisco, USA*, 2003.

[29] L. Wang, T. Gu, X. .Tao, and J. Lu. A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive and Mobile Computing*, 8(1):115–130, 2012.

[30] X. Wang, K. Smith, and R. Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13(3):335–364, 2006.

[31] Y. Wang, Y. Ru, and J. Chai. Time series clustering based on sparse subspace clustering algorithm and its application to daily box-office data analysis. *Neural Computing and Applications*, 31(9):4809–4818, 2019.

[32] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting time series motifs under uniform scaling. In *Proceedings of the 13th International Conference on Knowledge discovery and data mining, San Jose, USA*, 2007.

[33] X. Zhang, J. Liu, Y. Du, and T. Lv. A novel clustering method on time series data. *Expert Systems with Applications*, 38(9):11891–11900, 2011.

[34] N. Zhong, Y. Li, and S. Wu. Effective pattern discovery for text mining. *IEEE transactions on knowledge and data engineering*, 24(1):30–44, 2012.

[35] S. Zolhavarieh, S. Aghabozorgi, and Y. Teh. A review of subsequence time series clustering. *The Scientific World Journal*, 2014, 2014.

Un nouvel algorithme d'extraction de motifs à partir d'une série de données temporelles

Abdelhak Goudjil, Mathieu Pouliquen, Eric Pigeon, Olivier Gehan
University of Caen - 14050 Caen Cedex, France

Résumé: Le problème de la découverte de motifs à partir d'une série de données temporelles est un problème classique de la fouille des données. Il s'agit d'un processus d'exploration et d'analyse dont le but est de trouver des structures originales et d'extraire des informations significatives. Dans ce papier, nous présentons un algorithme d'extraction de motifs à partir de données temporelles basé sur des techniques de classification non supervisée en justifiant certains choix techniques. L'algorithme permet d'extraire des motifs et de construire une bibliothèque de motifs sans connaissance a priori sur la nature et le nombre de motifs.

Mots clés: Motif, série temporelle, classification non supervisée, distance, classe.

I. INTRODUCTION

Le problème de l'extraction de motifs a reçu une attention remarquable ces dernières années et suscite actuellement une importante activité de recherche. L'extraction de motifs peut être réalisée sur des séries temporelles ([32]), des documents texte ([34], [14]), des graphes [18] ou des séquences vidéo [22]. Ses domaines d'application sont très variés, e.g., la médecine, la robotique, le traitement d'image, l'analyse de données ou encore la détection d'anomalies ([31], [4], [15] [23], [9]). L'objectif de ce papier est de présenter un algorithme d'extraction de motifs à partir d'une série de données temporelles.

Une série temporelle représente un phénomène ou une série d'évènements; elle résulte d'un processus de collecte d'informations ou de mesures. Nous nous focalisons sur les séries temporelles monovariables à valeur réelle pour une question de simplicité d'écriture et de présentation. Dans la suite nous utilisons la définition suivante:

Définition 1: Une série temporelle X est une suite ordonnée de N variables réelles:

$$X = \{x_1, \cdots, x_N\} \qquad x_k \in \mathbb{R} \tag{1}$$

Des données météorologiques, financières, de consommation domestique, de suivi de populations sont des exemples des séries temporelles considérées ici.

Dans le cadre de la recherche de motifs au sein d'une série temporelle nous ne nous intéressons pas aux propriétés globales de la série mais plutôt aux relations entre les sous-séquences de la série.

Définition 2: Soit X une série temporelle de taille N. Une sous-séquence S de X est une suite de taille $m \leq N$ constituée d'éléments de X à des instants de temps contigus

$$S = \{x_{k-m+1}, \cdots, x_k\}$$

avec $1+m \leq k \leq N$.

L'extraction de motifs à partir d'une série temporelle consiste à trouver toutes les sous-séquences qui apparaissent de manières récurrente à des instants distincts sur la série temporelle. Plus formellement, un motif peut être défini comme suit:

Définition 3: Soit une série temporelle X. Toute sous-séquence de X qui se produit à plusieurs reprises dans la série temporelle correspond à un motif.

Définition 4: Soit $S^{(1)}$, $S^{(2)}$, ...$S^{(p)}$, p sous-séquences de X correspondant à un même motif. La classe de motif est désignée par

$$C = \{S^{(1)}, \cdots, S^{(p)}\}$$

Du fait de la présence de bruit lors de la collecte d'informations, ou tout simplement parce que le phénomène observé présente certaines irrégularités, il est probable que des sous-séquences correspondant à un même motif ne soient pas identiques. Afin de prendre en compte ces imperfections lors de l'extraction de motifs il est nécessaire d'utiliser les notions de distance entre sous-séquences et de distance entre classes de motif.

- La distance entre deux sous-séquences $S^{(1)}$ et $S^{(2)}$ est notée $d_{S^{(1)},S^{(2)}}$. Le calcul de la distance entre deux sous-séquences permet d'estimer le degré de similarité entre les deux sous-séquences. Si la distance $d_{S^{(1)},S^{(2)}}$ est faible alors les deux sous-séquences $S^{(1)}$ et $S^{(2)}$ sont considérées comme similaires. Elles représentent alors le même motif et peuvent être agrégées au sein de la même classe.
- La distance entre deux classes $C^{(1)}$ et $C^{(2)}$ est notée $D_{C^{(1)},C^{(2)}}$. Le calcul de la distance entre deux classes permet d'estimer le degré de similarité entre les deux classes. Si la distance $D_{C^{(1)},C^{(2)}}$ est faible alors les deux classes représentent le même motif et peuvent être agrégées au sein de la même classe.

Différentes distances, entre sous-séquences et entre classes, sont proposées dans la littérature. Le choix de telle ou telle distance conditionne le résultat de la recherche de motif. Le choix de la distance entre classe oriente par exemple la morphologie des classes résultantes de l'analyse.

Différents algorithmes sont disponibles dans la littérature pour l'extraction de motifs. Nous nous focalisons sur les méthodes de recherche de motifs à partir d'un ensemble de sous-séquences extraites via une fenêtre glissante à partir d'une seule série temporelle (Subsequence time series clustering). Trois aspects permettent de caractériser ces méthodes: le choix d'une mesure de similarité, le choix d'un modèle de classe et le choix d'un algorithme de classification.

Le choix d'une mesure de similarité conditionne d'une manière importante les performances des algorithmes d'extraction de motifs. Un choix approprié dépend de la caractéristique de la série temporelle, la longueur de la série et bien sûr de l'objectif attendu. Différentes mesures de similarité peuvent être utilisées [2],[8]. Les plus utilisées sont: la distance euclidienne et la distance DTW (Dynamic Time Warping). La distance Euclidienne est largement utilisée comme un outil de mesure de similarité entre des sous-séquences de même longueur dans les algorithmes d'extraction de motifs ([28], [6] [21], [24]). La distance DTW est utilisée dans plusieurs algorithmes comme un outil de mesure de similarité entre des sous-séquences de longueurs différentes ([33], [29], [11]).

Un autre aspect qui permet de différencier les algorithmes d'extraction de motifs entre eux est le choix d'un modèle pour une classe de sous-séquences. Trouver un modèle de classe ou un représentant de classe est une étape essentielle dans les algorithmes d'extraction de motifs. En effet, la qualité des classes dépend de la qualité des modèles de classes. Plusieurs approches sont utilisées pour définir les modèles des classes: certains algorithmes utilisent la moyenne de toutes les sous-séquences de la classe comme étant le modèle de la classe ([5] et [25]). D'autres méthodes considèrent la medoïde de la classe comme étant le modèle de la classe ([10] et [13]). La medoïde d'une classe est la sous-séquence de la classe présentant la distance minimale avec l'ensemble des autres sous-séquences de la classe.

Outre la mesure de similarité et le modèle des classes, la majorité des algorithmes d'extraction de motifs reposent sur des méthodes de classification afin de regrouper des sous-séquences similaires au sein des classes de telle manière que les sous-séquences d'une même classe soient aussi proches les unes des autres que possible. Deux types de classification sont disponibles: la classification supervisée et la classification non supervisée. Il ne s'agit pas ici d'être exhaustif mais de présenter différents éléments permettant de nous orienter dans les choix techniques.

- L'objectif de la classification supervisée est de trouver des règles permettant de classer des sous-séquences dans des classes déjà identifiées et connues. Par la suite, ces règles permettent de répartir de nouvelles sous-séquences au sein des classes. Parmi les algorithmes d'extraction de motifs basés sur des méthodes de classification supervisée nous pouvons citer: [17], [19], [16]. Ces algorithmes supposent la connaissance a priori du nombre de motifs et de sous-séquences appartenant à ces motifs.
- La classification non supervisée a pour objectif de structurer et regrouper au sein d'une même classe des sous-séquences qui présentent des caractéristiques communes. L'idée est de grouper les sous-séquences dans des classes homogènes selon certains critères tout en favorisant l'hétérogénéité entre ces différentes classes. Plusieurs méthodes de classification non supervisée sont disponibles dans la littérature. Les plus utilisées sont: les méthodes de partitionnement [12] et les méthodes hiérarchiques [13]. Parmi les algorithmes d'extraction de motifs reposant sur des techniques de classification non supervisée, nous citons: [7], [20], [30], [26], [11]. Certains de ces algorithmes supposent la connaissance a priori du nombre de classes, d'autres non.

Pour un état de l'art détaillé sur les algorithmes d'extraction de motifs à partir des séries temporelles, le lecteur est invité se référer à ([35] et [1]).

Dans cet article, nous présentons un algorithme d'extraction de motifs à partir de données temporelles basé sur des techniques de classification non supervisée en justifiant certains choix techniques. L'algorithme permet d'extraire des motifs et de construire une bibliothèque de motifs sans connaissance a priori sur la nature et le nombre de motifs.

II. Objectifs et contraintes

Nos objectifs en terme d'extraction de motifs dans ce papier sont les suivants:

- déceler la présence éventuelle de motifs sur une série temporelle;
- réaliser la construction des classes de motif;
- modéliser le motif pour chaque classe;
- regrouper ces modèles dans une bibliothèque de motifs;

Les difficultés posées par ces objectifs sont multiples. Tout d'abord, le nombre de motifs présent sur la série temporelle est inconnu, il est même possible qu'aucun motif n'apparaisse. Il va donc être nécessaire d'estimer le nombre de motifs. De même, toutes les sous-séquences de la série temporelle ne correspondent pas forcément à un motif ou ne peuvent être inclues dans une classe de motifs. Il est probable que de nombreuses sous-séquences ne ressemblent à aucune autre sous-séquence. Ensuite, si au moins un motif est présent sur la série temporelle, ses instants d'occurence et sa longueur sont inconnus. Enfin, les séries temporelles doivent être analysées sans connaissance a priori sur la nature des informations qu'elles représentent. Il n'est donc pas possible d'utiliser de connaissance a priori (fréquence d'apparition, régularité des occurences, etc.) pour la réalisation des tâches précédentes. Rajoutons que ces différentes tâches doivent être mise en œuvre sous forme de briques logicielles écrites en C/C++ afin de pouvoir assurer un maximum de portabilités.

Nous proposons dans ce papier un algorithme, reposant sur des techniques de classification non supervisée, pour la recherche, l'identification et la localisation des motifs apparaissant sur une série de données temporelles. Il s'agit

aussi de regrouper ces motifs identifiés dans une bibliothèque de motifs.

III. Description de l'algorithme

A. Structure de l'algorithme

La recherche et la localisation des motifs se réalisent en mode batch. Il est nécessaire de récupérer l'ensemble des données avant de réaliser le traitement. L'algorithme nécessite en outre la disposition de données échantillonnées de manière régulière.

Les principaux points saillants de l'algorithme sont:

- une transformation de la série de données sous forme de sous-séquences de longueur m;
- une répartition des sous-séquences en classes de motifs;
- une validation des classes de motifs.

Les paragraphes ci-dessous détaillent les choix réalisés sur ces différents points.

1) Transformation de la série temporelle sous forme de sous-séquences de longueur m: Les instants d'occurrence des motifs sont inconnus, par conséquent toute sous-séquence peut a priori faire partie d'une classe de motif. La première étape de l'algorithme consiste donc à récupérer toutes les sous-séquences possibles, pour une longueur m fixée, en utilisant une fenêtre glissante. Ceci est illustré sur la figure 1. Il convient de noter qu'entre deux sous-séquences consécutives il y a chevauchement de $m-1$ échantillons, ceci devra être pris en compte lors de la formation des classes.

La longueur des motifs est inconnu a priori. Il est par conséquent nécessaire de réaliser l'analyse pour une gamme de longueur $m \in [m_{min}; m_{max}]$. Le choix de la longueur maximale m_{max} et la longueur minimale m_{min} dépend de la nature des données mais aussi des résultats attendus. Il est conseillé de chercher dans un premier temps les motifs de longue durée, c'est à dire de réaliser l'extraction de motif pour $m = m_{max}$ et ensuite de décrémenter la valeur de m jusqu'à $m = m_{min}$.

Pour chaque longueur m, une fois les classes de motif créées, il peut être nécessaire de supprimer de la série temporelle les sous-séquences appartenant à ces classes.

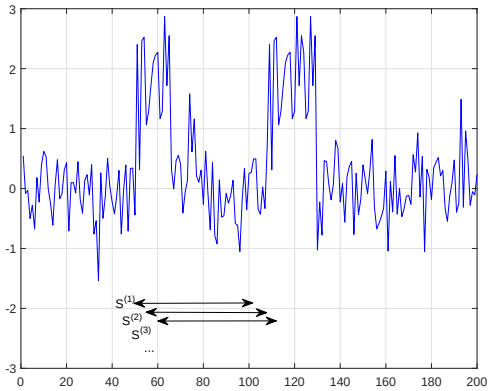

Fig. 1. Transformation de la série temporelle sous forme de sous-séquences

2) Répartition des sous-séquences en classes de motif: La répartition des sous-séquences en classes de motif est réalisée via une méthode de classification non-supervisée. Les méthodes de classification par partitionnement ne sont pas appropriées ici du fait de la nécessité pour ces méthodes de connaître a priori le nombre de classes de motifs.

L'utilisation des méthodes de classification hiérarchiques semble ici plus adaptée. L'idée est d'estimer dans un premier temps la similitude entre les différentes sous-séquences et de former l'arbre hiérarchique (figure 2). Cet arbre hiérarchique va nous permettre d'une part de localiser les "paquets" de sous-séquences similaires et d'autre part d'isoler les sous-séquences ne ressemblant à aucune autre sous-séquence.

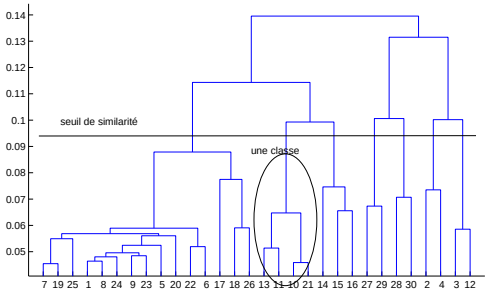

Fig. 2. Construction de l'arbre hiérarchique

Nous proposons de fixer un seuil de similarité afin de couper l'arbre hiérarchique d'une manière automatique. Ceci va assurer la création de classes constituées de sous-séquences similaires. Ceci va aussi permettre de ne pas avoir à construire nécessairement l'intégralité de l'arbre, d'où un gain de temps en terme de mise en œuvre.

Le choix de la classification hiérarchique nous contraint à deux choix pour la mise en œuvre: le premier sur le choix de la distance entre sous-séquences, le second sur le choix de la distance entre classes.

Choix de la distance entre sous-séquences

Étant donné qu'une des contraintes pour l'extraction de motifs est une mise en œuvre automatique, nous avons choisi une distance normalisée. Ceci nous permet de fixer par avance le seuil de similarité et non d'avoir à choisir de seuil au cas par cas, suivant l'amplitude des échantillons de la série temporelle.

Pour ce qui est du choix de la distance normalisée, nous proposons de favoriser l'usage d'une distance pour laquelle la charge de calcul est réduite. Les distances par cosinus et par corrélation permettent d'analyser des relations linéaires entre les sous-séquences et présentent des performances proches. La différence entre les deux réside dans le fait que la distance de corrélation est insensible à la présence d'un offset (centrage des données).

Choix de la distance entre classes

Le choix de la distance entre classes doit permettre de former des classes naturelles et homogènes. Les distances

entre classes les plus communes sont les suivantes: (single linkage, complete linkage, average linkage, weighted linkage, ward linkage, ...[3]). Étant donné que le critère de saut minimum ("single linkage") peut générer un effet de chaînage et qu'il est probable que certaines sous-séquences des classes formées soient éloignées les unes des autres, nous excluons ce critère des choix possibles. D'autre part, comme l'efficacité de différents critères d'agrégation dépend du volume et de la nature des données, les différents critères d'agrégation pour construire l'arbre hiérarchique doivent être évalués par le coefficient de corrélation cophenetic [27] via des simulations numériques sur les données de la série temporelle. Le critère qui donne un coefficient de corrélation le plus proche de 1 sera sélectionné. Ceci est illustré dans la prochaine section.

