Conception d'un récepteur AIS détectant les falsifications de messages : développement de stratégies et prototypage sur FPGA

Maelic Louart

COLLEGES SCIENCES
BRETAGNE POUR L'INGENIEUR
LOIRE ET LE NUMERIQUE

ENSTA BRETAGNE

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE
DE TECHNIQUES AVANCÉES BRETAGNE

ÉCOLE DOCTORALE N° 648
Sciences Pour l'Ingénieur et le Numérique
Spécialité : *Sciences et technologies de l'Information et de la Communication*

Par
Maelic LOUART

Conception d'un récepteur AIS détectant les falsifications de messages : développement de stratégies et prototypage sur FPGA

Thèse présentée et soutenue à l'École Navale, le 07 juillet 2023
Unité de recherche : Le LabSTICC (pôle SHARP, équipe ARCAD) et l'IRENAV (équipe MoTIM)

Rapporteurs avant soutenance :

François AUGER, Professeur des Universités (Université de Nantes)
Christophe JEGO, Professeur des Universités (Bordeaux INP / ENSEIRB-MATHMECA)

Composition du Jury :

Examinateurs :	Elsa DUPRAZ	Maître de conférences, IMT-Atlantique
	Jean-Yves TOURNERET	Professeur des Universités, Toulouse INP/ENSEEIHT
	Daniel MENARD	Professeur des Universités, INSA de Rennes
Enc. de thèse :	Jean-Christophe LE LANN	Maître de conférences, ENSTA Bretagne
Enc. de thèse :	Jean-Jacques SZKOLNIK	Ingénieur de Recherche, Ecole Navale
Dir. de thèse :	Abdel-Ouahab BOUDRAA	Professeur des Universités, Ecole Navale & ENSAM

Invités :

Amer BAGHDADI	Professeur des Universités, IMT-Atlantique
Frédéric LE ROY	Maître de conférences, ENSTA Bretagne

Remerciements

Tout d'abord, je souhaite commencer mes remerciements en les adressant à mes quatre encadrants de thèse : Abdel-Ouahab BOUDRAA, Jean-Jacques SZKOLNIK, Jean-Christophe LE LANN et Frédéric LE ROY.

Je remercie Abdel-Ouahab BOUDRAA, mon directeur de thèse, pour le temps qu'il a passé, parfois durant le week-end, à corriger mes articles et mon manuscrit. Il se rendait disponible malgré ses cours à l'Ecole Navale et les sept autres doctorants qu'il encadrait. Ses conseils et corrections m'ont permis d'améliorer la qualité de mes articles et ont rendu leur rédaction et soumission plus simple. Par ailleurs, nous avons eu régulièrement des discussions très intéressantes sur des sujets divers comme la science, la politique, et même parfois la théologie, discussions dont je garde un bon souvenir et me souviendrai longtemps.

Je remercie Jean-Jacques SZKOLNIK pour l'aide précieuse qu'il m'a apportée, en particulier sur le plan technique. Cela faisait déjà plusieurs années qu'il mettait en place des expérimentations à l'École Navale pour enregistrer des signaux et messages AIS et qu'il développait des algorithmes. L'exploitation de ses travaux m'a fait gagner beaucoup de temps et m'a permis de tester mes algorithmes sur des données réelles. En plus, grâce à son expérience et son recul sur le sujet de la détection des falsifications de messages AIS, j'ai pu rapidement, sur ses indications, dégager un plan de thèse et orienter mes recherches dans la bonne direction. Pour finir, du fait de sa grande disponibilité, j'ai travaillé avec lui en étroite collaboration (mise en place d'expérience, discussion et exploitation des résultats), et ai fréquemment profité de ses conseils et de son avis sur les stratégies que je développais et les résultats obtenus. Cette collaboration étroite a été un des aspects les plus intéressants de ma thèse.

Je remercie également Jean-Christophe LE LANN qui m'a dirigé, à l'ENSTA Bretagne, sur la partie concernant l'implémentation du récepteur AIS sur FPGA. L'implémentation de systèmes sur FPGA était le domaine dans lequel j'avais le moins de connaissances et pour lequel j'ai dû fournir le plus d'efforts. Heureusement, grâce aux connaissances de Jean-Christophe dans ce domaine et à ses qualités de pédagogue, j'ai pu rapidement rattraper ce retard et réussir à générer l'architecture matérielle complète du récepteur AIS que j'avais développé durant mes recherches. Par ailleurs, son niveau d'exigence élevé m'a obligé à me dépasser durant ces trois années et à revoir ma méthode de travail afin d'être plus efficace. Cela a permis de cloturer cette thèse dans les temps avec un nombre d'articles publiés suffisants.

Pour finir, je souhaiterais remercier Frédéric LE ROY, qui bien qu'étant déjà très

occupé par ses deux autres doctorants, prenait néanmoins le temps de répondre à mes questions en radio-logicielle lorsque je venais le solliciter.

Je remercie également mes rapporteurs de thèse François AUGER et Christophe JEGO pour la lecture attentive de mon manuscrit. Leurs rapports m'ont permis de corriger quelques erreurs de forme et imprécisions qui m'avaient échappées et d'améliorer la qualité du manuscrit. Plus généralement, je remercie tous les membres du jury d'avoir accepté d'évaluer mes travaux et fait le déplacement à l'École Navale pour assister à ma soutenance. Ce jour là, leur attitude bienveillante m'a permis de présenter mes travaux sans stresser. Je suis aussi reconnaissant envers les membres du CSI, Amer BAGHDADI et Thierry CHONAVEL, qui ont suivi et évalué mes travaux pendant ces trois années en apportant un point de vue extérieur.

Je remercie également les personnes chargées de l'intendance de ma thèse : Gael MONOT pour l'École Navale et Annick BILLON-COAT et Patricia CABEL pour l'ENSTA Bretagne. La gestion de toute la logistique liée à ma soutenance par Gael MONOT m'a déchargé de beaucoup de préoccupations.

Par ailleurs, je tiens aussi à remercier l'ensemble de mes collègues de bureau. En premier lieu mes deux co-doctorants Grégoire DE BROGLIE et Louis MORGE-ROLLET. Grégoire m'a beaucoup aidé au début de ma thèse pour déboguer les nombreuses erreurs que je rencontrais avec le système d'exploitation Linux qui ne m'était pas familier, quitte à passer parfois plusieurs heures d'affilée dessus avec moi. Louis, quant à lui, m'a beaucoup aidé durant la deuxième moitié de ma thèse, lorsque je travaillais sur les signatures radiométriques des transpondeurs AIS. Sa grande disponibilité et sa maîtrise du sujet m'ont permis de rapidement trouver une signature discriminante et d'écrire un article sur le sujet. Ensuite, je tiens aussi à remercier Théotime BOLLENGIER pour son expertise en implémentation de FPGA et son aide pour développer le test bench vérifiant le comportement du récepteur AIS que j'avais conçu. J'ai aussi une pensée particulière pour Paul FRANÇOIS qui m'a initié au surf sur la presqu'île de Crozon, Quentin SAINT-CHRISTOPHE avec qui j'ai eu le plaisir de participer à la conférence du Gretsi à Nancy et Guillaume FASSE pour les nombreuses discussions intéressantes à l'École Navale. Pour finir, je tiens à remercier chaleureusement Estelle KERBRAT, Bastien DROUOT et Van Dong DO qui, avec Grégoire et Louis cités plus haut, après avoir assisté à la présentation, m'ont fait la joie de revenir le soir à Lanvéoc pour prendre part à mon deuxième pot de thèse, ce qui a permis de clôturer cette journée joyeusement. Je retiendrai en particulier de cette soirée le *one-man-show* improvisé par Bastien qui a bien amusé ma famille.

Je réserve la fin de ces remerciements aux personnes qui me sont le plus chères : ma famille. Je les remercie pour leur soutien durant ces presque quatre années de thèse. J'ai une pensée particulière pour mes grands-parents qui ont fait le déplacement à l'École Navale, malgré leur âge avancé. Je remercie mes tantes Cécile et Perrine et mes parents pour leur présence à la soutenance. Par ailleurs, ils m'ont suppléé avec brio dans la pré-

paration du pot de thèse : les plats cuisinés étaient délicieux en plus d'être nombreux. Je remercie ma mère pour sa relecture de tout mon manuscrit afin de corriger quelques fautes d'orthographe qui m'avaient échappées.

RÉSUMÉ EN FRANÇAIS

L'AIS est un système de communication pour bateaux très répandu. Il permet l'échange automatique d'informations de navigation comme la position, la vitesse, la route, l'identité, le port de départ et d'arrivé, etc. Ces informations aident à la navigation, sécurisent le trafic maritime et simplifient la surveillance des autorités de contrôle. Cependant, ayant été développé dans les années 90, l'AIS est peu sécurisé. Il est facile pour un utilisateur malveillant d'émettre de fausses informations, de brouiller les communications ou bien de faire apparaître des bateaux fantômes. L'intérêt peut être de cacher des activités frauduleuses, perturber le trafic ou justifier de prétendues "violations de territoires".

Partant de ce constat, dans ce manuscrit, trois stratégies ont été développées pour détecter ces attaques. La première suit les bateaux avec un filtre IMM pour détecter les falsifications de position et de vitesse. La deuxième vérifie le respect, par les bateaux, du mode d'accès TDMA lorsqu'ils communiquent. La vérification de ce mode d'accès, spécifié par la norme AIS, permet de détecter l'émission de faux messages et la création de bateaux fantômes. Enfin, la dernière stratégie calcule une signature radiométrique, à partir des signaux transmis par chaque transpondeur, afin de les identifier matériellement et non par l'identité qu'ils transmettent dans leurs messages. Cela permet de détecter les usurpations d'identité. Ces trois stratégies sont appliquées conjointement et ont été testées avec des signaux réels enregistrés dans la rade de Brest. Des taux d'erreur de première et de deuxième espèce respectivement de 0.0035 et 0.0171 sur la détection de l'usurpation d'identité ont été obtenus. Par ailleurs, des simulations de Monte Carlo ont montré une sensibilité aux falsifications de position allant de 35m à 250m.

Une partie de ces stratégies a été implémentée sur un système embarqué de type FPGA pour être utilisée par la douane ou la marine française. Toute la chaîne de démodulation des signaux AIS a aussi été implémentée sur la même carte pour récupérer les signaux et les informations des messages utilisés par ces stratégies. La génération du code implémentant le FPGA s'est faite en utilisant la HLS qui est un outil de synthèse haut niveau. Ce cas d'application a montré que cet outil était aujourd'hui une technologie mature pouvant synthétiser efficacement et automatiquement des systèmes complets de grande taille à partir d'une modélisation haut niveau, supprimant ainsi la séparation qui existait autrefois entre les activités de traitement de signal et les activités de conception matérielle. Le récepteur AIS intelligent ainsi conçu peut, à partir de signaux en bande de base, détecter, en temps réel, les falsifications de position.