3) Validation des classes de motif: Chaque classe formée à l'étape précédente contient probablement des sous-séquences avec chevauchement. Il est aussi possible que certaines classes ne soient qu'artificielles dans le sens que les sous-séquences les constituants ne soient similaires les unes des autres que par coïncidence. Ainsi, il s'agit ici de prendre en compte uniquement les sous-séquences sans chevauchement dans une classe et d'exclure les classes dont le cardinal n'est pas significatif. La procédure est la suivante:

- Création d'un modèle pour chaque classe obtenue à l'étape précédente.
- Pour chaque classe, recherche sur la série temporelle des sous-séquences qui ont une forte similarité avec le modèle. Sont considérées comme similaires au modèle les sous-séquences dont la distance avec le modèle est inférieure au seuil utilisé dans l'étape de classification.
- Si deux sous-séquences se chevauchent, on ne garde que la sous-séquence qui présente la distance minimale avec le modèle.
- Élimination des classes dont le cardinal est inférieur à un seuil préfixé.

Le modèle d'une classe est une sous-séquence représentant au mieux la classe. Plusieurs options sont possibles pour la construction de ce modèle. Il peut être la moyenne de toutes les sous-séquences de la classe, la médiane ou encore la sous-séquences de la classe présentant la distance minimale avec l'ensemble des autres sous-séquences de la classe.

Le cardinal de la classe représente le nombre d'occurrences du motif sur la longueur de la série temporelle. Le seuil sur le cardinal d'une classe peut être lié à une fréquence minimale d'apparition du motif sur un intervalle de temps. Si le nombre de sous-séquences constituant une classe est supérieur à la fréquence d'apparition minimale f_{min}, on sauvegarde le motif dans la bibliothèque de motifs. Une fois qu'on a sauvegardé le motif, on supprime toutes les sous-séquences correspondantes à ce motif de la série de données. Si le nombre de sous-séquences sauvegardées pour une classe est inférieur à la fréquence d'apparition minimale f_{min}, le motif sera ignoré.

- ***Entrées: série de données*** T***,*** $seuil_{distance}$***,*** f_{min}***,*** m_{max} ***et*** m_{min}
- ***Sorties: Une bibliothèque de motifs***
- ***Début***
- ***(1): pour*** m ***décroissant*** $m_{min} \leq m \leq m_{max}$
- ***(2): Pour chaque longueur*** m***:***
  - ***(2-1): Transformer la série de données sous forme de sous-séquences en utilisant une fenêtre glissante.***
  - ***(2-2): Mesurer la similarité entre les sous-séquences en utilisant une distance normalisée".***
  - ***(2-3): Former les classes en utilisant la classification hiérarchique.***
  - ***(2-4): Les sous-séquences qui ont une corrélation très élevée entre elles (distance inférieure au*** $seuil_{distance}$***) sont réunies dans la même classe. Les classes constituées de peu d'individus ne sont pas prises en compte.***
  - ***(2-5): Un modèle de chaque classe est calculé.***
  - ***(2-6): Pour chaque classe, garder uniquement les sous-séquences sans chevauchement.***
  - ***(2-7): -Si le nombre de sous-séquences détectées d'un motif*** $\geq f_{min}$***:***
    - ***Sauvegarder le motif(modèle-longueur*** m ***- fréquence d'apparition - instant d'apparition)***
    - ***Supprimer toutes les sous-séquences du motif***
    - ***Aller chercher d'autres motifs avec d'autres fréquences d'apparition (s'il y en a) ou d'autres longueurs*** m***.***
  - ***(2-7): -Si le nombre de séquences détectées d'un motif*** $< f_{min}$
    - ***Ignorer la classe formée et sa modèle***
    - ***Aller chercher d'autres motifs avec d'autres fréquences d'apparition ou d'autres longueurs*** m
- ***Fin***

TABLE I
LE PSEUDO CODE DE L'ALGORITHME PROPOSÉ

B. Résumé de l'algorithme

Le résumé descriptif de l'algorithme est présenté dans le tableau I. On note que l'algorithme a des paramètres de synthèses qui doivent être spécifiés convenablement par l'utilisateur. Le choix de ces paramètres conditionne d'une manière importante les performances de l'algorithme. Ces paramètres de synthèses sont:

- La longueur maximale m_{max} et la longueur minimale m_{min} des motifs recherchés: Le choix de la longueur maximale m_{max} et la longueur minimale m_{min} dépend essentiellement de la nature et de l'amplitude des données, et dépend aussi des objectifs souhaités. L'algorithme cherche à identifier des motifs de longueur m telle que $m_{min} \leq m \leq m_{max}$.
- La fréquence d'apparition minimale f_{min}. La fréquence d'apparition minimale f_{min} permet d'élaguer les motifs non fréquents et par conséquent, éviter la création de motifs correspondant à des coïncidences.
- Le seuil de distance $seuil_{distance}$. Si le seuil distance fixé par l'utilisateur est élevé, des sous-séquences ne se ressemblant pas risque d'être inclues dans une même classe, c'est-à -dire assignées au même motif qui ne sera pas, représentatif d'un évènement. Si ce seuil est fixé trop proche de zéro, des sous-séquences représentant le même évènement, à un bruit de mesure prêt, risque d'être assignées à deux motifs différents. Ceci va générer un nombre de classes important avec un cardinal faible.

IV. Exemples de mise en œuvre

Afin d'illustrer certains aspects de la solution proposée, nous analysons les données de consommation d'eau d'un foyer sur plusieurs semaines. La figure 3 présente les données analysées sur une période de 10 semaines avec une période d'échantillonnage égale à 1 minute. Le nombre de données disponibles est alors $N = 100800$. Cette série de données correspond au cumul de différentes activités régulières (douche, bain, chasse d'eau, lave-linge, lave-vaisselle, etc). L'objectif est de permettre l'extraction automatique de ces activités.

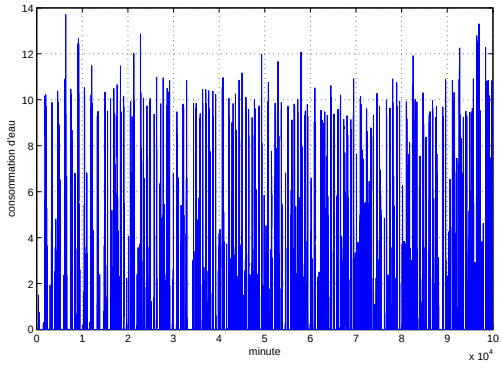

Fig. 3. Consommation en eau d'un foyer sur plusieurs semaines

Dans un premier temps, l'objectif est de choisir la meilleure distance entre sous-séquences, la meilleure distance entre classes et d'étudier l'évolution du temps de calcul (CPU time) de l'algorithme en fonction de nombre de données disponibles N. Deux distances entre sous-séquences peuvent être utilisées (distance par cosinus et distance par corrélation) et plusieurs distances entre classes peuvent être utilisées. Afin de sélectionner la meilleure distance entre sous-séquences et la meilleure distance entre classes, nous procédons à une simulation numérique en utilisant les différentes distances possibles. La qualité de ces distances est évaluée par le coefficient de corrélation cophénétique. Nous appliquons notre algorithme sur une période de 3 semaines ($N = 30240$) avec le paramétrage suivant:

- Fréquence d'apparition minimale $f_{min} = 3$.
- $m_{min} = 10$ minutes, $m_{max} = 10$ minutes
- $seuil_{distance} = 0.05$.

La longueur maximale m_{max} est choisi volontairement égale à la longueur minimale m_{min} car l'objectif de cette première simulation n'est pas la recherche de motifs, mais plutôt la sélection de la meilleure distance entre sous-séquences et la meilleure distance entre classes. La fréquence d'apparition minimale est choisie égale à $f_{min} = 3$, ceci signifie qu'on cherche des motifs qui apparaissent au moins une fois par semaine. Le calcul du coefficient de corrélation cophénétique et le temps nécessaire à l'exécution de l'algorithme pour les différentes distances possibles sont données dans le tableau II.

Il apparaît qu'avec les deux distances entres sous-séquences (distance par cosinus et distance par corrélation),

Distances entre sous-séquences	Distances entre classes	Corrélation cophénétique	Temps de calcul (sec)
Distance par cosinus	Average	0.7322	4.6251
	Centroid	0.6946	4.5788
	Complete	0.6029	4.4951
	Median	0.6273	4.5807
	Ward	0.6919	4.5212
	Weighted	0.6921	4.4819
Distance par corrélation	Average	0.7295	4.6962
	Centroid	0.6939	4.5436
	Complete	0.7022	4.5313
	Median	0.6130	4.5925
	Ward	0.7006	4.7127
	Weighted	0.6433	4.5116

TABLE II

Coefficient de corrélation "cophenetic" c et le temps de calcul pour les différents choix possibles

la distance entre classes "average" donne les meilleurs résultats. La qualité des résultats avec la distance par corrélation n'est pas trop dégradée par rapport à la qualité des résultats avec la distance par cosinus. Par contre, le temps nécessaire à l'exécution de l'algorithme avec la distance par corrélation est plus grand que le temps nécessaire à l'exécution de l'algorithme avec la distance par cosinus. Étant donné que les algorithmes hiérarchiques souffrent déjà de l'augmentation de la complexité de calcul d'une manière cubique en fonction du nombre de données, un choix d'une mesure de distance par corrélation est un handicap lors du traitement de séries de données de grand volume. Ceci nous oriente à choisir la distance par cosinus comme distance entre sous-séquences et le critère d'agrégation "average" comme distance entre classes.

Afin d'illustrer l'évolution du temps de calcul nécessaire à l'exécution de l'algorithme en fonction du nombre de données, nous procédons à plusieurs expériences pour différentes valeurs de N. Le temps d'exécution de l'algorithme pour différentes valeurs de N est donné dans le tableau III.

N	Temps d'exécution (sec)
$N = 20160$	2.1744
$N = 50400$	6.7369
$N = 70560$	11.2510
$N = 100800$	24.7646

TABLE III

Evolution du temps d'exécution en fonction de N

Selon les résultats donnés dans le tableau III, on voit clairement que le temps de calcul nécessaire à l'exécution de l'algorithme augmente d'une manière exponentielle avec l'augmentation du nombre de données. Ceci est dû à l'utilisation de la classification hiérarchique comme cœur de l'algorithme. A noter que le temps d'exécution de l'algorithme proposé dépend aussi de l'étendue de l'intervalle $[m_{min}, m_{max}]$. Plus l'étendue de l'intervalle est large, plus le temps d'exécution de l'algorithme est important.

Dans un second temps, l'objectif est d'extraire les motifs présents sur la série de données. Nous appliquons notre

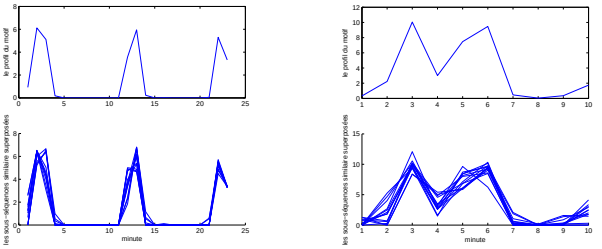

Fig. 4. Le modèle et les sous-séquences superposées de deux motifs identifiés

algorithme sur la série de données ($N = 100800$) avec le paramétrage suivant:

- Fréquence d'apparition minimale $f_{min} = 10$.
- $m_{min} = 10$ minutes, $m_{max} = 50$ minutes.
- $seuil_{distance} = 0.05$.

Nous cherchons des motifs de longueurs entre $[10\ ,\ 50]$ minutes avec une apparition au moins une fois par semaine. Figure 4 illustre un exemple de deux motifs identifiés par l'algorithme. L'observation a posteriori des instants d'apparition, le volume et la longueur de ces deux motifs révèlent qu'elles correspondent probablement à un lave-linge (motif de 25 minutes) et une douche (motif de 10 minutes).

V. CONCLUSION

Nous avons présenté un algorithme d'extraction de motifs à partir d'une série de données temporelles qui permet de rechercher, localiser des motifs et les retourner sous forme d'une bibliothèque de motifs. L'algorithme repose sur l'utilisation des techniques de classification non supervisée. Nous avons testé notre algorithme d'extraction de motifs sur des données réelles. Les résultats illustrent que l'algorithme permet bien d'extraire des motifs interprétables présents sur les séries de données.

REFERENCES

[1] S. Aghabozorgi, A. Shirkhorshidi, and T. Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015.

[2] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *International Conference on Foundations of Data Organization and Algorithms, Chicago, USA*, 1993.

[3] P. Berkhin. A survey of clustering data mining techniques. *Grouping multidimensional data*, pages 25–71, 2006.

[4] G. Bode, T. Schreiber, and M. Baranski. A time series clustering approach for building automation and control systems. *Applied energy*, 238:1337–1345, 2019.

[5] A. Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, Canada*, 2001.

[6] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[7] J. Ernst, G. Nau, and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21(suppl 1):i159–i168, 2005.

[8] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. *Fast subsequence matching in time-series databases*, volume 23. ACM, 1994.

[9] J. Gu, Z. Jiang, W. Fan, J. Wu, and J. Chen. Real-time passenger flow anomaly detection considering typical time series clustered characteristics at metro stations. *Journal of Transportation Engineering, Part A: Systems*, 146(4):04020015, 2020.

[10] V. Hautamaki, P. Nykanen, and P. Franti. Time-series clustering by approximate prototypes. In *19th International Conference on Pattern Recognition, Florida, USA*, 2008.

[11] H. Izakian, W. Pedrycz, and I. Jamal. Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 39:235–244, 2015.

[12] A. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[13] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[14] K. Kim, K. Jung, and J. Kim. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1631–1639, 2003.

[15] N. Krishnan and D. Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.

[16] N. Krishnan and D. Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.

[17] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Journal of bioinformatics and computational biology*, 3(03):527–550, 2005.

[18] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings IEEE International Conference on Data Mining, San Jose, USA*, 2001.

[19] O. Lara and M. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013.

[20] T. Liao. Clustering of time series dataa survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[21] E. Maharaj and P. DUrso. Fuzzy clustering of time series in the frequency domain. *Information Sciences*, 181(7):1187–1211, 2011.

[22] S. Mashtalir, M. Stolbovoi, and S. Yakovlev. Hybrid approach to clustering various lengths video. *Journal of Automation and Information Sciences*, 51(3), 2019.

[23] A. Mueen and N. Chavoshi. Enumeration of time series motifs of all lengths. *Knowledge and Information Systems*, 45(1):105–132, 2015.

[24] S. Nanda, B. Mahanty, and M. Tiwari. Clustering indian stock market data for portfolio management. *Expert Systems with Applications*, 37(12):8793–8798, 2010.

[25] F. Petitjean, A. Ketterlin, and P. Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[26] P. Rodrigues, J. Gama, and J. Pedroso. Hierarchical clustering of time-series data streams. *IEEE transactions on knowledge and data engineering*, 20(5):615–627, 2008.

[27] R. Sokal and F. J. Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40, 1962.

[28] M. Vlachos, J. Lin, E. Keogh, and D. Gunopulos. A wavelet-based anytime algorithm for k-means clustering of time series. In *proceedings Workshop on Clustering High Dimensionality Data and Its Applications, San Francisco, USA*, 2003.

[29] L. Wang, T. Gu, X. .Tao, and J. Lu. A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive and Mobile Computing*, 8(1):115–130, 2012.

[30] X. Wang, K. Smith, and R. Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13(3):335–364, 2006.

[31] Y. Wang, Y. Ru, and J. Chai. Time series clustering based on sparse subspace clustering algorithm and its application to daily box-office data analysis. *Neural Computing and Applications*, 31(9):4809–4818, 2019.

[32] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan. Detecting time series motifs under uniform scaling. In *Proceedings of the 13th International Conference on Knowledge discovery and data mining, San Jose, USA*, 2007.

[33] X. Zhang, J. Liu, Y. Du, and T. Lv. A novel clustering method on time series data. *Expert Systems with Applications*, 38(9):11891–11900, 2011.

[34] N. Zhong, Y. Li, and S. Wu. Effective pattern discovery for text mining. *IEEE transactions on knowledge and data engineering*, 24(1):30–44, 2012.

[35] S. Zolhavarieh, S. Aghabozorgi, and Y. Teh. A review of subsequence time series clustering. *The Scientific World Journal*, 2014, 2014.

SPECTRUM ESTIMATION FOR TIME-SERIES BASED ON BINARY DATA

Hicham Oualla[1,2]*, Mathieu Pouliquen*[1]*, Miloud Frikel*[1]*, Said Safi*[2]

[1]: Laboratoire d'Automatique de Caen - EA 7478
Normandie Univ, UNICAEN, ENSICAEN, LAC
Caen, France
[2]: Department of Mathematics and Informatics
University Sultan Moulay Slimane
Beni Mellal, Morocco

ABSTRACT

This paper focuses on the spectral analysis of time series. The samples of the time series are assumed to be unknown, we only know if samples are lower or higher than a given threshold. The paper proposes an algorithm for the estimation of the (power) spectrum, this algorithm considers first the estimation of the auto-correlation function of the time series. Simulation results are given to show the effectiveness of the proposed approach.

Index Terms— Spectral analysis, Binary time series

1. INTRODUCTION

In this paper we are interested in spectral analysis of time series in a particular context: we only know if samples are lower or higher than a given threshold. This corresponds to a one-bit quantization of the time-series. Such an analysis is interesting if binary valued sensors are employed (technological and economic constraints implying the use of a one-bit ADC for instance) or for the analysis of categorical data (shiny/cloudy or detected/not-detected for instance).

Similar analysis have already been considered in the literature. In [7] a Walsh-Fourier approach to the analysis of binary time series is proposed. The approach developed in [8] can also be considered because it allows a frequency domain analysis of categorical time series and a binary time series is a particular categorical time series. In [5] two spectral analysis for binary time series are compared: the square waves approach and the sinusoidal functions approach. It can be noticed that these approaches are only partial solutions to the problem considered in this paper. Indeed, these solutions focus on an analysis of the binary time series, they don't allow the spectral analysis of the primary time series and consequently the quality of the analysis is affected by the loss of information.

In order to have a more reliable spectral analysis we propose to realize the analysis of the primary time series. To this end we propose an algorithm for the estimation of its (power) spectrum. From the fact that the spectrum is the Fourier transform of the auto-correlation function, the first step of the algorithm consists in the estimation of the auto-correlation function using binary data.

The paper is organized as follows: the problem is formulated in Section 2. The proposed estimation algorithm is described in section 3. It has been tested on a numerical example, results are given in section 4. Section 5 concludes the paper.

2. PROBLEM STATEMENT

In this paper we consider a stationary stochastic process $\{y_t\}$. This corresponds to the primary time series. We want to analyze the frequency contents of this signal over a finite time interval, to this end we are interested in its (power) spectrum defined by

$$\Phi_y(f) = \sum_{\tau=-\infty}^{\infty} R_y(\tau) e^{-i\tau 2\pi f} \tag{1}$$

where $R_y(\tau) = \mathscr{E}\{y_t y_{t-\tau}\}$ is the auto-correlation function of $\{y_t\}$ of lag τ.

Samples of $\{y_t\}$ are assumed to be unknown, however we have access to the samples of the binary process $\{z_t\}$ defined by

$$z_t = \begin{cases} 1 \text{ if } \frac{y_t}{\sigma_y} \geq C \\ 0 \text{ else} \end{cases} \tag{2}$$

where C is a constant relative threshold which can be different from zero and $\sigma_y^2 = \mathscr{E}\{y_t^2\}$ is the variance. Roughly speaking, we know when $\frac{y_t}{\sigma_y}$ is superior or inferior to C. Fig. 1 illustrated the relation between $\{y_t\}$ and $\{z_t\}$.

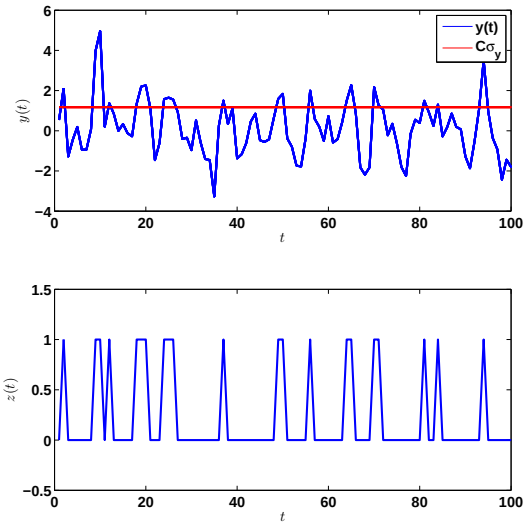

Fig. 1. Relation between $\{y_t\}$ and $\{z_t\}$

Objective: *The objective is to estimate the spectrum* $\Phi_y(f)$ *of* $\{y_t\}$ *using N samples of* $\{z_t\}$.

The following assumptions complete the description of the problem:

Assumption 1: $\{y_t\}$ is with normal distribution with a zero mean.

Assumption 2: σ_y is known.

3. ESTIMATION ALGORITHM

The algorithm proposed in this paper is a three steps algorithm. These three steps are detailed below.

- **Step 1**

  Define $\widehat{R}_z(\tau)$ the estimation of the auto-correlation function $R_z(\tau)$. In the first step $\widehat{R}_z(\tau)$ is computed, for $\tau \geq 0$, using samples $\{z_t\}_{t\in[1;N]}$ as follows:

$$\widehat{R}_z(\tau) = \frac{1}{N-\tau}\sum_{t=\tau+1}^{N} z_t z_{t-\tau} \tag{3}$$

  This estimate depends on N. For the sake of simplicity, we omit this dependence on N in the notation.

- **Step 2**

  Notice that $R_z(\tau)$ corresponds to

$$\begin{aligned} R_z(\tau) &= \mathscr{P}_r\{z_t = 1, z_{t-\tau} = 1\} \\ &= \mathscr{P}_r\{\tfrac{y_t}{\sigma_y} \geq C, \tfrac{y_{t-\tau}}{\sigma_y} \geq C\} \end{aligned} \tag{4}$$

  This means that $R_z(\tau)$ is the proportion of points $(y_t; y_{t-\tau})$ such that $\frac{y_t}{\sigma_y} \geq C$ and $\frac{y_{t-\tau}}{\sigma_y} \geq C$.

  In the following we use $\overline{R_x}(\tau) = \frac{R_x(\tau)}{\sigma_x^2}$ to denote the normalized auto-correlation function of lag τ. The previous proportion, denoted by $P_C\left(\overline{R_y}(\tau)\right)$ below, depends on C and $\overline{R_y}(\tau)$.

  From the fact that $\{y_t\}$ is with normal distribution with a zero mean, it follows that $P_C\left(\overline{R_y}(\tau)\right)$ depends on C and $\overline{R_y}(\tau)$ in the following manner:

$$P_C\left(\overline{R_y}(\tau)\right) = \int_C^{+\infty}\int_C^{+\infty} \psi(y_t, y_{t-\tau})\ dy_t dy_{t-\tau} \tag{5}$$

  where $\psi(y_t, y_{t-\tau})$ is the following distribution function

$$\psi(y_t, y_{t-\tau}) = \frac{1}{2\pi\sqrt{1-\left(\overline{R_y}(\tau)\right)^2}} e^{-\frac{y_t^2+y_{t-\tau}^2-2\left(\overline{R_y}(\tau)\right)y_t y_{t-\tau}}{2\left(1-\left(\overline{R_y}(\tau)\right)^2\right)}} \tag{6}$$

  For C given, $P_C\left(\overline{R_y}(\tau)\right)$ is a continuous monotone strictly increasing function of $\overline{R_y}(\tau)$, it is then possible to define the function $P_C^{-1}(.)$ such that

$$P_C^{-1}\left(P_C(x)\right) = x \tag{7}$$

  Define $\widehat{\overline{R_y}}(\tau)$ the estimation of the normalized auto-correlation $\overline{R_y}(\tau)$. The second step of the algorithm consists in computing $\widehat{\overline{R_y}}(\tau)$ from

$$\widehat{\overline{R_y}}(\tau) = P_C^{-1}\left(\widehat{R}_z(\tau)\right) \tag{8}$$

  Currently, there is no analytical expression for $P_C^{-1}(.)$, consequently in practice $\widehat{\overline{R_y}}(\tau)$ is computed minimizing the criterion:

$$\widehat{\overline{R_y}}(\tau) = argmin_{\overline{R_y}(\tau)}\left\{\left|\widehat{R}_z(\tau) - P_C\left(\overline{R_y}(\tau)\right)\right|\right\} \tag{9}$$

- **Step 3**

  The relation between the spectrum $\Phi_y(f)$ and $R_y(\tau)$ is given by (1).

  From the fact that we use a limited number of estimates $\widehat{\overline{R_y}}(\tau)$, the third step of the proposed algorithm consists in computing the estimation of $\Phi_y(f)$ as follows

$$\widehat{\Phi_y}(f) = \sigma_y^2 \sum_{\tau=-\gamma}^{\gamma} w_\gamma(\tau)\widehat{\overline{R_y}}(\tau)e^{-i\tau 2\pi f} \tag{10}$$

  with

$$w_\gamma(\tau) = \int_{-\pi}^{\pi} W_\gamma(\xi)e^{i\xi\tau}d\xi \tag{11}$$

  $W_\gamma(\xi)$ represents a window function and γ is the width of the window.

The choice of the window is not the subject of this paper, the default window is then the well-known Hamming window such that:

$$w_\gamma(\tau) = \begin{cases} \frac{1}{2}\left(1+cos\left(\frac{\pi\tau}{\gamma}\right)\right) & \text{if } |\tau| < \gamma \\ 0 \text{ else} \end{cases} \quad (12)$$

The proposed algorithm is summarized in Algorithm 1.

Algorithm 1: Estimation algorithm for $\Phi_y(f)$

input : n, $\{z_t\}_{t\in[1;N]}$

1- For $\tau \in [0;\gamma]$ compute $\widehat{R}_z(\tau)$ from

$$\widehat{R}_z(\tau) = \frac{1}{N-i}\sum_{t=i+1}^{N} z_t z_{t-\tau} \quad (13)$$

2- For $\tau \in [0;\gamma]$ compute $\widehat{\overline{R_y}}(\tau)$ from

$$\widehat{\overline{R_y}}(\tau) = argmin_{\overline{R_y}(\tau)}\left\{\left|\widehat{R}_z(\tau) - P_C\left(\overline{R_y}(\tau)\right)\right|\right\} \quad (14)$$

3- Compute $\widehat{\Phi_y}(f)$ from

$$\widehat{\Phi_y}(f) = \sigma_y^2 \sum_{\tau=-\gamma}^{\gamma} w_\gamma(\tau)\widehat{\overline{R_y}}(\tau)e^{-i\tau 2\pi f} \quad (15)$$

Remark 1 *It can be noticed that, from a result presented in [6], for $C = 0$ it is shown in [3] that $\overline{R_y}(\tau)$ can be written as a function of $R_z(\tau)$ as follows*

$$\overline{R_y}(\tau) = cos\left(\pi\left(1-2R_z(\tau)\right)\right) \quad (16)$$

This can be rewritten as

$$\overline{R_y}(\tau) = sin\left(2\pi\left(R_z(\tau) - \frac{1}{4}\right)\right) \quad (17)$$

This equality is used in [4], [2] and [9] for the estimation of $\overline{R_y}(\tau)$. (17) is equivalent to (8) for $C = 0$, i.e.

$$P_{C=0}^{-1}\left(R_z(\tau)\right) = sin\left(2\pi\left(R_z(\tau) - \frac{1}{4}\right)\right) \quad (18)$$

Remark 2 *In [1] an algorithm is proposed for the estimation of $\overline{R_y}(\tau)$ in the case of a threshold different from zero. The solution used in step 2 in the present paper is a normalized extension of the algorithm proposed in [1].*

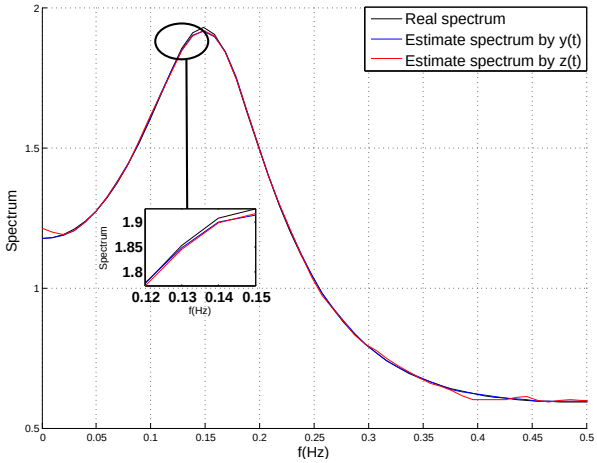

Fig. 2. Real spectrum and its estimations using $\{y_t\}$ and using $\{z_t\}$ for $N = 10^6$ and $C = 0$

4. NUMERICAL EXAMPLE

In this section, numerical simulations are used, in different situation, to illustrate performance of the proposed algorithm. Data are generated using (2) where $\{y_t\}$ is obtained with the following third-order AR process:

$$y_t = -0.5297y_{t-1} + 0.2685y_{t-2} + 0.1113y_{t-3} + n_t \quad (19)$$

where n_t is a zero mean white gaussian noise with unitary variance.

In a first experiment we compare the real spectrum and the estimated spectrum of y_t using a threshold $C = 0$ and $N = 10^6$ samples. Fig. 2 presents the real spectrum (obtained from the knowledge of the AR process), the estimated spectrum using $\{y_t\}$ and the estimated spectrum using $\{z_t\}$. Results show that the proposed algorithm, using $\{z_t\}$, works well.

In a second experiment we investigate the influence of C. 3 Monte-Carlo simulations are carried out with 100 runs for C from -1.5 to 1.5. In the first simulation we use $N = 10^4$, in the second $N = 3.10^4$ and the third $N = 5.10^4$. Performance of the algorithm is evaluated with the mean of the error on the spectrum:

$$Er = mean\sum_f |\Phi_y(f) - \widehat{\Phi_y}(f)| \quad (20)$$

Fig. 3 presents Er as a function of C for different value of N. Fig. 3 shows that the error Er depends on C. The best results are obtained with C not far from 0. If $|C|$ increases, then performance degrades. It can also be seen that this can be compensated with a higher value of N.

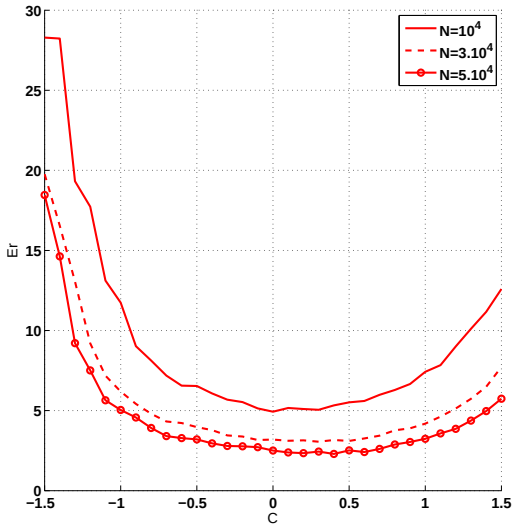

Fig. 3. Influence of the threshold C on the algorithm: $Er = mean \sum_f |\Phi_y(f) - \widehat{\Phi_y}(f)|$ as a function of $C \in [-1.5; 1.5]$, for $N = 10^4$, $N = 3.10^4$ and $N = 5.10^4$.

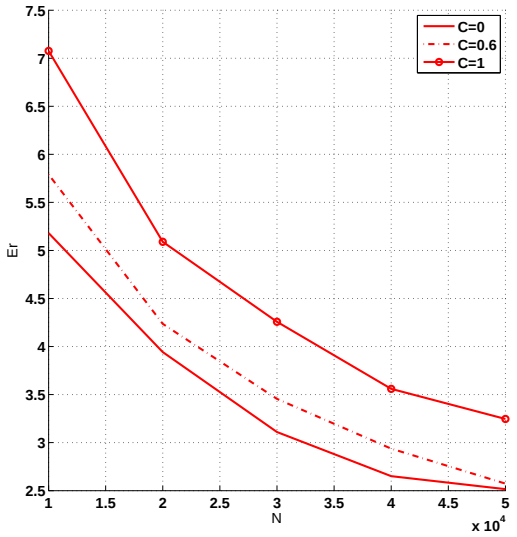

Fig. 4. Influence of the number of available data N on the algorithm: $Er = mean \sum_f |\Phi_y(f) - \widehat{\Phi_y}(f)|$ as a function of $N \in [10^4; 5.10^4]$, for $C = 0$, $C = 0.6$ and $C = 1$.

In a third experiment we confirm this phenomenon. 3 Monte-Carlo simulations are carried out with 100 runs for N from 10^4 to 5.10^4. Performance of the algorithm is evaluated with Er. Fig. 4 presents Er as a function of C for different value of C. It can be seen that Er decreases when N increases.

5. CONCLUSION AND FUTURE WORKS

In this paper a spectral analysis algorithm is proposed. This algorithm is designed for the analysis of time series based on binary data. The key step of the algorithm is the estimation of the auto-correlation function in a first time, the spectrum being estimated using Fourier transform in a second time. Numerical simulation results show the effectiveness of the proposed algorithm. Some suitable implementation conditions are given.

As future work, we may envision to extract other information such as higher order moments or to estimate the cross spectrum in the systems identification framework.

6. REFERENCES

[1] R. Auber, M. Pouliquen, E. Pigeon, M. M'Saad, O. Gehan, P.A. Chapon, and S. Moussay. Estimation of auto-regressive models for time series using binary or quantized data. *18th IFAC Symposium on System Identification, Stockholm*, 2018.

[2] A.J. Bagnall and G.J. Janacek. Clustering time series with clipped data. *Machine Learning*, 58:151–178, 2005.

[3] F.N. David. A note on the evaluation of the multivariate normal integral. *Biometrika*, 40:458–459, 1953.

[4] B. Kedem. Estimation of the parameters in stationary autoregressive processes after hard limiting. *Journal of The American Statistical Association*, 75:146–153, 1980.

[5] A. Kowalski, F. Musial, P. Enck, and K.T. Kalveram. Spectral analysis of binary time series: Square waves vs. sinusoidal functions. *Biological Rhythm Research*, 31:481–498, 2000.

[6] W. F. Sheppard. On the application of the theory of error to cases of normal distribution and normal correlation. *Philosophical Transactions of the Royal Society A*, 192:101–167, 1899.

[7] D.S. Stoffer and T. Panchalingam. A Walsh-Fourier approach to the analysis of binary time series. *Time Series and Econometric Modelling*, pages 147–163, 1987.

[8] D.S. Stoffer, D.E. Tyler, and A.J. McDougall. Spectral analysis for categorical time series: Scaling and the spectral envelope. *Biometrika*, 80:611–622, 1993.

[9] X. Zhen and I.V. Basawa. Estimation for binary models generated by gaussian autoregressive processes. *Journal of Statistical Computation and Simulation*, 80:1041–1051, 2010.