Pour finir, un laboratoire virtuel sur FPGA, simulant, en même temps, le récepteur et son environnement, a été aussi développé. Il permet de générer facilement des scé-

narios de test variés afin de vérifier le comportement du récepteur dans ses conditions limites d'utilisation. La considération conjointe du récepteur, de son environnement et de leurs interactions nous a amené à parler des systèmes cyber-physiques et des difficultés, toujours actuelles, associées à la modélisation et la simulation de tels systèmes. Une nouvelle méthode a été proposée pour résoudre ces difficultés. Cette méthode modélise le CPS comme un ensemble d'acteurs s'exécutant de manière concurrente, utilise le FPGA comme plateforme de simulation et applique la HLS pour générer automatiquement la description matérielle à partir de la modélisation à base d'acteurs. Un laboratoire virtuel a été créé sur FPGA pour vérifier le comportement de notre récepteur. Il permet de simuler en même temps notre récepteur et son environnement maritime, pour lequel le nombre de bateaux communiquant, leur trajectoire, le SNR de leurs signaux transmis et le CFO de leur transpondeur peuvent être modifiés. En plus, des falsifications de position et des usurpations d'identité peuvent aussi être ajoutées aux messages échangés durant la simulation.

TABLE DES MATIÈRES

TABLE DES FIGURES

Liste des tableaux

Acronymes

ADC	Analog-to-digital converter.
ADS-B	Automatic dependent surveillance–broadcast.
AIS	Automatic identification system.
ASIC	Application-specific integrated circuit.
BRAM	Bloc RAM.
C.T.	Continuous time.
CA	Constant acceleration.
CDFG	Control data flow graph.
Cerema	Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement.
CFG	Control flow graph.
CFO	Carrier frequency offset.
CPS	Cyber-physical system.
CPU	Central processing unit.
CRC	Contrôle de redondance cyclique.
CT	Coordinated turn.
CV	Constant velocity.
CWNA	Continuous white noise acceleration.
DE	Discret event.
DFG	Data flow graph.
DSE	Design space exploration.
DSP	Digital signal processing.
EDA	Electronic design automation.
EGM96	Earth gravitational models 1996.
EKF	Extended Kalman filter.
ESL	Electronic system level.
FATDMA	Fixed acces time division multiple access.
FF	Flip-flop.
FFT	Fast Fourier transform.
FHT	Full hypothsis tree.
FIFO	First in first out.

FMI	Functional mock-up interface.
FPGA	Field programmable gate arrays.
GMSK	Gaussian minimum-shift keying.
GNSS	Global navigation satellite systems.
GPB	Generalized pseudo bayesian.
GPS	Global positioning system.
GPU	Graphics processing unit.
HDLC	High-level data link control.
HIL	Hardware-in-the-loop.
HLS	High level synthesis.
IALA	International association of marine aids to navigation and lighthouse authorities.
IE	Input estimation.
IMM	Interacting multiple model.
IMO	International maritime organization.
IOT	Internet of things.
IP	Intellectual property.
ITDMA	Incremental time division multiple access.
ITU	International telecommunication union.
KF	Kalman filter.
KPN	Khan process networks.
LNA	Low noise amplifier.
LSB	Less significant bit.
LUT	Look-up table.
MBD	Model-based design.
MC	Monte-Carlo.
MMSI	Maritime mobile service identity.
MOC	Model of computation.
MSB	Most significant bit.
NI	Nominal increment.
NMEA	National marine electronics association.
NRZI	Non return to zero inverted.
NS	Nominal slot.
NSS	Nominal start slot.
NTS	Nominal transmission slot.
OSI	Open systems interconnection.
PIPO	Ping-pong.

RATDMA	Random acces time division multiple access.
RI	Reporting interval.
RMSE	Root-mean-square error.
RTL	Register transfer level.
SAR	Synthetic-aperture radar.
SI	Selection interval.
SIL	Software-in-the-loop.
SNR	Signal-to-noise ratio.
SoC	System on chip.
SOG	Speed over ground.
SOLAS	Safety of life at sea.
SOTDMA	Self-organized time division multiple access.
SR	Synchrone-reactif.
STO	Slot time-out.
TCXO	Temperature compensated X (crystal) oscillator.
TDMA	Time division multiple access.
TOA	Time of arrival.
TS	Time slot.
USRP	Universal software radio peripheral.
UTC	Coordinated universal time.
VHDL	Very high speed integrated circuit hardware description language.
VHF	Very high frequency.
WGS84	World geodetic system 1984.

Introduction

Contexte général

Sécurisation du transport maritime

Ces dernières années, par l'intensification des communications et des échanges entre les états, la mondialisation a eu des effets profonds sur les économies et les sociétés du monde entier. Elle a par exemple favorisé la croissance économique et la création d'emplois, ainsi que l'accès à une gamme plus large de biens et de services. Ce phénomène a entraîné un essor du commerce international, avec une augmentation du transport de marchandises de 6 % par an, en moyenne, entre 1950 et 2019. Cette augmentation a été en grande partie rendue possible grâce au transport maritime, qui a un rôle clé dans cette interconnexion généralisée du monde, rôle bien plus important que celui joué par le transport routier ou aérien [44]. En effet, aujourd'hui, plus de 80 % du commerce international, en volume, est acheminé par voie maritime, ce qui représente plus de 10 milliards de tonnes de marchandises transportées par an [43].

L'utilisation des voies maritimes pour l'échange de personnes et de marchandises est très ancien. Très tôt, des règles pour sécuriser et fiabiliser ce moyen de transport furent édictées. Les premières règles connues furent imposées par les Égyptiens, durant l'Antiquité, pour réguler le trafic sur le Nil. Plus tard, au fil des siècles, de nouvelles règles plus élaborées sont apparues, en réponse à l'augmentation de la navigation maritime et à la densification du trafic. Par exemple, au XVIIe siècle, des phares, des bouées et des signaux ont été érigés le long des côtes pour aider les marins à naviguer en toute sécurité. Au XXe siècle, la sécurité maritime est devenue une préoccupation encore plus importante suite à l'expansion du commerce international et l'explosion du trafic maritime. Des règles de sécurité plus strictes ont donc été promulguées pour les navires, les équipages et les ports, et des organisations internationales ont été créées pour surveiller et réglementer la sécurité maritime mondiale.

Parmi ces organisations internationales, on compte notamment l'IMO (International maritime organization) et l'IALA (International association of marine aids to navigation and lighthouse authorities), qui furent respectivement créées en 1948 et 1957. De la collaboration de ces deux organismes avec l'ITU (International telecommunication union) est née, à la fin des années 90, l'AIS (Automatic identification system). L'AIS est un système d'échange automatique d'informations de navigation en temps réel, comme présenté sur la figure 1. Ce système fut développé pour limiter les risques d'accidents [4] et de collisions

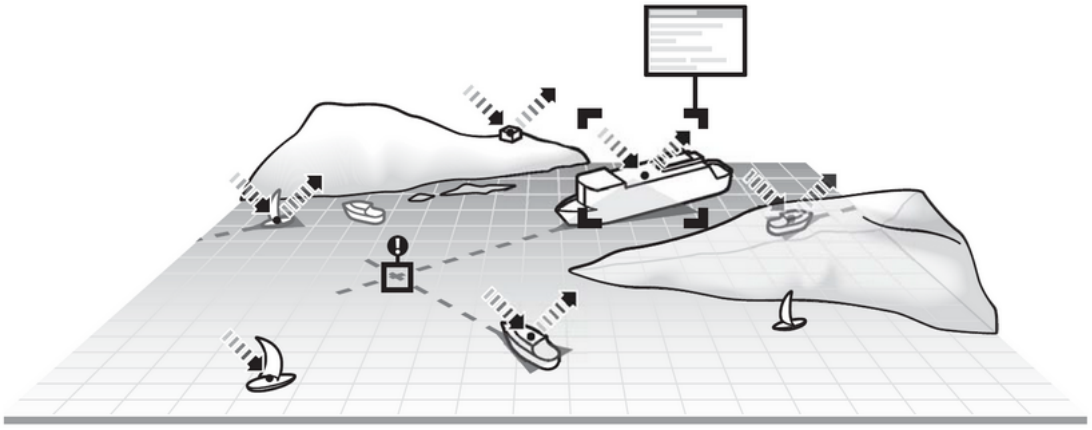

FIGURE 1 – Schéma représentant plusieurs bateaux et une station de base communiquant entre eux en utilisant un AIS [39]

entre navires [112], et pour améliorer l'assistance et le sauvetage des navires en difficulté [161]. Il est installé sur différentes catégories de navires pour aider à la navigation et à la surveillance du trafic maritime [139], mais aussi dans les ports et le long des côtes pour former un réseau de balises côtières sur des positions à risque et d'intérêt.

Caractéristiques techniques de l'AIS

Les informations échangées par l'AIS peuvent être classées en trois catégories [133] :

1. des informations statiques (numéro MMSI (Maritime mobile service identity), nom, numéro IMO, longueur, largeur, etc.) ;
2. des informations dynamiques (position, vitesse, route, cap, etc) obtenues grâce à un récepteur GPS (Global positioning system) raccordé à l'AIS ;
3. des informations liées au trajet (port de départ, port de destination, nature de la cargaison, etc.).

Ce système est aujourd'hui la source principale des informations de navigation utilisée pour connaître l'activité maritime et représente la plus grande avancée dans l'aide à la navigation depuis l'introduction des radars. Depuis 2002 et l'accord de SOLAS (Safety of life at sea) de l'IMO, son utilisation est obligatoire pour tous les navires internationaux d'une taille supérieure à 300 tonneaux (>850 m^3 et les navires transportant des passagers dans les eaux internationales [61]. Aujourd'hui, environ 500 000 navires utilisent l'AIS pour leurs opérations régulières [154]. La Figure 2 représente le trafic maritime en date du 15 mai 2023 à 10 h 12 min. On note la forte densité des routes maritimes. Il existe un système équivalent pour la circulation aérienne qui est l'ADS-B (Automatic dependent surveillance–broadcast).

L'échange de données se fait automatiquement en VHF (Very high frequency) sur deux canaux dédiés (161,975 MHz et 162,025 MHz) à l'aide d'un transpondeur. Il existe deux

types d'AIS, les transpondeurs de classe A, imposés par l'accord SOLAS, sur lesquels nous nous pencherons plus particulièrement dans ces travaux, et les transpondeurs de classe B, utilisés par les navires de petite taille, notamment les bateaux de plaisance. Les AIS de classe A sont plus onéreux et notablement plus puissants (12,5 W). Leur portée sur un navire varie entre 25 et 40 km, et dépend de nombreux facteurs, tels que l'altitude relative des antennes, leur type respectif et les conditions météorologiques. Les émissions se font toutes les 2 à 10 s lorsque le bateau est en mouvement, et toutes les 3 min lorsqu'il est à l'arrêt. Les AIS de classe B sont moins performants et moins onéreux, ils sont essentiellement utilisés par les bateaux de plaisance et les petits navires de pêche pour faciliter leur navigation et améliorer leur sécurité, notamment en limitant le risque d'abordage. Les périodes d'émission sont plus longues et la puissance plus faible (5 W).