Un système de contrôle d'accès au parking automatisé basé sur RFID

E. ROUAN
Faculté des Sciences et Technique
Université Sultan Moulay Slimane
Email:rouanelhassania@gmail.com

S. SAFI
Faculté Polydisciplinare
Université Sultan Moulay Slimane
Email: safi.said@gmail.com

A. BOUMEZZOUGH
Faculté Polydisciplinare
Université Sultan Moulay Slimane
Email: ahmed.boumezzough@gmail.com

***Abstract*—Les majeurs systèmes de gestion de stationnement existants nécessitent des efforts humains que ce soit pour contrôler l'accès des véhicules ou l'enregistrement des véhicules autorisés dans des feuilles Excel ou sur papier, ce qui peut être plus difficile dans un immense parking et conduit à un certain nombre d'erreurs de stationnement telles que l'abandon des véhicules, l'accès par des personnes ne disposant pas de droits de stationnement, les fraudes au paiement des redevances de stationnement, la sécurité publique, congestion, etc ...**

Avec la technologie d'identification par radiofréquence, un système de stationnement intelligent peut être mis en place afin de réduire l'implication humaine, ce qui permet de minimiser les problèmes de stationnement. Les véhicules ne peuvent pas accéder au parking sans une étiquette RFID et la vérification à son entrée et à sa sortie peuvent être effectués rapidement en évitant le problème de congestion près des barrières du parking. Les utilisateurs n'auront pas besoin d'attendre l'identification de leurs véhicules car cela se fera automatiquement grâce aux étiquettes qui leur sont attachés. Ainsi, cela garantira également la sécurité, car seuls les véhicules enregistrés sont autorisés à accéder au parking.

Dans ce document, un système intelligent de contrôle d'accès au parking est proposé et modélisé en utilisant la technologie RFID combinée avec la carte NodeMCu V3.

Mots clés*-: *Contrôle d'accès, Automatique, Système de stationnement, Radio Frequency IDentification.

I. INTRODUCTION

Avec le développement rapide de l'économie mondiale et l'accélération de l'urbanisation, le nombre de véhicules est en augmentation, ce qui provoque de nombreux problèmes tels que les accidents de la route, la congestion, le gaspillage de carburant et d'énergie... etc. Les parkings jouent un rôle important pour les impacts sociaux et économiques dans le développement du pays [1].

La plupart des systèmes de stationnement existants nécessitent soit un équipement, une infrastructure et un déploiement coûteux, soit ils sont semi-automatisés nécessite des interventions humaines pour surveiller l'accès des véhicules, ce qui entraîne de nombreux problèmes de stationnement [2].

Récemment, des nombreuses recherches se sont concentrées sur l'identification par radiofréquence (Radio Frequency IDentification; RFID) pour résoudre les problèmes de stationnement. L'utilisation de cette technologie dans les systèmes de stationnement peut les rend entièrement automatiques avec des coûts beaucoup plus bas et avec une sécurité plus élevée. RFID est très utile pour authentifier rapidement et simultanément les véhicules qui se passent sur le parking. Dans un tel système, les véhicules sont équipés d'une étiquette RFID. Chaque véhicule entrant dans l'espace de stationnement a un code d'identification unique et la barrière ne s'ouvrira que si le véhicule est reconnu comme un immatriculé, ce qui rend l'espace de stationnement plus sécurisé.

Dans ce document, un système de contrôle d'accès automatique au parking basé sur RFID a été conçu à l'aide d'une carte NodeMCU V3 et du logiciel Arduino IDE. Le système développé combine la technologie RFID, en utilisant le module RFID RC522 qui agit comme un lecteur de étiquettes RFID et NodeMCU V3 pour accomplir la tâche requise. Le lecteur RFID permet la détection d'étiquettes RFID à l'entrée/sortie du parking. Il capture l'identifiant de l'utilisateur ID et le compare avec les IDs stockés pour une correspondance. Si l'ID capturé correspond à l'un des IDs stockés, l'accès est accordé; sinon l'accès est refusé.

Le reste de ce document est organisé comme suit: La section II fournit une brève description de la technologie RFID, y compris ses principaux composants et applications. La section III est consacrée à la description des matériels et des logiciels utilisés dans ce projet. La section VI décrit le prototype fonctionnel du système proposé. Finalement des conclusions sont tirées dans la section V.

II. BREVE DESCRIPTION DE LA TECHNOLOGIE RFID

Radio Frequency IDentification ou l'identification par radiofréquence est une technologie émergente qui est largement utilisée pour identifier des objets animés ou inanimés par des ondes radio, d'en suivre le cheminement ou d'en connaître ses caractéristiques à distance, dans une distance bien déterminée selon la fréquence utilisée et dans un minimum de temps [3].

RFID est une technologie de communication sans fil qui permet une communication sans contact et une identification d'objets étiquetés sans avoir besoin d'une ligne de vision directe, ce qui améliore l'efficacité de ses processus et sa facilité d'utilisation.

La technologie RFID a été inventée dans les années 1930-1940 pendant la seconde guerre mondiale mais elle était inconnue pour les applications commerciales jusqu'aux années 1980.

L'une de ses premières applications est un système Rader qui est utilisé pour identifier les avions alliés ou ennemis [4].

Aujourd'hui, la RFID joue un rôle important dans de nombreux domaines grâce à ses divers avantages et fonctionnalités tels que la gestion de la chaîne d'approvisionnement, y compris la production et la livraison de produits [5], le suivi des objets [6], l'inventaire [7], les systèmes de stationnement intelligents [2]... etc.

Un système RFID est composé principalement de [8]:

- Lecteurs ou interrogateurs.
- Tags ou étiquettes intelligentes.
- Middleware ou intergiciel.

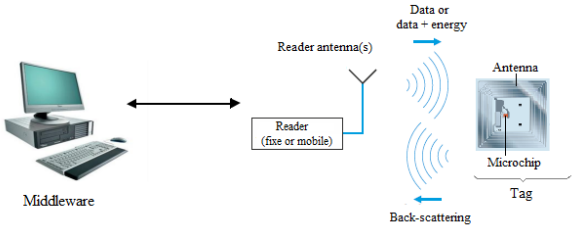

Fig. 1. Les éléments principaux d'un système RFID.

Les lecteurs RFID peuvent identifier les tags attachés à un objet, un animal ou une personne grâce à des ondes radio émises et reçues par ses antennes. Ce dispositif permet d'activer, si nécessaire, et d'interroger un tag qui passe à proximité de sa zone d'interrogation en lui fournissant l'énergie dont il a besoin pour fonctionner. Le tag répond par un signal radiofréquence contenant une information permettant d'identifier l'objet auquel ce dernier est attaché [9]. Le middleware RFID représente la partie intelligente du système RFID. Il est chargé de surveiller les lecteurs, de gérer les données échangées et de les agréger aux applications dédiées [8].

En général, les étiquettes peuvent être classifiées en trois catégories en fonction de leur source d'alimentation: passive, semi-passive et active. Une étiquette active dispose d'une source d'énergie interne qui fournit l'énergie à la fois au circuit intégré (par exemple capteurs) et l'émetteur. Elle peut envoyer un signal radiofréquence sans être appelé par un lecteur RFID. Une étiquette passive n'a pas de source d'énergie intégrée, ce qui fait qu'elle ne dépend que du signal reçu du lecteur RFID pour générer sa propre réponse. Une étiquette semi-passive a une source d'alimentation interne qui ne maintient en permanence que sa propre puce électronique et d'autres circuits ou capteurs connectés au circuit de base mais elle est basée sur le même principe qu'une étiquette passive pour la transmission des données au lecteur RFID [8].

III. APERÇU DE LA CONCEPTION DU SYSTÈME PROPOSÉ

L'objectif de cette partie est de concevoir et modéliser un système de contrôle d'accès au parking qui n'autorise que les véhicules autorisés puis enregistre ses entrés/sorties dans une base de données comme ci-décrit dans la figure Fig.2.

Fig. 2. Parking access control system.

Ces véhicules doivent avoir un tag RFID collé à l'intérieur sur la pare-brise de telle sort qu'il n'y ait pas d'obstacle aux ondes entre le lecteur RFID et le tag RFID, à l'exception de la vitre qui n'a aucune influence sur la transmission des ondes.

A. *MATÉRIEL*

Pour mettre en œuvre un système de contrôle d'accès au parking basé sur RFID, divers composants ont été utilisés comme décrits la table suivante (TABLE I):

TABLE I
LISTE DES COMPOSANTS.

Composant	Description et caractéristique de chaque composant utilisé
Node MCU V3 (ESP8266)	• est un firmware open-source et un kit de développement qui joue un rôle important dans la conception d'un produit IoT. • agit comme une interface entre la partie logicielle et la partie matérielle du projet.
RFID module (RC522)	• une interface qui permet l'identification sans contact à partir d'un carte ou une clé RFID. Il utilise la bande 13.56MHz.
Afficheur LCD	• utilisé pour afficher les informations de stationnement.
I2C	• un bus de communication série et il est utilisé pour réduire le nombre de branches de LCD de 16 à 4.
Servo Motor	• un appareil à un axe rotatif. • utilisé comme un contrôleur de la barrière du parking.
Buzzer	• utilisé pour produit un son lorsqu'il y'a une lecture infructueuse du module RFID.
LED	• utilisé pour montrer une lecture réussie (lumière verte) ou infructueuse (lumière rouge) du module RFID.

La conception du système RFID proposé est illustrée dans la figure Fig.3. Le système proposé est simulé à l'aide du module RFID MF RC522. Le lecteur RC522 est connecté à la carte NodeMCU V3 afin d'établir la communication entre le module RFID et le programme et les autres composants sont également connectés à la carte NodeMCU V3 pour les faire agir en fonction de la situation en utilisant un ordre de connexion spécifique.

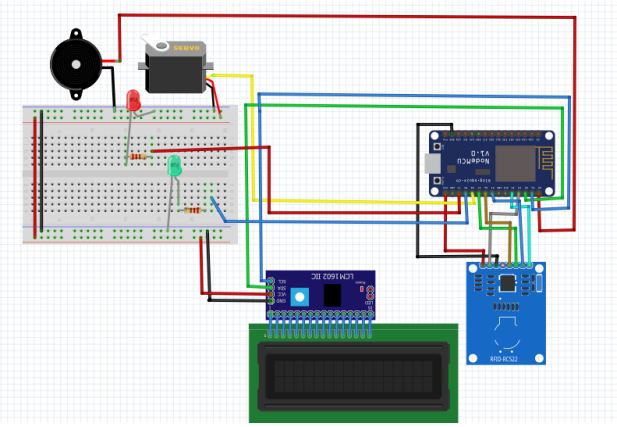

Fig. 3. Schéma général du système parking en Frizting.

B. LOGICIEL

Dans cette partie, nous avons utilisé différents logiciels (Arduino IDE, phpMyAdmin, MySQL) pour contrôler l'entrée/sortie du parking en permettant l'accès aux utilisateurs autorisés et capturer ses informations puis les stocker dans une base de données.
La base de données utilisée a été créée à l'aide de logiciel Wampserver. Chaque information d'utilisateur est entrée automatiquement dans la base de données à l'aide d'une interface web qui permet d'ajouter un nouveau utilisateur en utilisant ses informations de base (nom, email, numéro de téléphone), mettre à jour ses informations ou les supprimer comme indiqué sur les figures 4, 5, 7 et 8 .

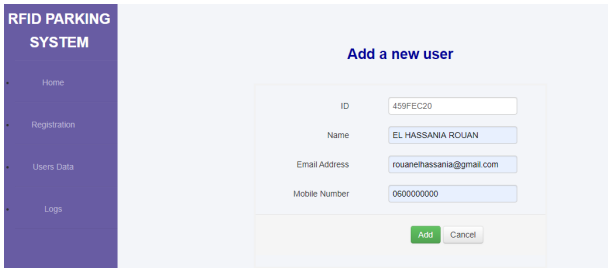

Fig. 4. Interface pour ajouter un utilisateur.

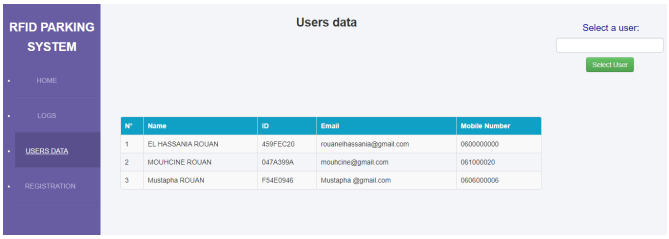

Fig. 5. Interface de utilisateurs enregistrés.

La figure 5 montre une option "select User" qui permet aux administrateurs de chercher un utilisateur en utilisant son code d'identification afin de mettre à jour ses informations ou de les supprimer comme indiqué sur la figure 6.

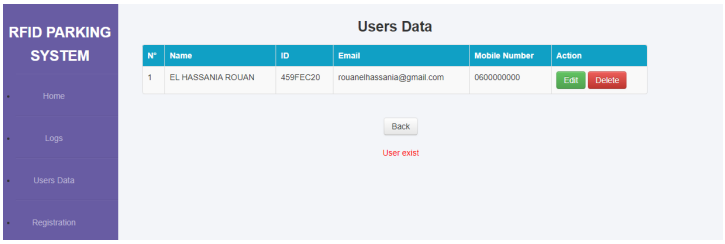

Fig. 6. Interface pour mettre à jour les informations d'un utilisateur.

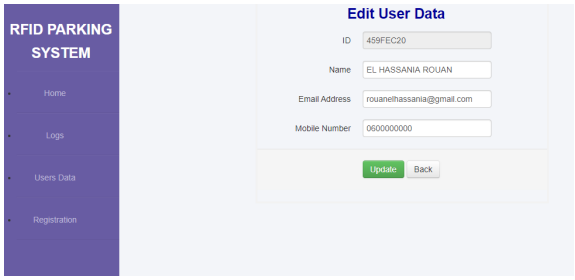

Fig. 7. Interface pour mettre à jour les informations d'un utilisateur.

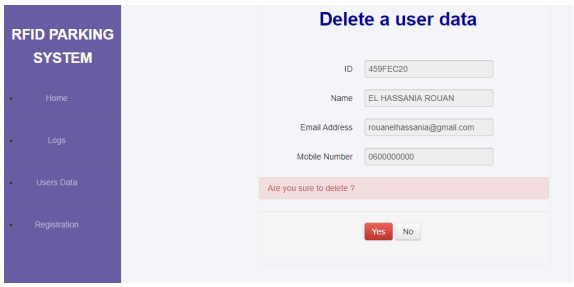

Fig. 8. Interface de suppression d'un utilisateur.

Lorsqu'une lecture valide du tag RFID d'un utilisateur est détectée, le système stocke automatiquement ses informations d'entrée/sortie dans la base de données et les affiche dans une interface Web.

IV. PRINCIPE DE FONCTIONNEMENT & RÉSULTAT

Cette section décrit le principe de fonctionnement du système proposé et sa mise en œuvre comme preuve de concept. Les procédures sont décrites à l'aide d'un organigramme comme la figure 9 montre.

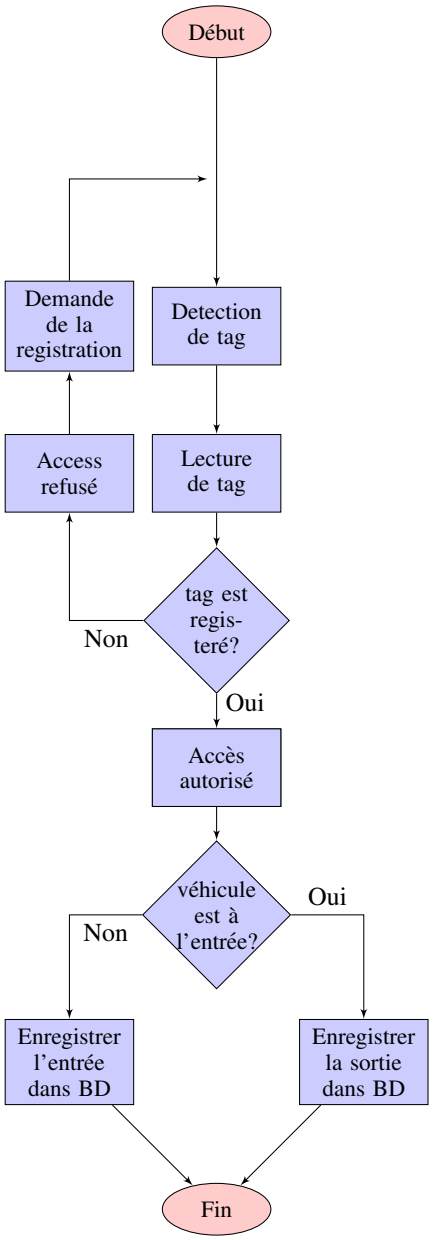

Fig. 9. Organigramme du système proposé.

Dans ce système, les IDs enregistrés sont stockés dans la carte NodeMcu V3 et aussi dans la base de données. Lorsqu'un véhicule s'approche de l'entrée ou la sortie du parking, le lecteur RFID détecte si le présent véhicule a un tag RFID ou non. Si le véhicule ne le dispose pas, la barrière sera toujours fermée, sinon le lecteur RFID peut lire l'ID stocké dedans et transmettre l'ID capturé à la partie logicielle pour vérifier son enregistrement. Dans ce cas, la partie logicielle sera en mesure de comparer l'ID capturé avec les IDs stockés dans la base de données, si l'ID capturé correspond à l'un des IDs stockés, la barrière s'ouvrira automatiquement et un accès autorisé sera affiché à l'écran (Figure 10) . Si l'ID capturé ne correspond pas, la barrière restera fermée et l'alarme se déclenchera pour indiquer un accès infructueux (Figure 11). Parallèlement, le temps d'entrée et de sortie seront stockés dans la base de données et affichés dans une interface web comme illustré dans la figure 12.

Fig. 10. Accès autorisé.

La figure 10 montre que l'utilisateur a un ID valide et donc l'accès sera autoriser et s'afficher sur l'afficheur LCD et la barrière s'ouvrira automatiquement.

Fig. 11. Accès refusé.

La figure 11 montre que l'utilisateur a un ID non enregistré qui lui interdit d'accéder au parking et un message d'information sera afficher sur l'écran LCD et la barrière reste fermée.

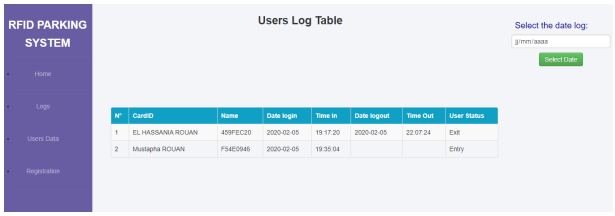

Fig. 12. Login/Logout interface.

V. CONCLUSION

Dans ce document, nous avons conçu et implémenté un système de contrôle d'accès automatique au parking basé sur la technologie RFID qui permet de surveiller son entrée et sa sortie. Le système proposé repose principalement sur l'utilisation de la technologie RFID et la carte NodeMCU pour différencier les véhicules autorisés des véhicules non autorisés. Le système proposé permet d'éviter le temps d'attente aux terminaux d'entrée et de sortie et d'augmenter la sécurité dans le parking en autorisant l'accès uniquement aux navetteurs autorisés, ce qui offre un moyen fiable d'accorder ou de refuser l'accès dans une zone restreinte. Il fournit également un système de surveillance efficace et bon marché tout en fournissant des informations en temps réel sur les mouvements des utilisateurs.

REFERENCES

[1] Kay Li Ng, Choo W. R. Chiong and Regina Reine, *Vehicle Recognition System using RFID Technology for Parking Management System*, IOP Conference Series: Materials Science and Engineering, Volume 495, Number 1, 2019.

[2] E. Tsiropoulou, J. Baras, S. Papavassiliou and S. Sinha, *RFID-based smart parking management system*, Cyber-Physical Systems, 2017.

[3] B. Salah, *Design, simulation, and performanceevaluation-based validation of a novel RFID-based automatic parking system*, Journal of SIMULATION, 2019.

[4] Castro, L., & Fosso Wamba, S. (2007). *An Inside Look at RFID Technology*. Journal of Technology Management & Innovation, 2(1), 128-141.

[5] C. Y. Lam and W. H. Ip, An Integrated Logistics Routing and Scheduling Network Model with RFID-GPS Data for Supply Chain Management, Wireless Pers Commun 105, 803–817, 2019.

[6] Ramudzuli ZR, Malekian R, Ye N. *Design of a RFID system for real-time tracking of laboratory animals*. Wirel Pers Commun. 2017;95(4):3883-3903.

[7] X. Zhang and X. Lian, *Design of warehouse information acquisition system based on RFID*, in IEEE International Conference on Automation and Logis- tics, pp. 2550-2555, Sept. 2008.

[8] Ahmed M. Mohammed, *Modelling and optimization of an RFID-based supply chain network*, thesis, 2018.

[9] A. Mbacké, N. Mitton and H. Rivano, *RFID reader anticollision protocols for dense and mobile deployments*, dans MDPI Electronics Special Issue "RFID Systems and Applications",pp 22, 2016.

El hassania Rouan
Faculté des Sciences et Techniques
Université Sultan Moulay Slimane
via B.P : 523 Béni - Mellal, Maroc
Tel: + 212 (0) 523 48 51
Fax: +212 (0) 523 48 52 01
E-mail: rouanelhassania@gmail.com

LMF versus Combined LMS/F Algorithm for BRAN B channel Identification

Mohammed Zidane[(1)], Said Safi[(2)], Miloud Frikel[(3)] and Mohamed Sabri[(4)]

[(1)] ERMAM Team, Department of Physics-Chemistry, Polydisciplinary Faculty, Ibn Zohr University, Ouarzazate, Morocco

[(2)] Department of Mathematics and Informatics, Polydisciplinary Faculty, Sultan Moulay Slimane University, Morocco

[(3)] GREYC Lab UMR 6072 CNRS, ENSICAEN, Caen, France

[(4)] Department of Physics, Faculty of Sciences and Techniques, Sultan Moulay Slimane University, Morocco

***Abstract*—In this framework, a comparative study of Broadband Radio Access Networks (BRAN) channel identification is analyzed and discussed. Indeed, the selected adaptive algorithms, such as Least Mean Square (LMS), Least Mean Fourth (LMF) and combined (LMS/F), are applied to BRAN B channel model. The numerical simulation results of various Signal-to-Noise Ratios (SNRs) and the parameters those are optimally chosen, show that the presented algorithms can simulate the measured BRAN B channel with different accuracy levels.**

***Index Terms*—Identification; LMF; LMS/F; BRAN B.**

I. Introduction

The purpose of communication systems is transport of information between source and recipient through transmission channel. The channel part is element in which the work is carried out. Indeed, transmission channel is an important element in digital communication chain, it brings more or less important disturbance to the transmitted signal created by the propagation conditions (equipment imperfection, multi-path propagation, presence of noise, weaknesses, ...), the latter causes degradation of transmitted signal which results in occurrence of transmission errors. These errors can be very troublesome for the faithful restitution of information to recipient. There are different identification techniques that can be implemented. In this work we use adaptive methods for identifying radio channel.

Adaptive systems identification has attracted attention in recent years, there are several applications such as interference cancelation, spectral subtraction, wireless localization, adaptive beamforming, and channel identification [1]. In this framework we address the application of adaptive algorithms in standard BRAN B channel identification. Channel identification is preformed to use LMS, LMF and combine (LMS/F). However, the LMS algorithm is one of most commonly used algorithms in adaptive signal processing, which is proposed by Widrow et al. [2]. Its popularity comes from the fact that it's very easy to implement [1]. It comes from whole family of algorithms based on steepest descent. This algorithm is widely studied in several works [1]−[5]. Regarding LMF algorithm it was introduced by Walach and Widrow as supplement to LMS [6]. On the other hand, this algorithm uses the fourth-order power optimization criterion in the place of the square power in LMS. Indeed, the complexity computational of LMF is very lofty, which is due to higher-order power optimization in its equation updated [1].

In order to improve the performance of these algorithms, Lim et al. [7] developed a new adaptive algorithm that combines the advantages of the LMS and LMF methods called LMS/F. Gui et al. [1] introduced the combined LMS/F algorithm to adaptive system identification by considering the tradeoff between convergence swiftness and steady state performance.

In this contribution, we use the LMS, LMF and combined LMS/F algorithms developed by Gui et al. [1] in adaptive channel identification. Indeed, to test effectiveness of these algorithms we have considered BRAN B practical frequency selective fading channel [8], [9] normalized for wireless communication systems, with non-Gaussian signal input, for different SNRs and fixed data input.

The rest of this paper is organized as follows. Adaptive system identification is described in section II. In section III an overview of the adaptive algorithms is presented. Numerical simulation results followed by an interesting discussion and analysis are provided in section IV and, finally, a conclusion is given in the last section of this investigation.

II. Adaptive channel identification

The BRAN B adaptive channel identification considered in this paper is described in figure 1.

At the receiver side, observed signal $z(k)$ is given by the following equation:

$$z(k) = \sum_{i=0}^{L-1} h(i)x(k-i) + w(k), \tag{1}$$

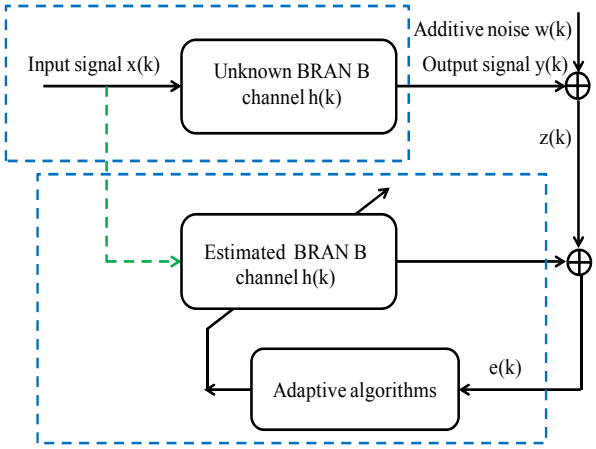

Fig. 1. Adaptive system identification configuration

where $h = [h_0, h_1, ..., h_{L-1}]$ and $x(k) = [x(k), x(k-1), ..., x(k-L+1)]$ denote channel coefficient and the L-length input signal vector of $x(k)$ respectively. $w(k)$ is an additive Gaussian noise.

For this system we assume that:

- The input signal, $x(k)$, is non-Gaussian and considered independent and identically distributed (i.i.d) with zero mean;
- The additive noise sequence $w(k)$ is Gaussian with zero mean, i.i.d, and independent of $x(k)$.

III. Adaptive algorithms identification

A. LMS algorithm

In order to identify the coefficients of channel using input-output relationship of signal the LMS algorithm is used. Let $h(k)$ be the identified coefficient vector of the adaptive channel at kth iteration.
In the classical LMS, the cost function $J_1(k)$ is defined as:

$$J_1(k) = \frac{1}{2}e^2(k), \tag{2}$$

where $e(k)$ is the instantaneous error:

$$e(k) = z(k) - h^T(k)x(k) \tag{3}$$

The channel coefficients vector is then updated by:

$$h(k+1) = h(k) - \mu_1 \frac{\partial J_1(k)}{\partial h(k)} = h(k) + \mu_1 e(k)x(k), \tag{4}$$

where μ_1 is the step-size parameter that tradeoffs stability and convergence rate of LMS algorithm.

B. LMF algorithm

From the classical LMS algorithm with a variable step size, we can deduce the LMF algorithm. In the standard LMF, the cost function $J_2(k)$ is defined as [1]:

$$J_2(k) = \frac{1}{4}e^4(k) \tag{5}$$

The filter coefficient vector is then updated by:

$$h(k+1) = h(k) - \mu_2 \frac{\partial J_2(k)}{\partial h(k)} = h(k) + \mu_2 e^3(k)x(k), \tag{6}$$

where μ_2 is the step size that controls stability and convergence rate of LMF algorithm.

C. Combined LMS/F algorithm

Using the cost functions of LMF and LMS algorithms, Gui et al [1] developed a cost function of the combined LMS/F algorithm can be constructed as:

$$J_3(k) = \frac{1}{2}e^2(k) - \frac{1}{2}\varepsilon ln(e^2(k) + \varepsilon), \tag{7}$$

to trade off between convergence swiftness and performance we use the threshold parameter ε.
Thus, the updating equation of combined LMS/F algorithm is given by [1]:

$$h(k+1) = h(k) + \mu_3 \frac{\partial J_3(k)}{\partial h(k)} = h(k) + \mu_3 \frac{e^3(k)}{e^2(k) + \varepsilon}x(k), \tag{8}$$

where μ_3 designed the gradient step-size used to control the performance and convergence rate.

IV. Numerical simulation results

This section contains numerical simulation results. Performances of LMF versus combined LMS/F algorithm which compared. We consider standard BRAN B radio channel where the parameters and the order are known. The measured impulse response parameters corresponding the standard BRAN B radio channel summarized in Table I.

TABLE I
Delay and magnitudes of 18 targets of BRAN B channel

Delay $\tau_i[ns]$	Mag. $A_i[dB]$	Delay $\tau_i[ns]$	Mag. $A_i[dB]$
0	−2.6	230	−5.6
10	−3.0	280	−7.7
20	−3.5	330	−9.9
30	−3.9	380	−12.1
50	0.0	430	−14.3
80	−1.3	490	−15.4
110	−2.6	560	−18.4
140	−3.9	640	−20.7
180	−3.4	730	−24.6

To measure the effect of noise, we define the SNRs by the following equation:

$$SNR = 10\ log_{10}\Big[\frac{\sigma_y^2(n)}{\sigma_w^2(n)}\Big] \tag{9}$$

To measure the precision of estimated values with respect to the real values, we define the Normalized Mean Square Error (NMSE) for each run as:

$$NMSE = \sum_{i=1}^{18}\Big[\frac{h(i) - \widehat{h}(i)}{h(i)}\Big]^2, \tag{10}$$

where $\widehat{h}(i)$ and $h(i)$, $i = 1, ..., L$, are respectively the estimated and the real parameters in each run.

A. Adaptive identification of BRAN B radio channel

In this subsection, the simulation results for BRAN B radio channel using adaptive algorithms in different SNRs and data input are presented. The parameters, in numerical simulations, are set as: $\mu_1 = 0.02$, $\mu_2 = 0.001$, $\mu_3 = 0.008$ and $\varepsilon = 0.1$.

In the figures 2, 3 and 4 we represents the estimation of the BRAN B parameters using LMS, LMF and combined LMS/F in case of SNRs=0, 10 and 20 dB, the data length of non-Gaussian signal input is N=2048 and for 100 Monte-Carlo runs.

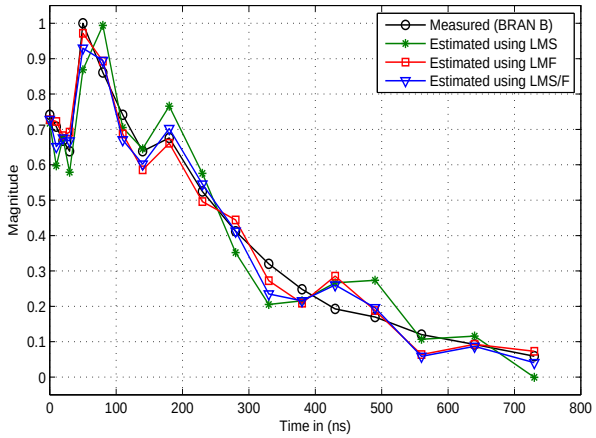

Fig. 2. BRAN B channel identification using the LMS, LMF and LMS/F algorithms for SNR=0 dB and an data input N=2048

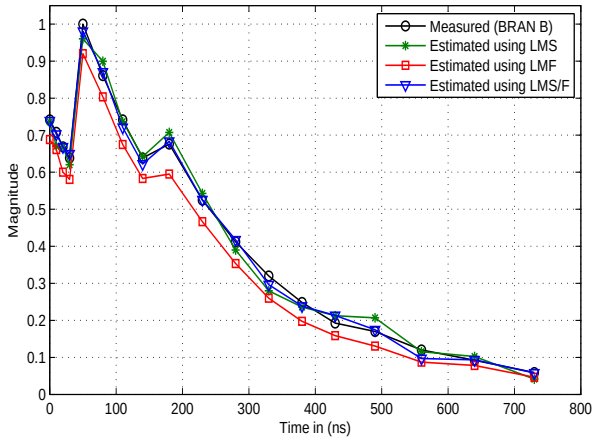

Fig. 3. BRAN B channel identification using the LMS, LMF and LMS/F algorithms for SNR=10 dB and an data input N=2048

For high noise environment (SNR =0 dB), the (LMF, LMS/F) algorithms show their efficiency principally the first 12^{th} values of the BRAN B impulse response it follows very well the real model, the sixth last target we have a minor

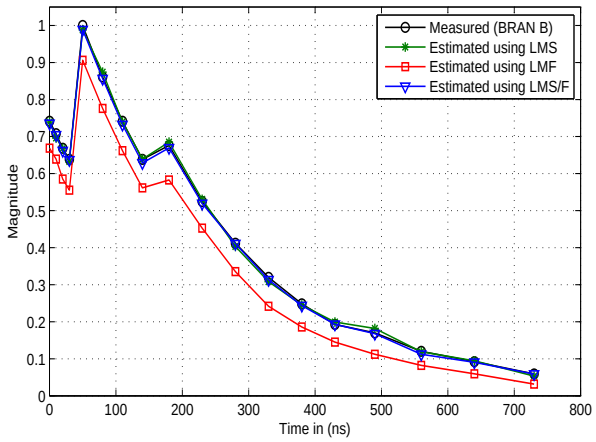

Fig. 4. BRAN B channel identification using the LMS, LMF and LMS/F algorithms for SNR=20 dB and an data input N=2048

difference between the estimated and measured. The use of the LMS algorithm we remark in the impulse response channel (BRAN B) identification where the estimated parameters do not follow those measured. This due that the (LMF, LMS/F) algorithms are very adequate in very noise environment than the LMS algorithm.

For (SNRs=10 dB and 20 dB), we observe that the (LMS, LMS/F) algorithms are more precise it follows very well the measured model of the BRAN B for all target, and we observe a meagre influence of the noise on the parameters estimation even for (SNR$\geq$10 dB). Although, the estimated parameters of BRAN B using LMF algorithm it follows the real model with over-estimation (Fig. 4). This due that the (LMS, LMS/F) algorithms are adequate in low noise environment than the LMF algorithm. For example, if SNR=20 dB, the NMSE(LMS)= 0.0185; NMSE(LMF)= 0.9075; NMSE(LMS/F)= 0.0068.