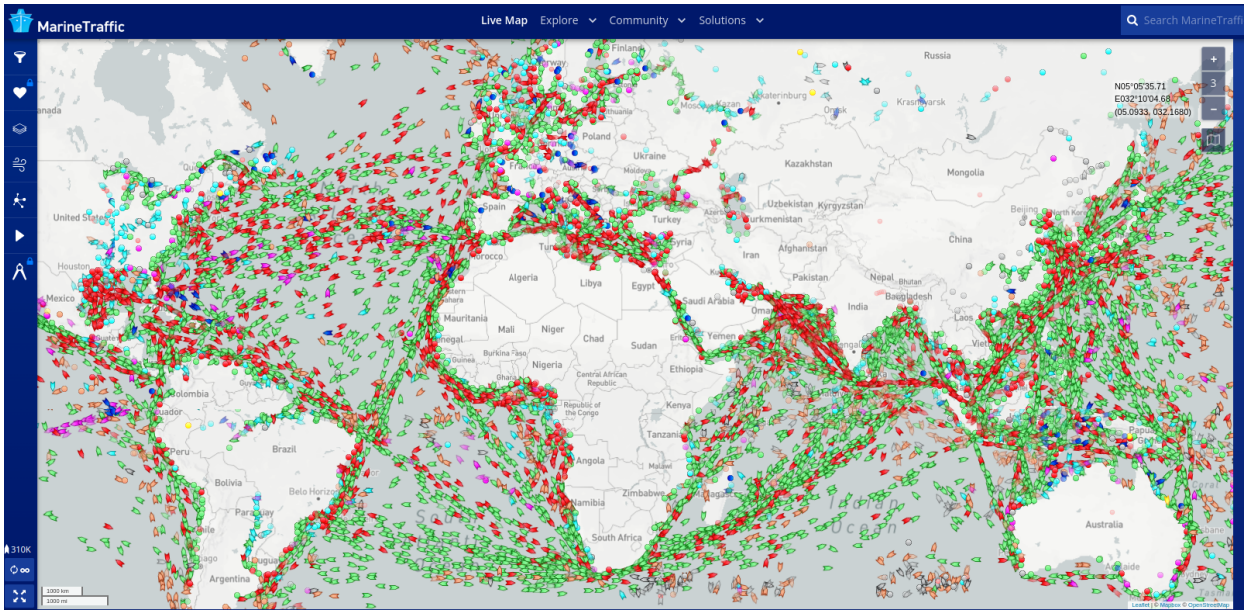

FIGURE 2 – Capture d'écran du site Marine Traffic [100]. Circulation maritime du 15 mai 2023 à 10 h 12 min. Vert : cargos, rouge : bateaux-citerne, bleu foncé : navires à passagers, bleu clair : remorqueurs et pousseurs, violet : bateaux de plaisance et marron : bateaux de pêche.

Problématiques

Vulnérabilités de l'AIS

Malgré une utilisation croissante, l'AIS ne comporte aucune fonction quant au contrôle d'intégrité des données échangées et peut donc être usurpé sans grande difficulté afin de transmettre des informations erronées pour une multitude de raisons [28]. En effet, la conception de l'AIS date de la fin des années 1990 et, par conséquent, il n'intègre pas les préoccupations en matière de sécurité de l'information actuelles. Une somme de contrôle est cependant calculée, pour chaque message transmis, afin de détecter les éventuelles

erreurs de transmission. Par ailleurs, d'autres manipulations frauduleuses peuvent être appliquées à l'AIS comme l'a montré Balduzzi et al. [10]. Nous présentons ci-dessous une liste des principales manipulations répertoriées sur l'AIS :

1. création de faux messages simulant la trajectoire de navires pour perturber la navigation maritime en faisant apparaître des alertes de collision (**bateau fantôme**) [10, 7] ;
2. falsifications des données de messages concernant la position [119, 145], la vitesse, le numéro MMSI [159], les données statiques [142] et les informations liées au trajet (fausses alarmes d'homme à la mer par exemple [10]) (**falsification de données**)
3. usurpation de l'identité d'un autre navire ou d'une autorité portuaire (**usurpation d'identité**) [3, 62] ;
4. arrêt volontairement de son AIS pour passer inaperçu (**arrêt volontaire AIS**) [60] ;
5. perturbation des communications (**brouillage**). En usurpant l'identité de l'autorité maritime, un utilisateur peut réserver la totalité des instants d'émission du mode d'accès TDMA (Time division multiple access) ce qui empêche les autres stations de communiquer (déni de service). Il peut aussi imposer à certaines stations un changement de fréquence de communication ou un délai avant émission d'un message [10].

Ces vulnérabilités de l'AIS ne doivent pas être négligées, car elles peuvent, en plus de perturber le trafic maritime, masquer des activités illicites, telles que la piraterie, la pêche illégale ou encore des attaques terroristes. Compte tenu de la généralisation de l'usage de l'AIS dans le monde maritime et du développement à venir des navires autonomes, l'amélioration de la cybersécurité de ce système est devenue une préoccupation majeure. C'est la raison pour laquelle l'objectif principal de cette thèse est la conception d'un récepteur AIS intelligent permettant de détecter les manipulations listées plus haut afin d'augmenter la fiabilité des données échangées. Par ailleurs, l'objectif visé est d'implémenter une solution de détection de ces manipulations en temps réel afin de fournir au commandant d'un navire ou à une station de contrôle à terre les informations nécessaires à l'évaluation de la réalité de la situation présente.

Conception d'un récepteur AIS

La conception du récepteur AIS intelligent, durant cette thèse, est soumise à un certain nombre de contraintes listées ci-dessous :

1. la durée de la conception est celle de la thèse (trois ans), ce qui est relativement court. Ce temps inclut à la fois le temps de développement des stratégies

de détection des falsifications, le temps d'implémentation du récepteur sur système embarqué et le temps de vérification de son comportement pour valider son fonctionnement ;

2. la conception nécessite un profil avec des compétences en traitement du signal et une expertise en méthodes et outils de conception de circuits et systèmes ;
3. Ce projet dispose de ressources matérielles et humaines limitées. Un doctorant y travaille à temps plein, assisté occasionnellement de quatre encadrants et d'un ingénieur de recherche pour des conseils et un soutien ponctuel.

Le respect de ces contraintes ajoute une difficulté supplémentaire à la conception du récepteur AIS intelligent qui déjà représentait un défi technique à plusieurs titres [103]. En effet, la conception d'un tel système nécessite une bonne expertise en développement de systèmes embarqués. Cette expertise implique la maîtrise de la conception matérielle et logicielle, des protocoles de communication, des spécifications des périphériques, des langages de programmation, des capteurs et des actionneurs, une bonne compréhension des circuits électroniques, de l'architecture des microcontrôleurs, des tests et du débogage. De surcroît, l'intégration d'algorithmes sophistiqués de traitement du signal sur le système embarqué pour identifier les messages falsifiés va augmenter la complexité de l'entreprise. La stratégie suivie est d'utiliser des outils d'aide à la conception pour appréhender rapidement et facilement tous ces domaines d'expertise.

Par ailleurs, le récepteur AIS intelligent a la particularité d'être un système embarqué en interaction étroite avec son environnement, notamment au travers des stratégies qui, à partir de signaux AIS reçus par le récepteur, détecteront, en temps réel, les falsifications d'informations. Ces interactions doivent être considérées précisément, en particulier durant l'étape de vérification, pour assurer le bon fonctionnement du système conçu dans son environnement réel. Pour cela, durant cette étape, le système embarqué et l'environnement physique sont considérés conjointement, l'ensemble formant ce qu'on appelle un *système cyber-physique* (CPS (Cyber-physical system)) [82]. Dans ce CPS, l'environnement physique est simulé pour simplifier la prise en considération de ses interactions avec le récepteur que nous concevons et faciliter la vérification des performances.

Cependant, modéliser et simuler un CPS est une tâche difficile à réaliser. En effet, ces systèmes sont très hétérogènes puisqu'ils contiennent à la fois des composants matériels, logiciels, numériques, analogiques qui se rapportent à une multitude de disciplines (mécanique, électronique, hydraulique, etc) [115, 121, 40]. Cette hétérogénéité se répercute tout le long du flot de conception, et implique l'utilisation de plusieurs outils, modèles et simulateurs différents [40]. Par ailleurs, en plus d'être variés, les composants interagissent entre eux de manière complexe et parallèle, et peuvent être en grand nombre, comme c'est le cas de l'IOT (Internet of things) [31, 146]. Cette hétérogénéité des composants et complexité des interactions doivent être modélisées précisément afin de reproduire fidèlement, lors de l'étape de vérification, le comportement de l'environnement physique du ou des

systèmes embarqués du CPS. En effet, pour un bon nombre d'applications, la fiabilité et la prédictibilité temporelle du comportement des systèmes embarqués sont soumises à des contraintes très strictes [173, 113] : une erreur de fonctionnement pour un système utilisé dans le domaine médical ou militaire peut avoir des conséquences tragiques [122].

Malheureusement, aujourd'hui, il n'existe pas de méthodes générales permettant de répondre à ces exigences de conception, c'est-à-dire proposant un modèle avec une sémantique temporelle adéquate pour modéliser précisément le CPS tout en permettant de vérifier rapidement le comportement du ou des systèmes embarqués du CPS. Ainsi, de nombreux systèmes ne seront pas concevables sans des changements substantiels dans les méthodes de conception actuelles [82]. Il est donc nécessaire de réexaminer les processus de conception afin de répondre aux problématiques concernant la vérification et la modélisation de CPS.

Collaboration avec l'ENSTA Bretagne et l'Ecole Navale

Pour mener à bien la conception du récepteur AIS intelligent et répondre à toutes les problématiques que nous venons de soulever, j'ai pu collaborer, durant cette thèse, avec l'Institut de Recherche de l'Ecole Navale (IRENav, EA 3634) et le Laboratoire des Sciences et Techniques de l'Information, de la Communication et de la Connaissance (Lab-STICC - UMR6285) dont l'ENSTA Bretagne est un des membres fondateurs. La collaboration entre l'IRENav et le Lab-STICC se fait en synergie avec la chaire de Cyberdéfense des systèmes navals (Ecole Navale, IMT-A, ENSTA Bretagne, NAVAL GROUP, THALES). Par ailleurs, le pôle d'excellence Cyber de la région Bretagne a financé les travaux de cette thèse. Dans le cadre de cette collaboration l'IRENav apporte ses compétences en traitement du signal dans le domaine maritime (équipe MoTIM) et le Lab-STICC apporte son savoir-faire en matière de méthodes et outils de conception de circuits (équipe ARCAD), mais également en matière de radio logicielle embarquée. Ce rapprochement des deux entités est une opportunité à plusieurs titres :

1. Exploration d'un domaine clé de la sécurité maritime : à savoir le système AIS avec la détection des falsifications constatées lors de l'utilisation de ce système.
2. Exploration des méthodes de conception de tels systèmes embarqués, dédiés au traitement du signal.

Au travers de ces travaux de thèse, nous ambitionnons de faire avancer, conjointement, l'état de l'art dans ces deux domaines, et d'aboutir à la conception d'un récepteur AIS intelligent détectant les falsifications de messages, comme présenté sur la Figure 3. L'approche duale adoptée entre, d'une part, le développement de nouvelles stratégies de détection de falsifications et, d'autre part, la conception du récepteur constitue l'originalité et la difficulté de ces travaux.

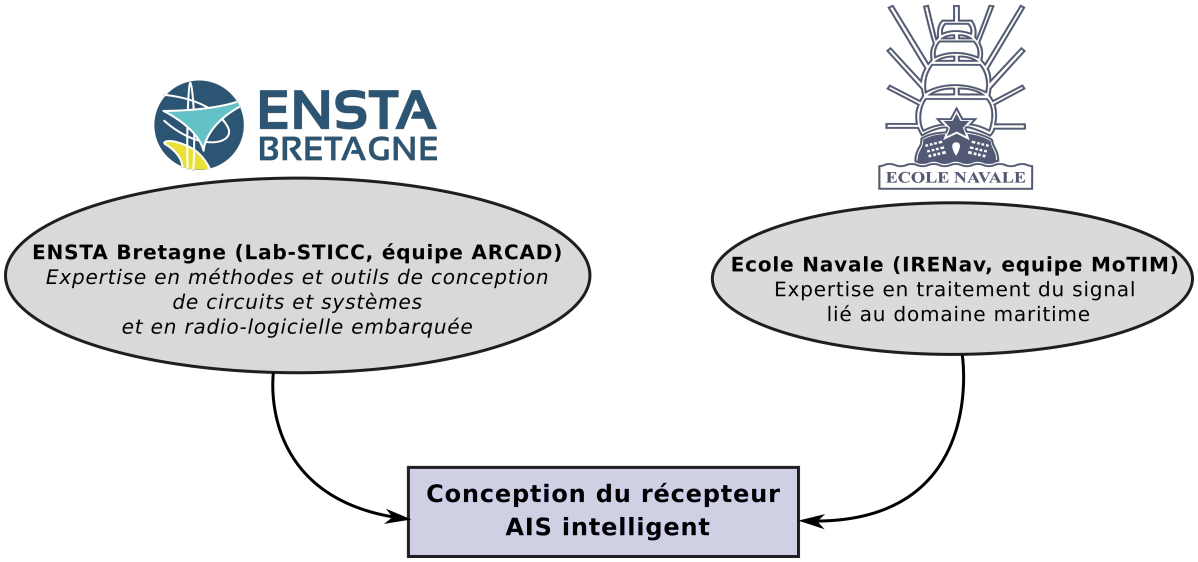

FIGURE 3 – Schéma présentant la collaboration entre l'Ecole Navale et l'ENSTA Bretagne pour concevoir le récepteur AIS intelligent

État de l'art

Détection des informations erronées dans les messages AIS

Malgré la prise de conscience récente (2010) de la vulnérabilité du système AIS, il n'y a pas eu de la part des organismes normalisateurs une refonte de la norme du système, et cela, pour plusieurs raisons liées aux limitations dans la conception même du système. Limitations au rang desquelles on peut citer un lien satellitaire unidirectionnel, les récepteurs ne peuvent pas recevoir les communications par satellites ou encore la saturation du système dans les zones à forte fréquentation. Par contre, depuis environ 2015 un nouveau système baptisé VDES (VHF Data Exchange System) est en cours de spécification. Il ne s'agit pas à proprement parler d'une évolution de l'AIS, mais plutôt d'un système de communication englobant plusieurs sous-systèmes dans la bande VHF marine, dont le système AIS. La mise en place de ce nouveau système va vraisemblablement prendre une décennie, voire davantage, aussi l'objet de ce travail est de proposer une solution relativement peu coûteuse et facile à mettre en œuvre qui permette un gain immédiat en terme de sûreté de fonctionnement sans avoir à modifier de manière substantielle les récepteurs existants.

Quelques stratégies et algorithmes, constituant des alternatives à notre approche, ont été proposés dans la littérature afin de combler certaines vulnérabilités, mais beaucoup de vulnérabilités restent toujours sans solutions. Par ailleurs, ces solutions restent purement théoriques et aucune d'entre elles n'a été réellement implémentée sur un système embarqué pour concevoir un AIS sécurisé. Ainsi, la nécessité de sécurisation de ce système pour assurer la fiabilité des informations échangées durant les communications reste d'actualité.

Néanmoins, de nombreux travaux ont été effectués sur des systèmes analogues dont la sûreté de fonctionnement présentait des lacunes. Les solutions techniques retenues sont,

d'une manière générale, plus complètes et abouties que pour l'AIS. C'est en particulier le cas des solutions proposées pour contrer les attaques du système GNSS (Global navigation satellite systems). Nous nous inspirons des solutions proposées pour ce système et de la manière dont elles sont classées [120, 64] afin de classer et d'organiser celles concernant l'AIS.

Stratégies basées sur des algorithmes complexes appliqués aux signaux reçus pour des récepteurs disposant d'une seule antenne : Ces algorithmes contrôlent l'évolution de la qualité des signaux reçus, en observant, par exemple, le SNR (Signal-to-noise ratio) ou l'évolution de la puissance du signal. L'apparition de distorsions ou de perturbations sur les signaux observés peut être due à des manipulations frauduleuses de l'AIS. Ces solutions ont été appliquées seulement deux fois à l'AIS non pas pour détecter des usurpations, mais pour détecter des arrêts injustifiés de transpondeurs. Par exemple, Guerriero et al. ont exploité les temps d'arrivée des messages pour détecter les arrêts d'émission trop longs et donc suspects [52], alors que Mazzarella et al. ont utilisé la loi de Friis en plus d'un réseau de neurones entraîné pour évaluer la probabilité qu'un arrêt d'émission de messages par un transpondeur soit justifié [105]. Par ailleurs, parmi ce type de stratégie, on compte aussi des méthodes extrayant des signatures radiofréquences des signaux reçus pour identifier matériellement chaque transpondeur [23] ou caractériser chacun de leurs canaux de transmission [118]. Ces signatures peuvent être extraites par des réseaux de neurones, comme dans la méthode de Leonardio et al. [84], ou directement à partir du signal en calculant des caractéristiques particulières (temps de montée du signal, temps de descente...), comme dans la méthode développée par Ray et al. [123]. Ces deux méthodes sont les seules à concerner l'AIS, et encore la première méthode concerne en réalité l'ADS-B qui est l'équivalent de l'AIS pour la circulation aérienne.

Trouver une signature discriminante afin d'identifier les transpondeurs est une difficulté récurrente. Dans les résultats reportés dans les références [84] et [123] les signatures ne sont pas assez discriminantes, en plus d'être très bruitées pour la méthode proposée dans [123], car calculées durant le régime transitoire des signaux. Ainsi, la stratégie que nous mettrons en place utilisera une signature plus robuste, à savoir l'offset en fréquence de la porteuse du signal appelé, dans la littérature, CFO (Carrier frequency offset). Cette signature s'est révélée discriminante dans beaucoup d'applications [136], et en particulier pour identifier les transpondeurs AIS.

Stratégies basées sur l'application de méthodes de chiffrement : De telles stratégies ont été proposées pour chiffrer les messages AIS. Par exemple, les solutions proposées dans les références [74, 48, 132, 9] sont des méthodes de cryptographie utilisant une clé publique. Ces solutions sont rétrocompatibles, c'est-à-dire qu'elles permettent l'interopérabilité avec les dispositifs AIS utilisés actuellement sans qu'il y ait besoin de modifier

le matériel ou le logiciel. Cependant, ce type de stratégie, pour être efficace, doit être appliquée par tous les utilisateurs d'AIS ce qui nécessite une mise à jour du logiciel et quelques changements dans le service AIS existant. Cette dernière condition limite son application et c'est pourquoi nous n'implémenterons pas ce type de solution.

Stratégies basées sur le contrôle des données reçues : Ce contrôle vise à détecter des incohérences dans les données reçues. Il peut concerner soit les données statiques comme dans la méthode de Campbell et al. [27] ou bien les données dynamiques comme dans la majorité des cas d'application. La cohérence des données dynamiques peut être vérifiée par les données provenant d'autres navires comme dans les études reportées dans les références [125, 137, 126]. L'étude des trajectoires des autres navires permet de définir un comportement dit "*normal*" duquel les navires ne doivent pas trop s'écarter. On parle alors de détection d'anomalies comportementales. Ce type de détection est très étudié dans la littérature et fait souvent appel à des algorithmes de machine learning ; plusieurs revues exposent ces travaux [137, 126]. Cependant, un bateau peut très bien avoir un comportement anormal tout en transmettant des données non falsifiées ou non usurpées et inversement. Il n'y a pas de lien entre une anomalie comportementale d'un bateau et une falsification de donnée. De plus, ce type de méthode ne permet pas une application en temps réel sur un système embarqué, car la phase d'apprentissage est longue et coûteuse en calculs. Ainsi, nous n'appliquerons pas ce type de méthode. D'autres solutions contrôlent la cohérence des données dynamiques d'un bateau à partir de l'ensemble des données qu'il a déjà transmises. Classiquement, ces méthodes cherchent à détecter des incohérences dans l'évolution de la position, comme un saut de position significatif que la vitesse du bateau ne peut justifier [124, 65, 81, 41, 138, 104, 57, 99, 135]. Des incohérences dans l'évolution de la vitesse peuvent aussi être vérifiées comme le propose Kontopoulos et al. [76]. Dans notre travail, nous contrôlerons l'évolution de la position et de la vitesse transmises pour chaque bateau afin de détecter des évolutions incohérentes. Néanmoins, nous utiliserons d'autres algorithmes que ceux proposés pour permettre une application en temps réel sur système embarqué. Pour finir, une toute autre stratégie, dont l'idée est présentée par Strohmeier et al. dans [143], propose de vérifier le respect de la norme technique par les messages [143]. L'idée est proposée pour être appliquée à l'ADS-B mais nous allons la reprendre pour l'appliquer à l'AIS et vérifier le respect du mode d'accès TDMA par chaque bateau, ce qui constituera à notre connaissance une première pour l'AIS.