In figure 5 we represent the NMSE for different SNRs, obtained using LMS, LMF and LMS/F algorithms. Based on the numerical simulation results presented in figure 5, we may conclude the following:

- First, the LMF algorithm show their efficiency in the term the NMSE of BRAN B channel identification with good precision and is adequate in strong noise environment principally if (SNR$\leq$6 dB) than the (LMS, LMS/F) algorithms;
- Second, the LMS/F algorithm gives good results for the NMSE of BRAN B channel, for different SNR$\geq$6 dB versus others algorithms and is adequate in this case;
- Third, the LMS is more precise than LMF algorithm principally if the SNR$>$8 dB.

In the goal to test the efficiencies of the presented algorithms in

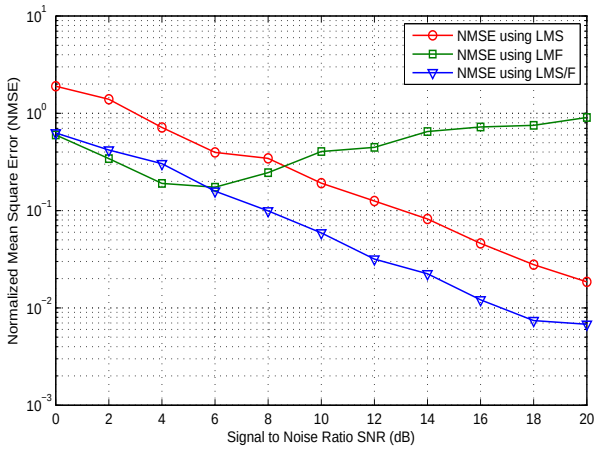

Fig. 5. NMSE values as a function of SNR using the LMS, LMF and LMS/F algorithms

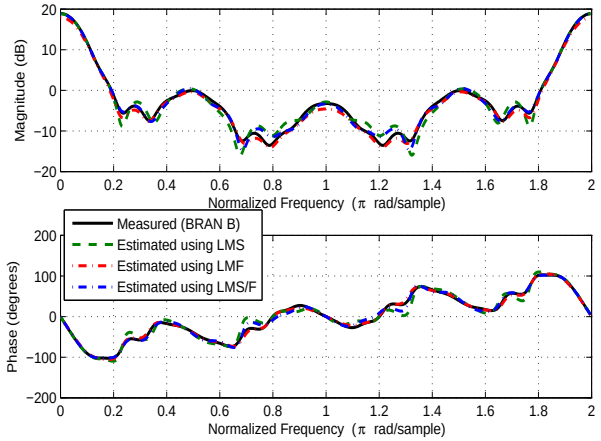

Fig. 7. Estimated magnitude and phase of the BRAN B channel for SNR=10 dB and data input N=2048

the frequency domain, we plot in figures 6 and 7 the estimation magnitude and phase of BRAN B channel in the case of SNR=0 dB and SNR=10 dB respectively.

From the figure 6 we can conclude that using the LMS

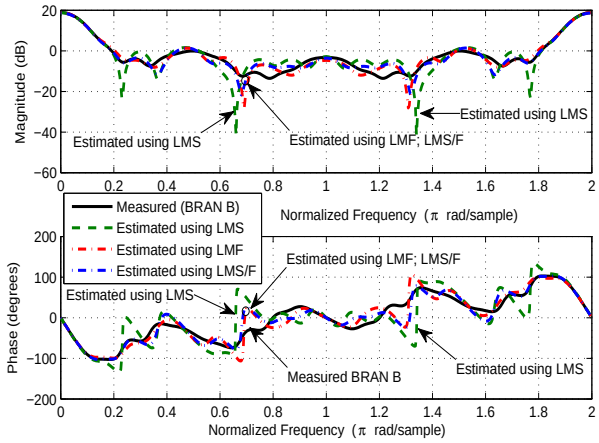

Fig. 6. Estimated magnitude and phase of the BRAN B channel for SNR=0 dB and data input N=2048

algorithm we have a more difference between the measured BRAN B channel and the estimated magnitude and phase. Furthermore, the estimated magnitude and phase follow the real model with a minor difference. This due that the (LMF and LMS/F) are adequate in the robust noise environment.

In case of SNR=10 dB (Fig. 7) using all algorithms, we observe that the estimated magnitude and phase uniformly converges to the real model, and we remark a little influence of the noise on the magnitude and phase estimation.

V. Conclusion

The work presented in this paper structured around identification of BRAN channel. Indeed, we have presented three adaptive algorithms: LMS, LMF and combined LMS/F. Initially, we started with overview of these algorithms. Also, to illustrate the identification performance we applied these algorithms in BRAN B channel model. According to numerical simulations results, we found that LMF is adequate in low SNRs region. LMS/F attain good performance in low and in high SNRs regions than others algorithms. We can also note that LMS can achieve much better performance than LMF in high SNRs region (superior to 8 dB).

References

[1] Gui, G., Peng, W., Adachi, F. (2014). Adaptive system identification using robust LMS/F algorithm. *International Journal of Communication Systems*, 27(11): 2956−2963.

[2] Widrow, B., McCool, JM., Larimore, MG., Johnson, CR. (1976). Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proceedings of the IEEE*, 64(8): 1151−1162.

[3] Gui, G., Mehbodniya, A., Adachi, F. (2015). Sparse LMS/F algorithms with application to adaptive system identification. *Wireless Communications and Mobile Computing*, 15(12): 1649−1658.

[4] Gui, G., Adachi, F. (2013). Improved least mean square algorithm with application to adaptive sparse channel estimation. *EURASIP Journal on Wireless Communications and Networking*, 2013(1): 204.

[5] Gui, G., Adachi, F. (2014). Sparse least mean fourth algorithm for adaptive channel estimation in low signal-to-noise ratio region. *International Journal of Communication Systems*, 27(11): 3147−3157.

[6] Walach, E., Widrow, B. (1984). The least mean fourth (LMF) adaptive algorithm and its family. *IEEE transactions on Information Theory*, 30(2): 275−283.

[7] Lim, S. J., Harris, J. G. (1997). Combined LMS/F algorithm. *Electronics Letters*, 33(6): 467−468.

[8] ETSI (1999, January). Broadband Radio Access Networks (BRAN); High Performance Radio Logical Area Network (HIPERLAN) Type 2; Requirements and architectures for wireless broadband access.

[9] ETSI (2001, December). Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical Layer.

Reconnaissance Automatique de la dialecte marocain en milieu réel à l'aide de PocketSphinx

A. OUISAADANE
Department of Mathematics
and Computer Science
Polydisciplinary Faculty
Sultan Moulay Slimane University
Benimellal ,Morocco
Email: Abdelkbir.wiss@gmail.com

S. SAFI
Department of Mathematics
and Computer Science
Polydisciplinary Faculty
Sultan Moulay Slimane University
Benimellal ,Morocco
Email: safi.said@gmail.com

M. FRIKEL
GREYC Laboratory
ENSICAEN School
LAC Laboratory
Caen-Normandie University
Caen, France
Email: miloud.frikel@ensicaen.fr

Résumé: **Dans ce travail, nous allons réaliser un système de la reconnaissance automatique de dialecte marocain arabe sous différents bruits additifs en utilisant l'outil open source PocketSphinx. Nous préparons un corpus de petite taille des dix salutations les plus célèbres au dialecte marocain dans les conversations téléphoniques. Ce corpus de parole enregistré par 60 locuteurs (30 hommes et 30 femmes), prennent 1800 expressions dans lesquels chaque locuteur prononce chaque expressions trois fois, les expériences se faites dans des conditions réelles (bruitées). L'extraction des caractéristiques se fait avec les coefficients cepstraux dans l'échelle des Mels (MFCC) est la modélisation acoustique basée sur la monophonie est effectuée par les modèles de Markov cachés (HMM). Les systèmes de reconnaissance automatique de la parole obtiennent des performances acceptables dans des conditions sans bruit, mais les performances se dégradent considérablement en présence de bruit.**

Keywords— Moroccan dialect, Noisy speech, HMM, Pocket Sphinx, ASR.

I. INTRODUCTION

La parole est l'un des moyens les plus naturels par lequel des personnes communiquent, ainsi c'est le moyen d'interaction entre l'humaine et machine. Ceci nécessite la création d'une interface homme-machine réalisée avec un système de reconnaissance vocale. Ce système a pour objet la transformation du signal acoustique en une séquence de mots qui, idéalement, correspond à la phrase prononcée par un locuteur. Différentes recherches dans le domaine de la reconnaissance de la parole sont menées avec différents angles, en passant en premier par l'étape d'analyse et de paramétrisation : MFCC,LPC, RASTA, etc. [1]. Les systèmes reconnaissance automatique de la parole (RAP en anglais ASR) utilisent des paramètres cepstraux comme représentation acoustique du signal vocal. Les paramètres acoustiques les plus performants et les plus utilisés sont les coefficients cepstraux de Mel (MFCC) [2], [3]. La procédure de calcul de ces coefficients s'effectue en plusieurs étapes et le choix de ces paramètres repose sur la capacité d'interpolation, robustesse au bruit et l'adaptation à la variabilité inter et intra locuteurs [4] [5]. Plusieurs méthodes de classification ont été utilisées pour RAP, dont certaines sont encore utilisées, certaines ont évolué, et des nouvelles méthodes sont apparues. Les réseaux de neurones profonds (DNN) et les Modèles de Markov Cachés (HMM) se sont les meilleures Algorithme et méthodes de classification utilisés dans l'apprentissage des signaux vocales [6], [7]. Le choix des méthodes d'apprentissage (HMM, DTW, SVM, GMM, ANN,DNN etc.) [7] [8] dépend des paramètres d'analyse, la taille des données et la capacité de généralisation. En somme, il faut tenir compte des méthodes de test et de généralisation pour valider le modèle d'apprentissage. La structure générale du système de reconnaissance vocale basé sur HMM comprend deux phases: une phase d'apprentissage dont le but est la construction de modèles acoustiques (modèles HMM) et la phase de reconnaissance qui renvoie le mot le plus probable [9] .Nombreux système de reconnaissance vocale open source sont également disponible, qui sont basé sur les modèles de Markov cachés (HMM) [9] tels que CMU Sphinx, HTK and Kaldi en déférent langues.

De nombreux travaux ont été menés dans ce domaine en se concentrant sur l'anglais. Aujourd'hui, les recherches se focalisent sur la reconnaissance des langues régionales comme les dialectes Arabe, la parole en champ lointain, les environnements bruyants, l'adaptation des modèles, robustesse au bruit, variabilité des canaux, l'accent du locuteur, …

Cet article décrit un système de reconnaissance vocale utilisant CMU PocketSphinx [10] pour reconnaître dix phrases courtes en dialecte marocain (DARIJA).

Ce papier est organisé comme suit : après cette brève introduction, la section 2 donne un aperçu sur quelques travaux liés à la reconnaissance de dialecte marocain. La section 3 présente notre système de reconnaissance proposé avec une présentation détaillée de ces différents modules. La section 4 montre les expériences réalisées, les résultats comparatifs obtenus et les discutions explicatives. La conclusion générale et les perspectives sont données dans la section 5.

II. TRAVAUX CONNEXES

H. Satori et al [11] ont proposé un système de reconnaissance automatique de la parole. Le système a été basé sur CMU Sphinx-4 ayant un modèle acoustique, un modèle de langue et un dictionnaire. Ils ont présenté des expériences pour démonter l'adaptabilité de ce système pour la langue arabe. Ils ont formé le système avec un corpus constitué de 300 piste de dix premiers chiffres de l'arabe classique de 0 à 9 (10 chiffres. 5 répétitions. 6 locuteurs). Les résultats obtenus étaient satisfaisants où il a atteint un taux de reconnaissance de 83%.

El Amrania et.al [12] ont proposé un système base sur CMUsphinx de reconnaissance de phonèmes arabes de la parole arabe appliqué au Saint Coran. Ils ont formé et évaluer un modèle de langage pour la narration « Hafs ». Les expériences ont abouti à un très faible taux d'erreur de mots

(WER) atteignant 1,5% pour un ensemble petit de fichiers audio pendant les phases d'apprentissage, et de test.

M. Elmahdy et al [13] ils ont proposé un système ASR à la pointe de la technologie pour l'arabe familier levantin (LSA) en utilisant un modèle acoustique de 50 heures de données vocales. Leur décodage par de l'ensemble de test (10 heures) a donné un WER de 30,5%.

Bezoui et al [14] ont construit un système d'identification du locuteur en dialecte marocain basant sur MFCC pour extraire les caractéristiques de la parole inconnue et les modèles de Markov cachés pour la classification .Le code à été développé dans MATLAB qui arrivais à identifier le locuteur de manière satisfaisante.

En terme d'étudier la robustesse des systèmes de reconnaissance vocale pour la langue arabe, quelques travaux sont faits dans ce domaine, on mentionne par exemple les recherches de Touazi et Debyeche [15] qui ont présenté ARADIGIT-2 une base de données de reconnaissance de chiffres arabe basée sur la boîte à outils pour modèles de Markov cachés (HTK) et l'indépendante des locuteurs arabes, ils ont l'utilisé pour l'évaluation des systèmes robustes au bruit. Les tests se faits dans différentes conditions et type du bruit. Les résultats obtenus sont donnés un WER de 0,44% dont l'apprentissage avec des donnés propre et de 0,58% dont l'apprentissage avec des donnes multi-conditions.

Certains systèmes vocaux ont été proposés pour la dialecte marocain de tamazight . Un exemple notable est le système de reconnaissance des 33 lettres de l'alphabet Amazigh réalisés par neuf locuteurs, conçu par Meryam Telmem et Youssef Ghanou [16] . Ils ont proposé une architecture basée sur des modèles de Markov cachés (HMM) avec la librairie open source CMUSphinx, et ont atteint une précision d'environ 82%.

On peut aussi mentionner quelques travaux qui portent sur la reconnaissance de la parole Arabe/Dialecte/Amazigh avec différentes approches par exemple : [17] ,[18],[19],[20],[21],[22].

III. SYSTEME DE RECONNAISSANCE DE DIALECTE MAROCAIN

A. Dialecte marocain

L'arabe dialectal marocain, appelé au Maroc DARIJA, est une langue-toit rassemblant plusieurs variétés d'arabe dialectal parlées au Maroc. Il appartient au groupe des dialectes maghrébins. L'Arabe dialectal qui est une langue peu dotée et présente des variations d'un pays à un autre et voire d'une ville à une autre. Les dialectes arabes sont des langues complexes et riches et sont bien utilisés dans la communication quotidienne, les réseaux sociaux, la radio, les émissions de télévision, les conversations téléphoniques, etc [11].

Naturellement, DARIJA est essentiellement basé sur l'arabe standard moderne (MSA). En outre, ils contiennent des mots, des expressions et des structures linguistiques originaux (CA), différentes dialectes Amazigh, français, espagnol ainsi que d'autres langues romanes méditerranéennes. Dans ce travail nous préparons notre propre corpus de petite taille des dix salutations les plus célèbres au dialecte marocain dans les conversations téléphoniques nommé DARIJA_MO. Nous présentons les détails de ce corpus dans le chapitre 4.

B. Architecture du système proposé

Le system de la reconnaissance vocale en dialecte marocain proposé est implémenté à l'aide de la boîte à outils PocketSphinx qu'est une librairie gratuite permettant d'intégrer la reconnaissance vocale dans des projets écrit en language C à l'aide des fonctionnalités du projet open source, elle manipule aussi les modèles de Markov cachés ,elle est développée à l'Université Carnegie Mellon (CMU Sphinx USA)[10]. Cette boite à outils se compose de :

Sphinxtrain : c'est un outil qui permet de créer le modèle acoustique et le modèle de langage.

Sphinx : qui est une bibliothèque de support requise par SphinxTrain.

PocketSphinx : qui fournit une interface python aux bibliothèques CMU Sphinxbase et Pocketsphinx.

Cmudict : permet la mise au point de dictionnaire de prononciation.

Pour installer Pocketsphinx sur un ordinateur mono-carte, certains composants préalables doivent être installées dans le système, par exemple compilateur gcc, python, libasound dev, alsa utils, bison. Après avoir téléchargé et compilé le code source de Pocketsphinx, un programme de test utilisant le modèle de langage, le modèle acoustique et le dictionnaire doit être tester par défaut.

Pocketsphinx utilise des HMM comme modèle acoustique et le modèle N-Gram comme modèle de langage. Elle utilise à la fois les deux modèles pour décoder le signal acoustique [11],[16] .

CMUSphinx a plusieurs modèles acoustiques et linguistiques tels que l'anglais américain, l'allemand, l'espagnol, l'indien, etc., mais le modèle arabe n'est pas encore disponible; nous avons ensuite construit la reconnaissance vocale de dialecte arabe du Maroc l'accent de région de Beni Mellal, comme une nouvelle langue pour CMUSphinx. Nous avions créé le modèle acoustique et le modèle de langage pour le corpus DARIJA. Le modèle acoustique a été formé sur des données principalement réels et quelques bruyantes.

La Fig.1 décrit l'architecture d'un système ASR. Le modèle acoustique, le modèle de langage et le dictionnaire sont les données d'entrées au module de reconnaissance.