Stratégies basées sur le calcul de la direction ou position des transpondeurs par l'utilisation de plusieurs capteurs. Dans cette catégorie, on trouve les méthodes de crowd-sourcing qui exploitent les mesures de plusieurs capteurs pour estimer la position, la vitesse ou la route des navires et ainsi en contrôler la cohérence avec les données transmises dans les messages [143]. Par exemple, la position peut être estimée en utilisant la différence

de temps d'arrivée des messages AIS sur un réseau de transpondeurs synchronisés AIS comme dans la méthode de Papi et al. [117]. De même, les signaux radars [72, 170, 158] ou radars imageurs (SAR (Synthetic-aperture radar)) [153, 53, 50] peuvent aussi offrir de nouvelles mesures de position, de direction et de vitesse du bateau. Enfin, l'exploitation de l'effet Doppler permet également, dans certaines configurations, de localiser un bateau [55]. L'une des principales limites de ce type de méthodes est qu'elles exploitent, outre l'AIS, d'autres capteurs, parfois placés à des endroits très éloignés les uns des autres [117]. De plus, l'estimation des positions, à cause de la complexité de mise en œuvre, peut être imprécise (quelques centaines de mètres pour [117] et plus de 10 km pour la [55]) ou peut prendre beaucoup de temps (quelques heures à une journée pour [153]) lorsqu'on utilise des données SAR par exemple, ce qui exclut de fait les applications en temps réel. C'est pourquoi nous n'exploiterons pas ce type de stratégie.

Stratégies basées sur l'application conjointe de plusieurs stratégies. Toutes les stratégies proposées présentent des faiblesses qui pourraient être exploitées par une manipulation sophistiquée. Cependant, dans certains cas, la force d'une stratégie peut compenser la faiblesse d'une autre. Ainsi, l'utilisation simultanée de plusieurs stratégies complémentaires peut constituer un moyen très puissant de détecter un nombre important et varié de manipulations comme rapporté par Psiaki et Humphreys [120]. C'est justement ce que nous faisons dans cette étude : trois stratégies sont appliquées conjointement.

Méthodes de conception d'un système embarqué

Le développement d'un système embarqué, comme notre récepteur AIS intelligent, passe forcément par l'application de trois étapes : la modélisation, la conception et la vérification [83]. Traditionnellement, ces trois étapes sont appliquées successivement : on parle de méthode de conception en cascade [128]. Ce type de méthode est représenté sur la partie gauche de la Figure 4 où les blocs *Exigences* et *Analyse* correspondent à l'étape de modélisation et les blocs *Conception* et *Mise en oeuvre* correspondent à l'étape de conception. La partie conception partitionne le système et sépare les fonctionnalités entre celles qui seront exécutées par logiciel et celles qui seront exécutées en matériel sur un circuit logique programmable. Ces fonctionnalités peuvent être développées par différentes équipes et les composants associés peuvent être ensuite assemblés pour former le système complet comme le préconise la méthode de conception *down-top*. L'inconvénient de ce type de méthode est que l'étape de vérification du système est réalisée tard dans le développement, après la composition du système complet. Or, lorsque la vérification montre que les performances attendues ne sont pas satisfaites, la modification et la correction du système conçu est rendue difficile à cause de l'état avancé du développement. Cela peut même se traduire par la nécessité de redévelopper certains composants du système, ce qui

constitue une perte importante de temps et d'argent [101].

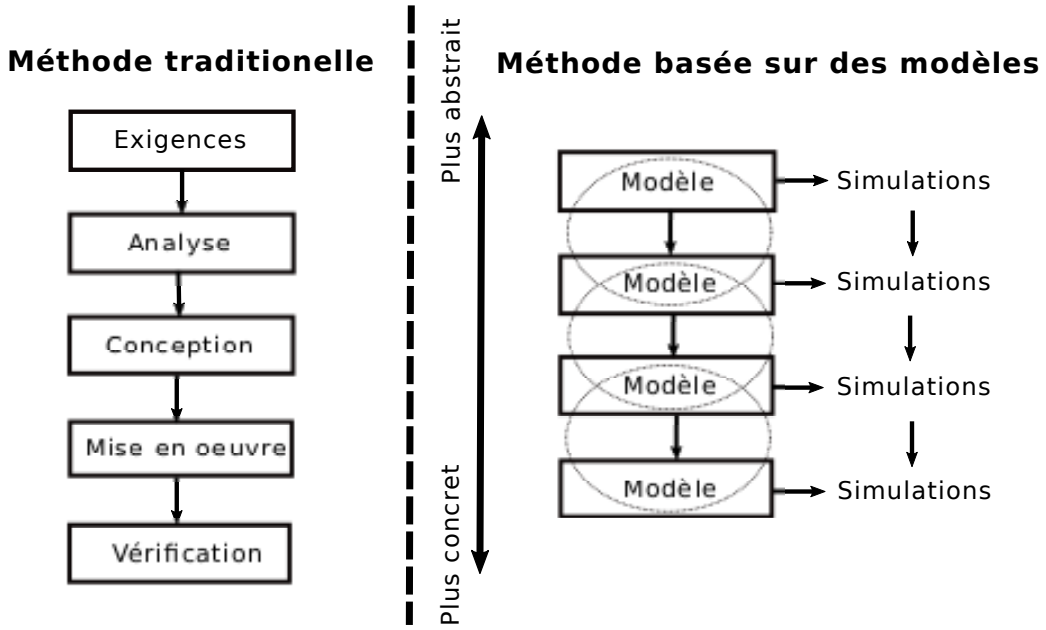

FIGURE 4 – Comparaison entre la méthode traditionnelle de conception et la méthode de conception basée sur des modèles [101]

Afin de résoudre ces difficultés, une nouvelle méthode de conception *top-down* a été proposée par l'ESL (Electronic system level) [68]. Cette méthode, présentée sur la partie droite de la Figure 4, part d'une modélisation comportementale du système avec un haut niveau d'abstraction et descend ensuite, par affinements successifs, à la description matérielle. À chaque niveau d'abstraction, une vérification est effectuée, afin de détecter précocement toute erreur dans la conception, et la corriger rapidement. Cette méthode considère uniquement des modèles du système à différents niveaux d'abstraction et les transformations réalisées entre ces modèles. C'est la raison pour laquelle on parle de conception basée sur des modèles (*model based design* (MBD (Model-based design)) en anglais) [114]. Chaque étape successive de modélisation rapproche la conception d'une mise en œuvre sur un système embarqué. Cela permet au développeur de se concentrer, durant les premières phases de développement, davantage sur les aspects généraux du système, plutôt que sur sa mise en œuvre. Ainsi, les étapes de modélisation, conception et vérification se chevauchent et sont appliquées plusieurs fois durant la conception, à mesure que le niveau d'abstraction diminue.

Afin d'accélérer la transition entre les différents niveaux d'abstraction, certains des outils développés par l'EDA (Electronic design automation) sont utilisés. L'EDA a vu le jour dans les années 1980 pour réduire la durée et la difficulté de la conception de SoC (System on chip) [130]. Cette industrie a développé des outils de conception automatisée de systèmes électroniques travaillant à différents niveaux d'abstraction [129] pour manipuler simultanément un grand nombre de transistors et augmenter la productivité des ingénieurs de conception [155]. Ces outils permettent, par exemple, à l'utilisateur de

passer automatiquement du niveau d'abstraction RTL (Register transfer level) au niveau d'abstraction des portes logiques, ou du niveau d'abstraction système au niveau d'abstraction RTL. Le passage du niveau d'abstraction système au niveau d'abstraction RTL a été rendu possible par l'utilisation d'outils de synthèse haut niveau (HLS (High level synthesis)). L'utilisation de ce type d'outils n'a connu de véritable essor que depuis les années 2010, car avant, la qualité des résultats obtenus n'était pas satisfaisante [102]. La modélisation au niveau système se fait avec un langage ayant un haut niveau d'abstraction (C++/C, Matlab, Python), ce qui simplifie grandement la prise en main par des ingénieurs système ou en traitement du signal. C'est la raison pour laquelle nous allons utiliser ce type d'outils pour la conception de notre récepteur.

Par ailleurs, du fait de la grande taille du système, une méthode de vérification statique n'est pas applicable, et c'est pourquoi une méthode de vérification dynamique est appliquée [171]. Parmi les méthodes de vérification dynamique envisageables, la co-simulation occupe une place importante [46]. Elle consiste en des techniques permettant la simulation globale d'un système via la composition de simulateurs simulant chacun une sous-partie du système. Les sous-parties peuvent être développées par différentes équipes utilisant leurs propres MOC (Model of computation) et outils de conception. Des prototypes physiques ou des solutions logicielles du système peuvent même être intégrés à la simulation à mesure que la conception du système progresse et s'affine. On parle de simulation Matérielle/Logicielle dans la boucle (respectivement HIL (Hardware-in-the-loop) et SIL (Software-in-the-loop) en anglais). Cependant, pour ce type de simulation, la composition de simulateurs reste difficile à mettre en place et sujette à de nombreuses erreurs de synchronisation. Ces erreurs peuvent entraîner des artefacts temporels tels que des retards aléatoires et de subtiles erreurs de causalité [29, 141]. Cela détériore la confiance globale dans la simulation et ne permet pas d'assurer une fiabilité et prédictibilité temporelle suffisante du comportement du CPS, comme cela est nécessaire [82].

Les mêmes inconvénients sont aussi rencontrés lorsque plusieurs MOC sont utilisés et composés pour modéliser le système complet et le simuler. C'est en particulier ce que proposent les outils tels que Ptolemy [25], Simulink, Simscape ou Modelica. Pour ce type de simulation, des mécanismes lourds de synchronisation entre simulateurs et MOC doivent être utilisés. Bien que des progrès importants aient été faits grâce à l'utilisation d'interfaces FMI (Functional mock-up interface) [47], ces mécanismes de synchronisation ralentissent néanmoins fortement la vitesse d'exécution de la simulation [45]. Ce ralentissement est acceptable lorsque le CPS simulé est de petite taille et peu complexe. C'est d'ailleurs la raison pour laquelle 60 % des ingénieurs utilisent Simulink pour simuler leurs CPS [171]. Mais pour simuler de plus gros CPS avec des composants ayant des interactions complexes et des contraintes d'exécution en temps réel à respecter [54] ce type d'outils de simulation n'offre pas de performances suffisantes [98, 42]. C'est pourquoi, compte tenu de la complexification croissante des CPS, il est nécessaire de développer d'autres méthodes de

vérification que la co-simulation. Ces méthodes doivent simuler le comportement du CPS sur une seule plate-forme et utiliser un seul MOC pour offrir une modélisation homogène du système, comme encouragé dans [45]. C'est justement une méthode remplissant ces critères que nous allons présenter et appliquer pour concevoir le récepteur AIS intelligent.

Pour résumer, compte tenu des différentes méthodes proposées dans la littérature scientifique et des contraintes qui pèsent sur la conception du récepteur comme rappelé plus haut, nous proposons une méthode de conception qui :

1. applique une méthode de conception *top-down* (MBD) qui part d'une modélisation comportementale du système avec un haut niveau d'abstraction pour descendre ensuite, par transformation successive, à l'implémentation matérielle ;
2. modélise le système avec un seul MOC et vérifie son comportement sur une seule plateforme de simulation ;
3. utilise un outil d'aide à la conception de type HLS pour générer automatiquement l'implémentation matérielle du système à partir de sa modélisation comportementale à un haut niveau d'abstraction.

Contributions

Renforcement de la sécurité de l'AIS

Pour palier aux vulnérabilités de l'AIS, nous avons développé plusieurs stratégies complémentaires, détectant les falsifications d'information contenues dans les messages. Ces stratégies considèrent à la fois les signaux bruts enregistrés par l'antenne reliée à l'AIS, mais aussi les données des messages. De plus, l'instant de réception des messages est considéré. Ces stratégies sont exécutées conjointement pour améliorer l'efficacité globale de détection des falsifications. Les stratégies implémentées sont :

1. Une stratégie détectant les falsifications de position et de vitesse en pistant les bateaux avec un filtre IMM (Interacting multiple model) pour vérifier la cohérence d'évolution des données dynamiques (position, vitesse) reçues (Stratégie 1) ;
2. Une stratégie détectant les faux messages et les bateaux fantômes en contrôlant le respect du mode d'accès TDMA par les bateaux lorsqu'ils transmettent leurs messages (Stratégie 2) ;
3. Une stratégie détectant les falsifications ou usurpations d'identité en identifiant les bateaux par la signature radiométrique de leur transpondeur (Stratégie 3).

Grâce à la juxtaposition de ces stratégies, toutes les manipulations, à part la falsification des données statiques, l'arrêt volontaire de l'AIS et le brouillage, sont détectées (voir Tableau 1). Seulement les messages de report de position émis par les transpondeurs de classe A sont considérés. Ces messages représentent 80 % des messages transmis [81] et

sont les seuls messages contenant l'information sur la position des bateaux. Il faut noter que nous avons découpé la catégorie falsification des données en trois sous-catégories : falsification de la position/vitesse, falsification des données statiques et usurpation d'identité, car les différentes stratégies sécurisent seulement certaines données.

Tableau 1 – Tableau présentant la réponse de la stratégie globale implémentée contre toutes les manipulations envisageables sur l'AIS.

	Stratégie 1	**Stratégie 2**	**Stratégie 3**
Bateau fantôme	✓	✓	✗
Falsification de la position/vitesse	✓	✗	✗
Falsification des données statiques	✗	✗	✗
Usurpation d'identité	✓	✓	✓
Arrêt volontaire AIS	✗	✗	✗
Brouillage	✗	✗	✗

Les stratégies 1 et 2 ont été présentées dans deux articles : un premier a été publié à la conférence du GRETSI en 2022 [97] et un autre dans le journal *Digital Signal Processing* [94]. Par ailleurs, la stratégie 3 a aussi été présentée dans un article qui a été soumis au journal *Expert Systems With Applications* en décembre 2022 et qui est en cours de révision.

Conception d'un récepteur AIS intelligent

Un récepteur AIS intelligent capable de détecter les falsifications de position, à partir de signaux AIS en bande de base, a été conçu sur FPGA (Field programmable gate arrays). Le FPGA a été implémenté par du code VHDL (Very high speed integrated circuit hardware description language) généré par application de la HLS (Vitis HLS [157]) au modèle comportemental du récepteur exprimé en C++ avec des variables à virgule fixe. Le langage C++ est un langage couramment utilisé et maîtrisé par des ingénieurs en traitement du signal ce qui leur assure une prise en main rapide de ce type d'outil. Cet outil a ainsi montré qu'il était, aujourd'hui, assez mature pour générer rapidement et efficacement la description matérielle d'un système complet, ce qui n'était pas évident, il y a encore quelques années. Le comportement du récepteur AIS intelligent conçu a été vérifié et validé avec de vrais signaux enregistrés dans la rade de Brest : les informations contenues dans ces signaux ont été correctement extraites. Par ailleurs, en observant le temps d'exécution, on a pu vérifier que l'exécution se faisait en temps réel.

Vérification sur FPGA du récepteur AIS intelligent plongé dans son environnement virtuel

Pour vérifier de manière plus exhaustive le comportement du récepteur AIS conçu, nous avons aussi présenté une nouvelle méthode de vérification de systèmes embarqués,

qui implémente un véritable laboratoire de simulation de CPS sur FPGA. Ce laboratoire simule en même temps et rapidement l'ensemble du CPS, et offre la possibilité de modifier l'environnement physique simulé afin de varier les scénarios testés. L'architecture matérielle du système embarqué est reproduite sur le FPGA et forme un véritable prototype, d'où le nom de laboratoire sur puce. Pour l'implémentation du FPGA, l'outil Vitis HLS est aussi utilisé. Cet outil synthétise automatiquement des bancs de test sur FPGA à partir d'une modélisation de l'ensemble du système sous forme d'un réseau d'acteurs, d'après le modèle de calcul KPN (Khan process networks) [83].

Cette méthode est appliquée pour vérifier le comportement de notre récepteur AIS intelligent, et le laboratoire de vérification sur FPGA permet :

1. de simuler plusieurs bateaux naviguant et transmettant des messages à notre récepteur dont l'amplitude de leurs signaux varie en fonction de l'éloignement au récepteur par application du modèle de propagation de Friis ;
2. de modifier le niveau de bruit des signaux transmis et les imperfections matérielles influençant la fréquence porteuse des signaux ;
3. de générer des scénarios de falsifications de messages, comme l'émission de fausses positions ou de fausses identités afin d'évaluer l'efficacité des stratégies de détection des falsifications de messages implémentées sur notre récepteur AIS intelligent.

Cette méthode a été publiée à la conférence Newcas en 2022 [96] et à la conférence du GRETSI en 2022 [95].

Publications

Revues Internationales

1. **M. Louart**, J.J. Szkolnik, A.O. Boudraa, J.C. Le Lann and F. Le Roy, "Detection of AIS messages falsifications and spoofing by checking messages compliance with TDMA protocol," *Digital Signal Processing*, vol. 136, pp. 1-16, 2023.
2. **M. Louart**, J.J. Szkolnik, A.O. Boudraa, J.C. Le Lann and F. Le Roy, "An approach to detect identity spoofing in AIS messages," *Expert Systems with Applications* (en révision).

Conférence Internationale

1. **M. Louart**, J.C. Le Lann, F. Le Roy, A.O. Boudraa and J.J. Szkolnik, "HLS-based accelerated simulation of large scale cyber-physical systems on FPGAs," *IEEE Interregional NEWCAS Conference*, pp. 332-336, 2022.

Conférences Nationales

1. **M. Louart**, J.J. Szkolnik, A.O. Boudraa, J.C. Le Lann et F. Le Roy, "Stratégie de détection des falsifications des positions des messages AIS basée sur l'application du filtre IMM," *Colloque GRETSI*, pp. 1-4, 2022 .

2. **M. Louart**, J.C. Le Lann, F. Le Roy, A.O. Boudraa et J.J. Szkolnik, "Émulation de systèmes cyber-physiques sur FPGA," *Colloque GRETSI*, pp. 1-4, 2022 .

Chapitre 1

Détection des falsifications de position et de vitesse

1.1 Introduction

Dans ce chapitre, nous développons une stratégie visant à détecter les falsifications de position et de vitesse introduites dans les messages AIS. La falsification de position est notamment observée lorsque l'utilisateur de l'AIS souhaite cacher sa position réelle aux autorités de contrôle pour masquer, par exemple, des activités illicites de pêche ou de contrebande. La falsification de vitesse seule est plus rarement constatée, elle résulte le plus souvent d'une falsification de position cohérente avec une fausse dynamique plus discrète qu'un brusque saut de position. Afin de détecter ces deux types de falsification, on applique un algorithme de pistage multi-cibles utilisant des filtres optimaux ou sous-optimaux, au sens de la minimisation de l'erreur quadratique [16]. Ainsi, chaque transpondeur AIS des bateaux est assimilé à une cible, et les données de position encapsulées dans les messages sont assimilées aux positions successives de la cible. L'algorithme de pistage prédit et estime (au sens de la théorie de l'estimation) la position des bateaux à partir de leurs positions passées et d'un modèle reproduisant leur dynamique, en plus des estimées des erreurs de prédiction et d'estimation. Toutes ces données sont utilisées par un test de conformité pour effectuer l'association cible/mesure et ainsi détecter les falsifications de position et de vitesse.

1.2 Algorithmes de pistage de cibles

Le pistage de cibles à l'aide de filtres optimaux date du milieu des années 50 [69] et fut d'abord utilisé pour la surveillance du trafic aérien à partir de signaux radars [15]. A cette époque, les mesures provenant du radar permettaient de repérer les avions par leur azimut et leur distance au radar. Or, lorsque le nombre d'avions détectés par le radar était important, il devenait difficile de faire l'association cibles(avions)/mesures, ce qui empêchait de connaître la vitesse de chaque avion comme expliqué dans [24]. Or, c'est grâce à la connaissance de la vitesse des avions que les militaires peuvent les intercepter et que les contrôleurs aériens peuvent prévenir d'éventuelles collisions. C'est pourquoi,

des algorithmes tirés de la théorie de la décision statistique [140] ont été développés pour pister chaque avion et trouver la correspondance cibles/mesures.

D'une manière analogue, la problématique de l'association cibles/mesures intervient également dans le contexte de la détection des falsifications de position ou de vitesse dans les messages AIS. En effet, une mesure (position du navire indiquée dans le message) trop différente de la position prédite par l'algorithme pistant le navire, conduit à considérer la mesure comme étant falsifiée : non conforme à la dynamique du navire.

1.2.1 État de l'art de l'utilisation des méthodes de pistage

Dans la littérature, de nombreux algorithmes, que nous avons cités dans l'Introduction, ont été développés pour pister des navires à partir des positions transmises dans les messages AIS [65, 81, 41, 138, 104, 57, 99, 135]. Une grande partie d'entre eux ne sont pas, en réalité, utilisés pour détecter les falsifications de position, mais, plutôt, pour aider à la navigation, en estimant les trajectoires les plus probables que suivront les bateaux. Parmi ces algorithmes, certains sont basés sur l'utilisation d'un filtre linéaire tel que le filtre de Kalman [65] ou le filtre IMM [138], et d'autres sur l'utilisation d'un filtre non linéaire comme le filtre de Kalman étendu [41, 66] et le filtre particulaire [104]. On peut citer également d'autres algorithmes qui ne mettent pas en œuvre de filtre, mais considère seulement les équations dynamiques du navire [81, 57]. De même, certains algorithmes estiment la dynamique des navires grâce à des méthodes d'apprentissage automatique basées sur des réseaux récurrents [99, 144, 66]. Enfin, il existe des algorithmes avec des méthodes exotiques, conçues spécialement pour utiliser les données provenant de messages AIS [135].