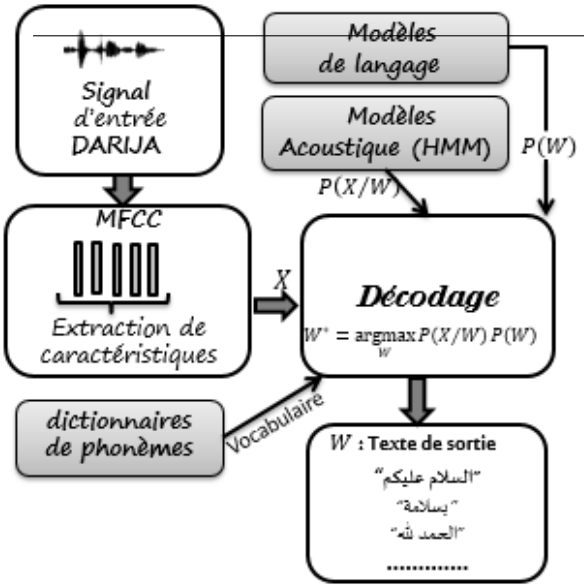

Fig. 1. Architecture du système de reconnaissance automatique de la dialectale arabe Marocaine.

C. *Extraction des caractéristiques*

Le signal acoustique contient de différentes sortes de renseignements sur la parole : la paramétrisation, la segmentation parole / non parole et des prétraitements. La paramétrisation MFCC (Mel Frequency Cepstral Coefficients) est basée sur la perception humaine de son : l'échelle de Mel, sur l'évidence connue que les renseignements portés par les composantes de la fréquence basse du signal de parole sont plus importants phonétiquement pour les humains que les composantes à haute fréquence.

Les coefficients cepstraux sur l'échelle Mel (MFCC, Mel-Frequency Cepstral coefficients) est la plus courante méthode d'extraction de caractéristiques utilisé dans les systèmes de reconnaissance de la parole et du locuteur. Les coefficients MFCC Sont plus discriminant, plus robustes au bruit ambiant et moins corrélés entre eux.

L'analyse acoustique est divisée en trois étapes, le filtrage analogique, la conversion numérique et le calcul de coefficients, nous donnerons brièvement les étapes d'une analyse MFCC dans la figure 2.

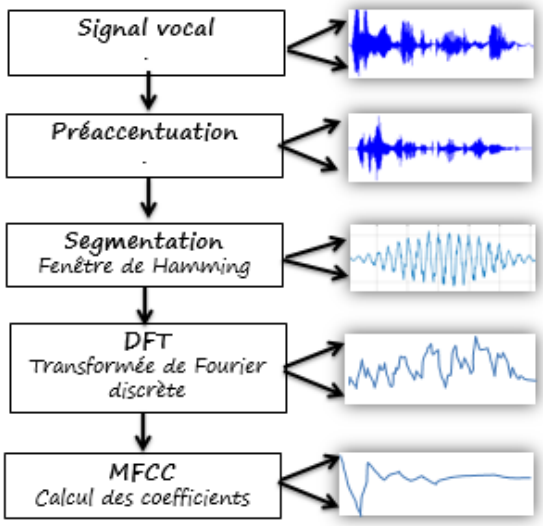

Fig.2 Etapes d'extraction des paramètres MFCC.

Pour l'approche basée sur HMM, l'outil SphinxTrain a utilisé des vecteurs acoustiques consiste en 13 coefficients MFCCs et Sphinx a utilisé des MFCCs de 39 en dimensions pour l'apprentissage du modèle. L'extraction des paramètres est réalisée avec l'outil Wave2feat de Sphinx.

D. *Modèles de Markov cachés*

Les modèles de Markov cachés (MMC) (ou HMM en anglais) sont les plus communément utilisés en RAP ces dernières années, Un modèle de Markov caché est un modèle stochastique markovien dans lequel le système modélisé est supposé être un processus de Markov avec des états cachés et dans lequel chaque état est censé produire un ou plusieurs segments stables ou transitoires du signal vocal. A chaque état est associée une distribution de probabilité d'émettre un vecteur spectral donc HHM est un ensemble d'états reliés par des transitions.

Les éléments d'un HMM:

Un modèle de Markov caché λ de N états est défini par le triplet λ = {*A*, *B*, π}, qui est caractérisé par les éléments suivantes :

- N, le nombre d'états dans le modèle et $S = \{s_1, s_2 ..., s_N\}$ C'est l'ensemble de N états cachées contenant un état initial s_1, un état final s_N et des états émettant des symboles. L'état en *t* est noté q_t .
- M, est le nombre d'observations différents du modèle. Les observations correspondent aux sorties du modèle. On les note $O = O_1, ..., O_M$. Le symbole observé au temps *t* est noté O_t .

- π= {πi} désigne un vecteur de distribution des probabilités initiales, où πi est la probabilité se situe dans l'état s_i à l'instant initial.

On a : $$\pi_i = P(q_1 = s_i), 1 \le i \le N \quad (1)$$

- $A = \{a_{ij}\}$ désigne une matrice de distribution des probabilités de transitions entre les états, où a_{ij} signifie la probabilité de passer de l'état i à l'état j, elle peut s'écrire:

$$a_{ij} = P(q_{t+1} = j \mid q_t = i), 1 \le i, j \le N, 1 \le t \le T \quad (2)$$

- $B = \{b_j(o_t)\}$: indique une matrice de distribution des probabilités d'émission à chaque état. Elle se calcule dans le cas que le modèle est discrets par :

$$b_j(o_t) = P(o_t \setminus q_t = j), 1 \le i, j \le N, 1 \le t \le T \quad (3)$$

Et dans le cas que le modèle est continus elle est définie par :

$$b_j(o_t) = \sum_{s=1}^{S} c_{js} Z\left(\mu_{js}, \sum_{js}, o_t\right) \quad (4)$$

Avec c_{js} est la somme des poids, μ_{js} est un vecteur de moyennes et $\sum_{js}$ c'est une matrice de covariance.

E. Système RAP basé sur MMC

Le but principal d'un système de la reconnaissance automatique de la parole basé sur les modèles de Markov cachés c'est de déterminer la probabilité d'une séquence d'observations et de transcrire un signal (le mot le plus probable) sous forme textuelle. Il s'agit donc de déterminer la suite de mots la plus probable parmi l'ensemble des séquences possibles W .

Supposons que X désigne une séquence d'observation acoustique, qui est générée par une séquence de mots de dialecte marocain W donc le problème de reconnaissance de DARIJA marocain est de maximiser la probabilité conditionnelle d'émission de W sachant X correspondant à l'équation suivante :

$$W^* = \underset{W \in L}{\operatorname{argmax}}\, P(W / X) \quad (5)$$

Avec:

$P(W / X)$ c'est la probabilité qu'une suite de mots (ou phrase) prononcés $W = w_1 w_2 .... w_n$ correspond au signal acoustique $X = x_1 x_2 .... x_T$ observées. Le décoder doit sélectionner la séquence de mots W satisfaisant à l'équation 5.

En exploitant la formule de Bayes, $P(W / X)$ peut s'écrire :

$$P(W / X) = \frac{P(X / W).P(W)}{P(X)} \quad (6)$$

Avec :

- $P(W)$ est la probabilité à priori de la suite de mots W .
- $P(X / W)$ est la probabilité du signal acoustique X , étant donné la suite de mots W .
- $P(X)$ est la probabilité du signal acoustique .

Par conséquent, l'Eq. (5) peut être simplifié comme suit :

$$W^* = \arg\max_{W} P(X / W).P(W) \quad (7)$$

donc le system renvoie la séquence de mots optimale W^* par la convergence du ce problème d'optimisation.

Généralement, les Systèmes RAP sont composés de deux éléments essentiels : le modèle acoustique et le modèle du langage.

Le modèle acoustique est déterminé par la probabilité conditionnelle $P(X / W)$ tandis que le modèle du langage est déterminé par la probabilité $P(W)$.

F. Modèle acoustique

Le modèle acoustique est un modèle qui se compose des unités acoustiques qui sont appelés des phonèmes qui forment collectivement le mot. Pour construire le modèle acoustique on utilise les outils Sphinxbase et le Sphinxtrain, chaque fichier audio du corpus est transformé en une séquence de vecteurs caractéristiques qui sont stockés dans un modèle de référence. Pendant la phase d'apprentissage, chaque unité acoustique ou phonème est représenté par un modèle statistique qui décrit la distribution des données. Le signal vocal est transformé en une série de vecteurs de caractéristiques, y compris les coefficients cepstraux (MFCC) [23]–[25]. Dans ce travail, le modèle acoustique a été produit en utilisant un signal vocal provenant de la base de données d'apprentissage « DARIJA_MO ».

G. Modèle de langue

Le modèle de langage accompagne le modèle acoustique pour créer une cohérence linguistique entre les différents éléments acoustiques prononcés. Il définit l'existence de chaque mot du modèle de langue dans un dictionnaire de prononciation. Il va permettre de sélectionner, parmi toutes les séquences de mots possibles, celle qui a la plus grande probabilité d'apparition. Il existe plusieurs types de modèles de langage : le modèle utilisé pour la reconnaissance de mots isolés, le second les grammaires et les modèles de langage statistique de n grammes utilisées pour la forme de la parole libre, et le dernier les modèles de langage statistique phonétique. Le choix d'un modèle de langue dépend de l'application [23].

IV. EXPERIENCES ,RESULTATS ET DISCUSSION

Dans cette section, nous décrivons les différentes expériences et les taches que nous avons réalisées. Nous avons essayé de tester le système en faisant varier les conditions d'environnement selon la base de données qu'on va présenter ensuite et basant sur l'architecture du système présentée dans la section précédente. Les tests sont effectués à l'aide de décodeurs PocketSphinx.

A. Corpus

Notre corpus est constitué de 33 fichiers décomposé de la façon suivante :

Dans ce travail nous préparons notre propre corpus de petite taille des dix salutations les plus célèbres au dialecte marocain dans les conversations téléphoniques nommé DARIJA_MO. Ce corpus de parole enregistré par 60 locuteurs (30 hommes et 30 femmes) la majorité sont des étudiants de l'Université Sultan Moulay Slimane viennent des régions de la ville de Beni Mellal, prennent 1800 expressions dans lesquels chaque locuteur prononce chaque expressions trois fois, cette base de données multi-locuteurs a été enregistrée dans les environnements réels de la vie courante (bruitées : City Bus, Café, salle de photocopie, Cour de faculté). Les signaux du corpus sont pour la plupart dégradés par la présence d'un bruit additif de fond pour avoir été enregistrés dans ces différentes conditions. Pour les données propres on utilise le logiciel Audacity pour le débruitage des fichiers audio dans le but d'améliorer le rapport signal sur bruit et de réduire l'effet du bruit de fond.

Nous avons utilisé différents appareils smartphone pour enregistrer les fichiers audio (audio mono à un taux d'échantillonnage 44 kHz, 16-bit). Nous avons donc obtenu un corpus de 1800 signaux classés par 2/3 données d'apprentissage et 1/3 de test. Dans le tableau 1 nous donnons certains détails techniques et paramètres de DARIJA_MO corpus.

Table 1 : Paramètres d'enregistrement utilisés pour la préparation du corpus Arabic digits.

Process	Description
Nombres des expressions	10
Nombres des expressions propres	$60 \times 10 \times 3 = 1800$
Nombres des expressions bruitées	$20 \times 10 \times 4 = 800$
Base de Traitement	40 locuteurs (20 hommes,20 femmes)
Base de Test	20 locuteurs (10 hommes,10 femmes)
Base de Test bruité	20 locuteurs et 4 bruites $20 \times 10 \times 4 = 800$

Types de bruit courent	4 (Autobus, Café, salle de photocopie, Cour de faculté)
Taille de base de données	1 GB
frequency d'échantillonnage, fs	44000 HZ
Logiciels du mixage et du débruitage	MATLAB R2018a v Trial et Audacity
Appareils de capture	Android mobile phones

Le tableau 2 représente les dix expressions de salutation en Dialecte marocaine utilisé dans des conversation téléphoniques et quotidienne.

Table 2 : Les expressions construis DARIJA_MO corpus

N°	Expression en Dialecte Marocain	Expression en Latin scriptes
1	السلام عليكم	Salam 3alaykom
2	صباح الخير	sbah l5ir
3	كيداير	kidayir
4	لباس عليك	Labas 3elik
5	الحمد لله	lhamdolilah
6	مالين الدار بخير	Malin dar bi5ir
7	كلشي بخير	Kolchi bi5ir
8	أش تتعاود	Ach tat3awed
9	الوليدات	Lwlidat
10	بالسلامة	beslama

La structure des répertoires et des fichiers SphinxTrain pour DARIJA_MO corpus est indiquée sur la figure 3:

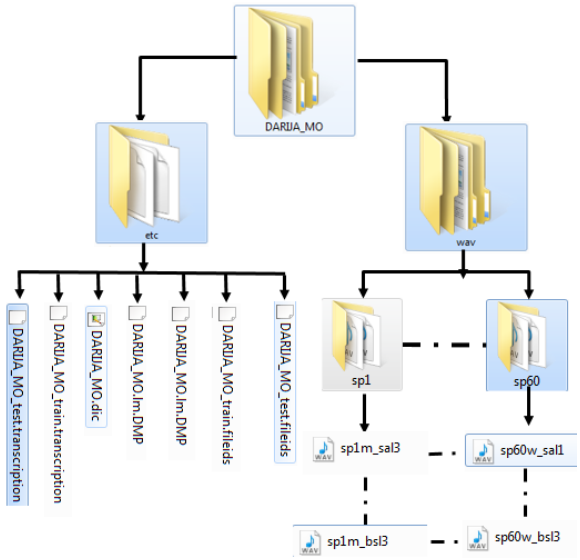

Fig.3 La structure des fichiers SphinxTrain pour la base de données DARIJA_MO.

Le répertoire principal du notre projet basé sur PocketSphinx nommé "DARIJA_MO" a deux dossiers supérieurs, à savoir le etc et le wav. Le répertoire « wav » contient aussi des sous-dossiers, chacun d'eux contient tous les fichiers sonores de chaque locuteur (du locuteur sp1m jusqu'à locuteur sp60w). Le répertoire « etc » contient des fichiers des paramètres de configuration nécessaires pour former le modèle acoustique:

- DARIJA_MO.dic : Dictionnaire phonétique de l'arabe.

- DARIJA_MO.phone : liste de phonèmes

- DARIJA_MO.lm.DMP: modèle de langage

-DARIJA_MO_train.fileids : Liste des fichiers d'apprentissage.

-DARIJA_MO_train.transcription : le fichier de transcription d'apprentissage

-DARIJA_MO_test.fileids : Liste des fichiers à tester

DARIJA_MO_test.transcription : le fichier de transcription pour le test.

Le fichier DARIJA_MO.filler doit avoir les symboles pour le silence. Le contenu du fichier etc/ DARIJA_MO.filler:

<s> SIL
<sil> SIL
</s> SIL

B. Tests réalisées

Dans l'étape d'évaluation du notre système. Nous avons essayé de tester le système en faisant varier les conditions d'environnement selon la base de données qu'on va présenter ensuite et basant sur l'architecture du système présentée dans la section précédente. Toutes les expériences ont été effectuées sur un seul ordinateur avec les spécifications suivantes: processeur Intel® Core (™) i5-4310U @ 2 ,20 GHZ 2,20 GHz × 8Go de RAM, Windows 7 Édition Intégral 64 bits. Nous avons essayé de tester les parties de notre système en utilisant avec la boîte à outils Pocketsphinx en faisant varier l'environnement après avoir construit un modèle acoustique pour un seul mot du DARIJA_Mo corpus. Pour la création de modèles acoustiques est nécessaire un ensemble de vecteurs caractéristiques calculés à partir des données audio d'apprentissage, un pour chaque enregistrement de ce corpus. Nous avons utilisé la technique de mélange du modèle de Markov cachés avec densités multigaussiennes (GMM – HMM) pour générer des modèles acoustiques de chaque mot. Nous avons appliqué l'outil **sphinx_fe** pour convertir des fichiers audio en fichiers acoustiques et Nous avons appliqué l'outil sphinx_fe pour convertir des fichiers audio en fichiers de caractéristiques acoustiques et les avons transformés en une séquence de vecteurs de caractéristiques comprenant les coefficients cepstraux Mel-Frequency (MFCC). L 'apprentissage a été effectuée à l'aide de 1800 expressions de données de la base DARIJA_MO. Le premier ensemble d'expériences a été effectué par des données propres, puis a été testé en ajoutant certains types de bruit. De plus, nous avons compartimenté les performances de notre système lorsque nous changeons le nombre d'états HMM (3 et 5) et les différentes valeurs du mélange gaussien (allant de 4 à 64) dans les conditions mentionnées précédemment. Toutes les modules acoustiques ont été créés avec des donnés propres et testées après avec des données dans des conditions différentes.

Nous avons classé les tests en quatres types : les tests avec des données propres, les tests en direct et les tests dans divers environnements comme expliqué dans les paragraphes suivants.

- La première expérience a été faites avec des données propres.
- La deuxième expérience a été testée avec des données enregistré dans des conditions de bruit réel.
- La troisième expérience c'est le test en direct c.-à-d. en utilisant un microphone dans un environnement spécifique.

- La quatrième expérience : On aussi fait des tests pour savoir l'effet des nombres d'états HMM et le nombre de gaussiennes sur la performance de notre système.