1.2.2 Intérêts du filtre IMM pour notre application

La prise en compte de ces différents algorithmes de pistage, et sachant que la dynamique des navires peut être modélisée par une équation linéaire, dont la statistique du bruit n'est pas toujours connue précisément et peut varier au cours du temps, nous incite à utiliser un filtre IMM. Ce filtre prend en compte plusieurs modes (ie. un nombre fini de bruits de dynamique). Il est une version sous-optimale de l'estimateur parfois désigné sous l'appellation de FHT (Full hypothsis tree) qui est basé sur la notion de modèles multiples définis à priori et qui couvre l'ensemble des modes du système [87]. Dans la pratique, le FHT n'est cependant pas utilisable en temps réel en raison d'une explosion combinatoire des modes au cours du temps.

Le modèle du filtre IMM implémenté dans notre cas comporte deux modes afin de rendre compte des dynamiques moyennes et extrêmes rencontrées dans le domaine maritime et apparaît particulièrement bien adapté à notre étude en raison des propriétés suivantes [13] :

1. Il offre un bon compromis entre le coût de calcul et la précision pour le pistage de cibles manœuvrantes ;
2. Il nécessite une capacité de calcul compatible avec une implémentation embarquée temps-réel ;
3. Il délivre une prédiction des erreurs de position utilisée par un test de validation pour accepter ou, au contraire, refuser la mesure de position reçue ;
4. Il permet la prise en compte des mesures non périodiques. En effet, certains messages AIS peuvent ne pas être reçus à cause de mauvaises conditions environnementales, ce qui induit une non-périodicité des mesures.

Avant de présenter le filtre IMM, nous introduisons, dans la prochaine partie, le filtre de Kalman, qui est un élément fondamental de son fonctionnement.

1.3 Pistage mono-cible, mono-modèle

L'approche la plus classique et la plus connue pour pister une cible est d'appliquer un filtre de Kalman. Ce type de pistage piste une seule cible à la fois et applique un seul modèle qui caractérise la dynamique de la cible. Les équations de ce filtre et le système de coordonnées utilisé pour exprimer les équations du mouvement de la cible sont présentées dans cette partie.

1.3.1 Système de coordonnées utilisé

Il existe plusieurs systèmes de coordonnées susceptibles d'être utilisés pour décrire le mouvement des cibles. Dans notre domaine d'application, le choix se fait, généralement, entre un système de coordonnées cartésiennes et un système de coordonnées sphériques. L'intérêt du système de coordonnées sphériques est que les mesures en provenance du GPS sont déjà exprimées dans ce système de coordonnées, il n'est donc pas nécessaire d'avoir à changer de système de coordonnées pour les exploiter. Cela évite l'application de matrices de changement de repère, permettant une économie substantielle en termes de coût calculatoire. Par contre, dans ce système de coordonnées, l'expression du mouvement de la cible est souvent complexe avec des équations non linéaires. Pour un système de coordonnées cartésiennes, c'est l'inverse, l'expression du mouvement se fait avec des équations linéaires simples mais les mesures GPS doivent être converties vers ce système de coordonnées sphériques [85]. Compte tenu du fait que dans notre cas d'application de nombreuses simplifications peuvent être faites sur l'expression de l'accélération dans des coordonnées sphériques, comme nous allons le montrer dans cette partie, nous décidons de choisir ce système de coordonnées. C'est la première fois dans la littérature que le pistage de bateau à partir de positions transmises dans les messages AIS s'effectue dans un sys-

tème de coordonnées sphériques ; habituellement, le système de coordonnées cartésiennes est utilisé.

Le système de coordonnées sphériques choisi est le système géodésique WGS84 (World geodetic system 1984) qui est représenté sur la Figure 1.1. Il est composé d'un ellipsoïde de révolution légèrement différent de l'IAG GRS 1980 et d'un géoïde (EGM96 (Earth gravitational models 1996)) pour modéliser, au mieux, la forme de la Terre. Un référentiel cartésien est associé à ce système, son centre est confondu avec le centre de l'ellipsoïde, son axe GX est orienté vers le méridien de Greenwich, son axe GZ est orienté vers le Nord géographique et son axe GY vers l'Est ; ce référentiel forme ainsi un trièdre direct. Du fait de la forme aplatie de l'ellipsoïde, deux rayons sont utilisés, un rayon équatorial (R_e) et un rayon polaire (R_p). Ce système géodésique est aussi utilisé par le GPS pour exprimer les positions des objets en fonction des coordonnées sphériques de latitude, longitude et altitude dans l'écrasante majorité des cas. Sur la Figure 1.1, λ représente la latitude du point M, ϕ sa longitude et r sa distance depuis le centre de la Terre G. La latitude λ varie dans l'intervalle $[-\frac{\pi}{2}; \frac{\pi}{2}]$ et la longitude ϕ varie dans l'intervalle $[0; 2\pi]$. Par ailleurs, le point H est le projeté du point M dans le plan $((\overrightarrow{GX}, \overrightarrow{GY})$.

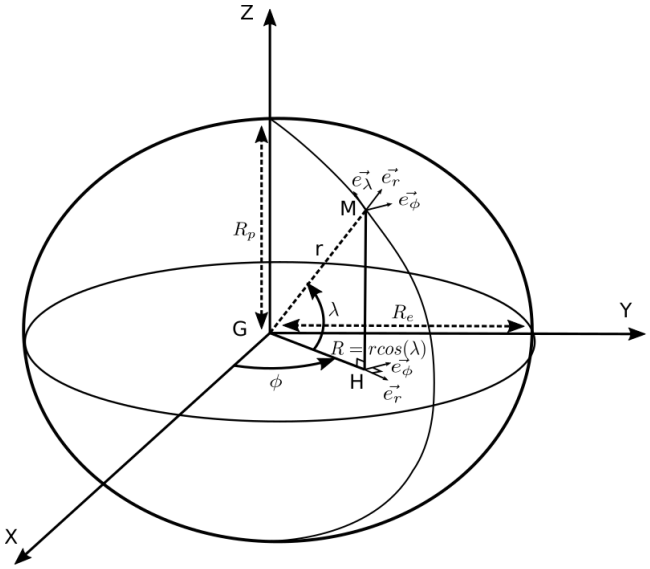

FIGURE 1.1 – Représentation du système WGS84

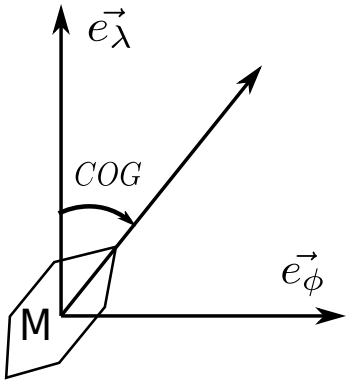

FIGURE 1.2 – Représentation de la route (COG) dans le plan (M,$\vec{e_\lambda}$,$\vec{e_\phi}$)

Comme nous l'avons dit, l'inconvénient majeur de l'utilisation d'un système de coordonnées sphériques réside dans le fait que, dans ce système de coordonnées, les équations de modélisation du mouvement des cibles sont complexes et non-linéaires. Cette complexité est causée par l'apparition de termes de couplage entre la longitude et la latitude lorsque la cible se déplace. Ces termes de couplage apparaissent en particulier dans l'expression de l'accélération de la cible [85]. Par exemple, sur les équations (1.1) et (1.2) de l'accélération suivant la latitude ($\vec{e_\lambda}$) et la longitude ($\vec{e_\phi}$) d'une cible ayant un mouvement sphérique, les termes de couplage sont : "$r\dot{\phi}^2 \sin(\lambda)\cos(\lambda)$" et "$-2r\sin(\lambda)\dot{\phi}\dot{\lambda}$". La

démonstration de ces équations est fournie en Annexe A.

$$\vec{e_\lambda} : r\ddot{\lambda} + r\dot{\phi}^2 \sin(\lambda)\cos(\lambda) = r\ddot{\lambda} + \tan(\lambda)\frac{v_\phi^2}{r} \tag{1.1}$$

$$\vec{e_\phi} : r\cos(\lambda)\ddot{\phi} - 2r\sin(\lambda)\dot{\phi}\dot{\lambda} = r\cos(\lambda)\ddot{\phi} - 2\tan(\lambda)\frac{v_\phi v_\lambda}{r} \tag{1.2}$$

avec v_λ la vitesse suivant le vecteur $\vec{e_\lambda}$ et v_ϕ la vitesse suivant le vecteur $\vec{e_\phi}$.

Les termes de couplage apparaissant sur l'accélération peuvent être majorées par :

$$a_{max} = \tan(\lambda)\frac{v_{max}^2}{R_{min}} \text{ : pour la latitude} \tag{1.3}$$

$$a_{max} = 2\tan(\lambda)\frac{v_{max}^2}{R_{min}} \text{ : pour la longitude} \tag{1.4}$$

avec v_{max} la vitesse maximale de la cible et R_{min} la rayon minimal représentant la distance entre le centre du système de coordonnées sphériques et la cible.

Dans notre cas d'application, la vitesse maximale d'un bateau pisté est de 45 kn (1 kn = 0,514 m s^{-1}) et le rayon minimal est le rayon de la Terre (r). Ainsi, en considérant la vitesse maximale des bateaux, et en sachant que le rayon de la Terre vaut 6371 km, pour $\lambda = 88°$, la valeur maximale du terme de couplage sur l'accélération est $a_{max} = 4{,}7 \times 10^{-3}$ kn s^{-1} sur la latitude et $a_{max} = 9{,}4 \times 10^{-3}$ kn s^{-1} sur la longitude. Ces deux valeurs sont négligeables.

Ainsi, ces deux termes de couplage dans l'expression de l'accélération sont négligeables sur presque n'importe quel point de la Terre à part aux pôles ($\lambda > 88°$). Les équations (1.1) et (1.2) de l'accélération suivant la latitude et la longitude peuvent alors être simplifiées et prendre l'expression des équations (1.5) et (1.6). Cette simplification est rarement appliquée, car souvent les mesures de position proviennent de capteurs de type radar, pour lesquels, le centre du système de coordonnées sphériques est le radar lui-même et non le centre la Terre. La distance R_{min} (distance entre le centre du système de coordonnées et la cible) est alors beaucoup plus faible, de l'ordre de grandeur de plusieurs dizaines de kilomètres, ce qui rend les termes de couplage sur l'accélération non négligeables [17].

$$\vec{e_\lambda} : r\ddot{\lambda} \tag{1.5}$$

$$\vec{e_\phi} : r\cos(\lambda)\ddot{\phi} \tag{1.6}$$

1.3.2 Dynamique de la cible

La dynamique des bateaux pistés suit deux modes différents :

— un mode pour lequel le bateau ne manœuvre pas - la vitesse et la route sont constantes (mode 1) ;

— un mode pour lequel le bateau manœuvre - la vitesse et/ou la route changent (mode 2).