L'évaluation de chaque expérience a été effectuée selon Le taux de reconnaissance des mots (il s'appelle aussi la précision (Accuracy)) est symbolisé par **WAcc** ou **word recognition rate (WRR)** en anglais défini par l'équation suivant :

$$WRR(\%) = \frac{H - I}{N} \times 100\% \quad (8)$$

Où N c'est le nombre total de mots de référence, I est le nombre d'insertions (mots ajoutés) et H est le nombre de mots correctement reconnus. Dans ce cas on considère I=0 puisque on utilise un dictionnaire des mots presque isolés.

C. Résultats des testes

Les performances du modèle de reconnaissance sont évaluées et analysées par le taux d'erreur sur les mots (WRR) mentionné ci-dessus. Ils sont évaluées sur 4 niveaux de rapport signal / bruit (SNR) de 4 types de bruit différents pour les tests dans les conditions du bruit réel et les autres dans les données presque propres ou dans des tests directs. Le tableau ci-dessous illustre les résultats qui sont obtenus à partir de la mise en œuvre du système ASR proposé. Nous avons pris par défaut 8 densités gaussiennes et 3 états par HMM.

Table 3 : Résultats des performances totales du système

Testes		**Résultats**			
	Expressions total		Correct	Erreur	WRR (%)
Test 1 : Dans les conditions Normal (SNR>15 dB)	800		786	14	98.25
Test 2: Dans les conditions bruyantes	Salle de photocopie (SNR 15dB)	800	548	252	68.5
	Cour de faculté (SNR 10dB)	800	397	403	49.62
	Café SNR 6 dB	800	301	499	37.62
	Autobus (SNR 2 dB)	800	145	655	18.12
Test 3 : tests en direct	Locuteur 1 : dans labo étude	50	27	23	54
	Locuteur 2 : Moi-même dans ma maison	50	39	11	78

Pour tester notre système avec un test spécifique on a utilisé cette commande dans le décodeur pocketsphinx :

pocketsphinx_continuous -infile
"C:\ProjectSphinx\Darija\test\test_Bus.wav" -hmm
"C:\ProjectSphinx\Darija\output\Darija_Mo.ci_cont" -dict
"C:\ProjectSphinx\Darija\output\Darija_Mo.dic" -lm
"C:\ProjectSphinx\Darija\output\Darija_Mo.lm.DMP"

Pour tester notre système directement par le microphone de notre machine on a utilisé cette commande dans le décodeur pocketsphinx :

pocketsphinx_continuous -inmic "yes" -hmm
"C:\ProjectSphinx\ Darija \output\Darija_Mo.ci_cont" -dict
"C:\ProjectSphinx\ Darija\output\ Darija_Mo.dic" -lm
"C:\ProjectSphinx\ Darija\output\ Darija_Mo.lm.DMP"

1) Test 4:l'effet du nombre de gaussiennes

Pour tester l'influence du changement du nombre de distributions de probabilité gaussiennes sur les performances du système, ce dernière a été formé et testé pour différentes valeurs gaussiennes allant de 2 à 64 par un HMM de 3 états. Le tableau 4 et la figure 4 successivement présentent les résultats des expériences décrites ci-dessus sur les taux de reconnaissance dans la partie de test de notre base de données DARIJA_MO dans conditions propres et bruités en variant le nombre de GMM.

Table 4 : Taux de reconnaissance comparatifs de l'effet des conditions réels en fonction du nombre de gaussiennes pour 3 états par HMM.

Nombre de gaussiennes GMM	Base de test Normal SNR>15	Base de test bruité				
		Salle photoC 15 dB	Faculté 10 dB	Café 6 dB	Autobus 2 dB	**Moyenne (SNRs)**
2	95.5	62.38	44	32.5	15	**49.88**
4	96.62	65.63	46	34.75	17.13	**52.03**
8	**98.25**	**68.5**	**49.62**	**37.62**	**18.12**	**54.42**
16	97.12	67.5	47.5	39.25	14.66	**53.21**
32	97.38	67.38	48.63	36.38	18.25	**53.60**
64	97.5	67.12	46.63	39	19.75	**54**

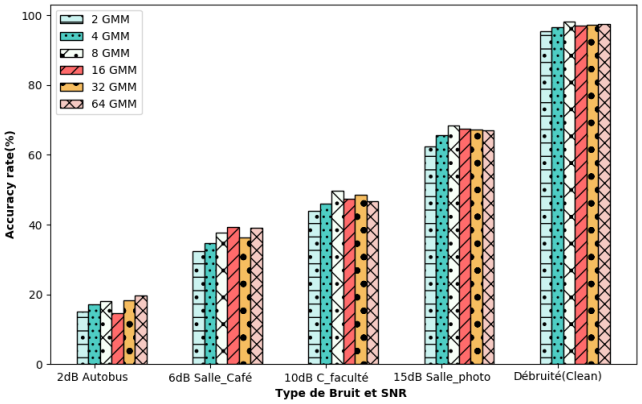

Fig.4 Evolution du WRR en fonction du nombre de Gaussiennes par un HMM à 3 états.

2) Test 4: l'effet du Nombre d'États par HMM

Pour savoir l'effet du nombre d'états HMM sur la performance de notre système au niveau des modèles acoustiques, le système a été piloté en utilisant un état puis trois états puis cinq états par HMM à 8 GMM. Les résultats des trois tests sont présentés consécutivement dans le tableau 5 et qui sont exposés dans la figure 5.

Table 5 : Taux de reconnaissance comparatifs de l'effet des conditions réels en fonction du nombre d 'états par HMM

Nombre des états HMM	Base de test Normal SNR>15	Base de test bruité				
		Salle photoC 15 dB	Faculté 10 dB	Café 6 dB	Autobus 2 dB	**Moyenne (SNRs)**
1	73.5	46.13	34.25	24.88	14.5	**38.65**
3	98.25	68.5	49.62	37.62	18.12	**54.62**
5	87.62	58.88	47.38	30.5	16.75	**48.23**

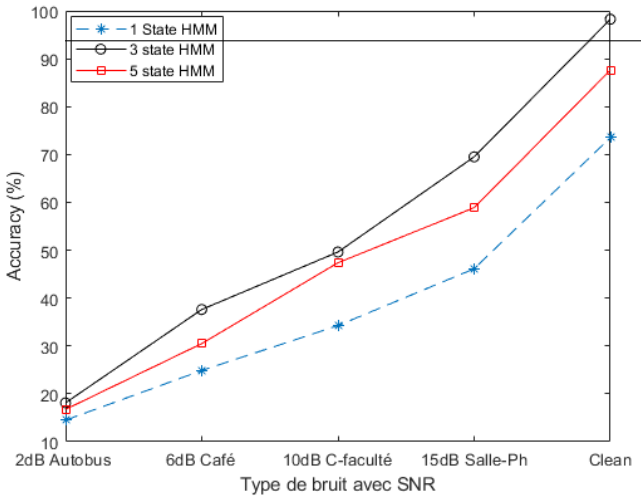

Fig.5 Evolution du (WRR %) en fonction du nombre d'états HMM et les conditions du bruit pour GMM=8.

D. Interprétation et Discussion

A note préliminaire, les résultats généraux obtenus d'après les tableaux 3 ,4 et 5 et la figure 5 montrent une relation presque linéaire entre le taux de reconnaissance de mots (WRR) et le niveau de bruit. Le taux diminue proportionnellement au niveau de bruit, ce qui indique que le modèle acoustique change avec la même intensité que le niveau de bruit est ajouté à l'ensemble de test, sans aucune dépendance du type de bruit. Cela signifie que n'importe quel modèle est capable d'apprendre les modèles du bruit mieux que l'autre.

Le pire taux enregistré pour le système est dans le bruit Autobus et le bruit de salle de café par un taux très faible de reconnaissance de 18,12% et 37,63% respectivement (tableau 3). Pendant que, le bruit produit dans une salle de photocopie ou le test fait directement avec le microphone dans la maison n'affecte pas beaucoup sur la précision de reconnaissance comparant avec les deux types précédents, il a atteint dans ce cas 68,5% et 78% successivement. Les autres bruits (cour de la faculté, salle de laboratoire) ont des tendances presque similaires. Cela peut s'expliquer par le babillage plus dans le bus et le café, où les mots affectent les modèles acoustiques comparant avec les autres (salle de photocopie, maison, cour de la faculté, laboratoire) ont des niveaux de bruit inférieurs. Dans les conditions propre (réduction du bruit naturel des donnés) le système ASR général obtient un taux d'erreur WRR de 98,25%.

Dans le tableau 4 et la figure 4 (Test 4), nous pouvons observer aussi que, l'accroissement du nombre de gaussiennes fait augmenter le taux de reconnaissance, les meilleures valeurs de WRR obtenue sont pour 8 GMM et à 3 état par HMM. Dans le tableau 5 et la figure 5 (Test 5), nous extrayons que le changement du nombre d'états HMM affecte de manière significative les résultats obtenus. Par conséquent, les résultats favorables sont atteints pour le cas de 3 états par HMM.

Le travail présenté a été comparé aux travaux similaires existants. Dans le document, [24]. Les auteurs ont présenté un système de reconnaissance vocale des chiffres Amazighs dans un environnement automobile bruyant. Ils ont utilisé le modèle de Markov caché pour modéliser les unités phonétiques correspondant aux mots extraits de la base d'apprentissage sous l'open source CMU Sphinx 4. Les résultats expérimentaux pour 1800 des données présentent un taux de reconnaissance de 88,22% dans un environnement propre, et des taux de 59,26% et 33,83% en condition bruyante à SNR 10 dB et 20 dB, respectivement. Dans le papier, [16] ils ont proposé un système de reconnaissance vocale pour de la langue Amazigh basant sur l'open source CMU Sphinx-4. Le système proposé comprend les étapes L'extraction des caractéristiques, la modélisation acoustique, la prononciation et la modélisation à l'aide de HMM. La taille de la base de données pour ce travail est de 2970 mots et produit 88% de précision. Dans l'étude scientifique, [14].Ils ont construit un système d'identification du locuteur en dialecte marocain basant sur MFCC pour extraire les caractéristiques de la parole inconnue et les modèles de Markov cachés pour la classification .Le code a été développé dans MATLAB qui arrivais à identifier le locuteur de manière satisfaisante.

Notre travail poursuivre les résultats des travaux connexes et similaires existants, la performance du système est très satisfaisante (98,12% WRR dans les propres conditions et d'une moyenne de 50,98% WRR dans les différentes conditions de bruit). Compte tenu, de la petite taille des données et qu'on a construit le modèle de langue et le modèle acoustique du dialecte arabe marocaine "DARIJA" comme une nouvelle langue. Donc, il est considéré parmi les premiers travaux traitant cette dialecte utilisant la librairie PocketSphinx.

V. Conclusion

Dans cet article, le système de reconnaissance automatique du dialecte marocain a été développé. Ce système est basé sur des modèles HMM de Markov cachés avec des mélanges gaussiens GMM pour générer des modèles acoustiques à l'aide du système PocketSphinx. Nous avons élaboré le corpus DARIJA_MO du dialecte marocain contient 1800 des donnés. Nos expériences et les tests fournissent une évaluation des performances la reconnaissance vocale de dialecte marocain des environnements bruyants réels à différents niveaux de décibels. Ce système a atteint un résultat de 98,12% WRR en réduisant le bruit de fond dans les données de traitement et il a atteint d'une moyenne de 50,98% WRR dans les différentes conditions de bruit. Considérant, les paramètres idéals dans ce cas c'est 8 de GMM et à 3 état par HMM. Le système est également montré que lorsque l'intensité de bruit augmente, le taux de reconnaissance diminue également et le modèle entraîné avec une parole propre donne des résultats considérablement meilleurs avec des niveaux de bruit plus faibles. C'est la raison pour laquelle ce type de modèle entraîné est principalement utilisé par les systèmes de reconnaissance vocale. Dans nos travaux futurs, nous pouvons étendre l'application pour augmenter la taille du vocabulaire Darija_Mo pour un système vocal connecté et continu. Nous pouvons aussi tester le système par d'autres types de bruit additif à différentes valeurs de SNR et pour d'autres approches comme DNN et autres boîtes à outils comme tensorflow.

References

[1] Yu, Dong and Deng, Li, *Automatic Speech Recognition - A Deep Learning Approach*, 1st ed. Springer-Verlag London, 2015.

[2] M. Wolf and C. Nadeu, 'Evaluation of different feature extraction methods for speech recognition in car environment', in *2008 15th International Conference on Systems, Signals and Image Processing*, Jun. 2008, pp. 359–362.
[3] Kohshelan and N. Wahid, 'Improvement of Audio Feature Extraction Techniques in Traditional Indian Musical Instrument', in *Recent Advances on Soft Computing and Data Mining*, Cham, 2014, pp. 507–516.
[4] S UMESH and Sadhana, 'Studies on inter-speaker variability in speech and its application in automatic speech recognition | SpringerLink', *Springer*, pp. 853–883, 2011.
[5] V. Kadyan, A. Mantri, and R. K. Aggarwal, 'Improved filter bank on multitaper framework for robust Punjabi-ASR system', *Int. J. Speech Technol.*, vol. 23, no. 1, pp. 87–100, Mar. 2020.
[6] D. E. Sturim, W. M. Campbell, and D. A. Reynolds, 'Classification Methods for Speaker Recognition', in *Speaker Classification I: Fundamentals, Features, and Methods*, C. Müller, Ed. Berlin, Heidelberg: Springer, 2007, pp. 278–297.
[7] H. H. O. Nasereddin and A. A. R. Omari, 'Classification techniques for automatic speech recognition (ASR) algorithms used with real time speech translation', in *2017 Computing Conference*, Jul. 2017, pp. 200–207.
[8] Rabiner, L and Juang. B, 'Fundamentals of Speech Recognition', *PTR Prentice Hall (Signal Processing Series)*, Englewood, 1993.
[9] Jean-Paul HATON, Christophe Cerisara, Dominique Fohr, Yves Laprie, and Kamel Smaïli, *Reconnaissance Automatique de la Parole Du signal à son interprétation*, Dunod. Techniques de l'Ingénieur, 2006.
[10] D. Huggins-Daines, M. Kumar, A. Chan, M. Ravishankar, A.I. Rudnicky, and A.W. Black, 'Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices - IEEE Conference Publication', Toulouse, France, 2006.
[11] H. Satori, M. Harti, and N. Chenfour, 'Arabic Speech Recognition System Based on CMUSphinx', in *2007 International Symposium on Computational Intelligence and Intelligent Informatics*, Mar. 2007, pp. 31–35.
[12] M. Y. El Amrani, M. M. H. Rahman, M. R. Wahiddin, and A. Shah, 'Building CMU Sphinx language model for the Holy Quran using simplified Arabic phonemes', *Egypt. Inform. J.*, vol. 17, no. 3, pp. 305–314, Nov. 2016.
[13] Mohamed Elmahdy, Mark Hasegawa-Johnson, and Eiman Mustafawi, 'A Baseline Speech Recognition System for Levantine Colloquial Arabic', 2012.
[14] B. Mouaz, B. H. Abderrahim, and E. Abdelmajid, 'Speech Recognition of Moroccan Dialect Using Hidden Markov Models', *Procedia Comput. Sci.*, vol. 151, pp. 985–991, Jan. 2019.
[15] Azzedine Touazi and Mohamed Debyeche, 'An experimental framework for Arabic digits speech recognition in noisy environments', *International Journal of Speech Technology*, pp. 205–224, 2017.
[16] M. Telmem and Y. Ghanou, 'Amazigh Speech Recognition System Based on CMUSphinx', in *Innovations in Smart Cities and Applications*, Cham, 2018, pp. 397–410.
[17] EL GHAZI, A, DAOUI, C., and IDRISSI, N, 'AUTOMATIC SPEECH RECOGNITION SYSTEM CONCERNING THE MOROCCAN DIALECTE (Darija and Tamazight)', *International Journal of Engineering Science & Technology*, p. 966, Mar. 2012.
[18] F. M. Aloqayli and Y. A. Alotaibi, 'Spoken Arabic Vowel Recognition Using ANN', in *2017 European Modelling Symposium (EMS)*, Nov. 2017, pp. 78–83.
[19] S. El Ouahabi, M. Atounti, and M. Bellouki, 'Building HMM Independent Isolated Speech Recognizer System for Amazigh Language', in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Cham, 2017, pp. 299–307.
[20] Y. A. Alotaibi, 'Comparing ANN to HMM in implementing limited Arabic vocabulary ASR systems', *Int. J. Speech Technol.*, vol. 15, no. 1, pp. 25–32, Mar. 2012.
[21] F. S. Al-Anzi and D. AbuZeina, 'Exploring the language modeling toolkits for Arabic text', in *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Nov. 2017, pp. 1–4.
[22] W. Helali, Z. Hajaiej, and A. Cherif, 'Arabic corpus implementation: Application to speech recognition', in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, Mar. 2018, pp. 50–53.
[23] O. Zealouk, H. Satori, M. Hamidi, and K. Satori, 'Pathological Detection Using HMM Speech Recognition-Based Amazigh Digits', in *Embedded Systems and Artificial Intelligence*, Singapore, 2020, pp. 281–289.
[24] O. Zealouk, M. Hamidi, H. Satori, and K. Satori, 'Amazigh Digits Speech Recognition System Under Noise Car Environment', in *Embedded Systems and Artificial Intelligence*, Singapore, 2020, pp. 421–428.
[25] N. Shmyrev, 'Training an acoustic model for CMUSphinx', *CMUSphinx Open Source Speech Recognition*. http://cmusphinx.github.io/wiki/tutorialam/.

Liste des auteurs