Dans le mode 1 (nous parlons du mode 2 dans la partie (1.5.1)), la dynamique du bateau est modélisée par une accélération nulle sur chacune de ses coordonnées ($\ddot{\phi} = 0$ et $\ddot{\lambda} = 0$). Ce type de modélisation a déjà été proposé dans la littérature pour d'autres applications [17], et s'adapte bien à notre application car, comme présenté dans la partie précédente (1.3.1), les termes de couplage sur l'accélération peuvent être négligées. Ce modèle est un modèle classiquement utilisé pour suivre des cibles, il s'agit du modèle à vitesse constante (CV (Constant velocity))[33, 86]. Comme montré en annexe (B) son équation d'état à temps discret a pour expression :

$$X_{n+1} = F_n X_n + V_n \tag{1.7}$$

avec

— $F_n = \begin{pmatrix} 1 & \Delta T_n \\ 0 & 1 \end{pmatrix}$ la matrice de transition ;

— $X_n = \begin{pmatrix} \xi_n \\ \dot{\xi}_n \end{pmatrix}$ le vecteur d'état (ξ peut désigner λ ou ϕ) ;

— $\Delta T_n = t_{n+1} - t_n$ l'intervalle de temps entre deux messages ;

— V_n le bruit blanc supposé gaussien du modèle tel que $\mathbb{E}[V_n V_k^T] = Q_n \delta_{nk} = \begin{pmatrix} \frac{\Delta T_n^3}{3} & \frac{\Delta T_n^2}{2} \\ \frac{\Delta T_n^2}{2} & \Delta T_n \end{pmatrix} \tilde{q} \delta_{nk}$.

avec $\mathbb{E}[.]$ l'espérance mathématique, Q_n la matrice de covariance du bruit de modèle, $\tilde{q}$ la variance du bruit continu sur l'accélération définie en B.3, et δ_{nk} la fonction de Kronecker.

Cette équation décrit un système à temps discret, variant dans le temps, obtenue par discrétisation du système à temps continu. L'expression de Q_n est bien adaptée pour suivre des systèmes pour lesquels la période entre les mesures n'est pas constante [13], comme c'est le cas pour l'AIS.

Une équation d'état en temps discret similaire peut être obtenue en définissant directement le bruit en temps discret et non pas en discrétisant le bruit modélisé en temps continu. Dans ce cas, le bruit de modèle sur l'accélération est supposé être une suite de bruits blancs constants par morceau sur chaque période d'échantillonnage et indépendant entre les périodes. Pour ce type de modélisation, le bruit s'écrit :

$$V_n = \Gamma_n \mathcal{V}_n \tag{1.8}$$

avec : $\Gamma_n = \begin{pmatrix} \frac{\Delta T_n^2}{2} \\ \Delta T_n \end{pmatrix}$

On obtient alors :

$$Q_n = \Gamma_n \sigma_v^2 \Gamma_n' = \begin{pmatrix} \frac{\Delta T_n^4}{4} & \frac{\Delta T_n^3}{2} \\ \frac{\Delta T_n^3}{2} & \Delta T_n^2 \end{pmatrix} \sigma_v^2 \tag{1.9}$$

L'intérêt d'utiliser la matrice Γ_n est de faciliter le calcul de la matrice Q_n car σ_v est alors l'écart-type de l'erreur sur l'accélération tel que $0.5 \times a_M \leq \sigma_v \leq a_M$, avec a_M l'accélération maximale d'un bateau utilisant un AIS de classe A. Cependant, ce type de modélisation du bruit voit ses performances diminuer lorsque la période d'échantillonnage n'est pas constante. Nous présentons tout de même cette modélisation, car elle est utilisée, dans la partie 1.3.5, pour calculer l'indice de manœuvre du bateau.

1.3.3 Modélisation des observations

Les observations des cibles sont les positions transmises dans les messages AIS. Ces positions sont mesurées pour chaque transpondeur par un capteur GPS auquel il est branché. L'équation d'observation a la forme suivante :

$$Z_n = HX_n + w_n \tag{1.10}$$

Avec $H = \begin{pmatrix} 1 & 0 \end{pmatrix}$ la matrice d'observation et w_n le bruit de mesure qui est supposé être un bruit blanc gaussien d'écart-type $\sigma_w = 5$ m [71]. Ainsi la matrice de covariance du bruit de mesure a pour expression :

$$\mathbb{E}[w_n w_k] = R\delta_{nk} = \sigma_w^2 \delta_{nk} \tag{1.11}$$

Le bruit de mesure est indépendant du bruit de modèle :

$$\mathbb{E}[V_l w_k] = 0 \tag{1.12}$$

avec $(k, l) \in \mathbb{N}^2$ et V_k le vecteur du bruit du modèle à l'instant k.

1.3.4 Équations du filtre de Kalman

Le filtrage de Kalman discret est un algorithme permettant d'estimer l'état X_n d'un système linéaire à temps discret modélisé par les équations de dynamique et d'observation suivantes :

$$X_{n+1} = F_n X_n + V_n \tag{1.13}$$

$$Z_n = HX_n + w_n \tag{1.14}$$

avec

$$\begin{array}{llll} \mathbb{E}[w_k] = 0 & \mathbb{E}[V_k] = 0 & \mathbb{E}[w_k w_l^T] = R_k \delta_{kl} & \mathbb{E}[V_k V_l^T] = Q_k \delta_{kl} \\ \mathbb{E}[V_k w_l] = 0 & \mathbb{E}[X_0 w_k^T] = 0 & \mathbb{E}[X_0 V_k^T] = 0. & \end{array}$$

avec $(k, l) \in \mathbb{N}^2$

Cet estimateur fut développé en 1961 [70] par Rudolf Kalman et intégré par le programme Apollo aux ordinateurs des fusées [51]. Depuis, il est utilisé dans beaucoup d'applications différentes comme présenté dans [8]. Ce filtre calcule l'estimée $\widehat{X}_{n|n}$ du vecteur d'état X_n et la matrice de covariance de l'incertitude associée à cette estimée $\widehat{P}_{n|n}$, à partir de l'ensemble des mesures Z_n obtenues depuis l'instant initial, noté Z^n. De plus, ce calcul est exécuté de manière récursive, permettant ainsi de ne pas avoir à accumuler toutes les mesures Z^n. Le calcul consiste à estimer les deux premiers moments de la densité de probabilité, $p(X_n|Z^n)$ soit :

$$\widehat{X}_{n|n} = \mathbb{E}[X_n|Z^n] \tag{1.15}$$

$$\widehat{P}_{n|n} = \mathbb{E}[(X_n - \widehat{X}_{n|n})(X_n - \widehat{X}_{n|n})^T|Z^n] \tag{1.16}$$

Ce filtre est l'estimateur d'état à erreur quadratique moyenne minimale sous l'hypothèse gaussienne concernant l'incertitude sur le vecteur d'état à l'instant initial et tous les bruits entrant dans le système. Plusieurs voies sont possibles pour établir les équations du filtre ; on peut chercher :

— l'estimateur à variance minimale ;
— l'estimateur qui maximise la probabilité à posteriori de X_n sachant Z^n ;
— l'estimateur qui maximise la vraisemblance de X_n ;
— la solution linéaire récursive au problème des moindres carrés pondérés ;

Sous les hypothèses gaussiennes et pour un système linéaire, on peut montrer que toutes ces approches conduisent aux mêmes équations. Ce sont les équations du filtrage optimal de Kalman [70]. Ces équations se décomposent en deux étapes. Une étape de prédiction qui, à partir des estimées à l'instant précédent $n-1$ ($\widehat{X}_{n-1|n-1}$ et $\widehat{P}_{n-1|n-1}$), et de l'équation d'état du système (1.13), prédit le vecteur d'état ($\widehat{X}_{n|n-1}$) et sa matrice de covariance ($\widehat{P}_{n|n-1}$) à l'instant n. Cette étape est résumée par les équations (1.18) et (1.19). Une étape d'estimation qui, à partir des prédictions ($\widehat{X}_{n|n-1}$ et $\widehat{P}_{n|n-1}$) à l'instant n, et de l'équation d'observation du système (1.14), estime le vecteur d'état ($\widehat{X}_{n|n}$) et sa matrice de covariance ($\widehat{P}_{n|n}$) au même instant n. Cette étape est résumée par les équations (1.24) et (1.25).

Covariance du bruit de modèle :

$$Q_n = COV(V_n) = \mathbb{E}[V_n V_n^T] = \begin{pmatrix} \frac{\Delta T_n^3}{3} & \frac{\Delta T_n^2}{2} \\ \frac{\Delta T_n^2}{2} & \Delta T_n \end{pmatrix} \tilde{q}; \tag{1.17}$$

Vecteur d'état prédit :

$$\widehat{X}_{n|n-1} = F_n \widehat{X}_{n-1|n-1} \tag{1.18}$$

Covariance du vecteur d'état prédit :

$$\widehat{P}_{n|n-1} = COV(\widehat{X}_{n|n-1}) = F_n \widehat{P}_{n-1|n-1} F_n^T + Q_n \tag{1.19}$$

Covariance du bruit d'observation :

$$R_n = COV(Z_n) = COV(w_n) = \mathbb{E}[w_n w_n^T] = \sigma_w^2 \tag{1.20}$$

Innovation du filtre :

$$\tilde{Z}_n = Z_n - H\widehat{X}_{n|n-1} \tag{1.21}$$

Covariance de l'innovation :

$$S_n = COV(\tilde{Z}_n) = R_n + H\widehat{P}_{n|n-1}H^T \tag{1.22}$$

Gain du filtre :

$$K_n = \widehat{P}_{n|n-1}H^T[S_n]^{-1} \tag{1.23}$$

Vecteur d'état estimé :

$$\widehat{X}_{n|n} = \widehat{X}_{n|n-1} + K_n\tilde{Z}_n \tag{1.24}$$

Covariance du vecteur d'état estimé :

$$\widehat{P}_{n|n} = COV(\widehat{X}_{n|n}) = (I - K_nH)\widehat{P}_{n|n-1} \tag{1.25}$$

1.3.5 Indice de manœuvre de la cible

L'intérêt d'utiliser un filtre de Kalman à un seul modèle dynamique, par rapport à un filtre de pistage avec plusieurs modèles tel que le filtre IMM, dépend de l'indice de manœuvre de la cible. Cet indice, sans dimension physique, est fonction de l'incertitude sur le modèle dynamique de la cible, de l'incertitude sur les mesures et de l'intervalle de temps entre les mesures. Il est défini par l'équation suivante considérant le bruit de modèle selon la modélisation (1.9) :

$$\lambda_m = \frac{\sigma_v \Delta T^2}{\sigma_w} \tag{1.26}$$

Si cet indice est inférieur à 0,5, le filtre de Kalman offre les mêmes performances que le filtre IMM pour un coût de calcul inférieur [75]. Or, dans notre application, $\sigma_w = 5$ m, $\sigma_v = a_M = 1$ kn s^{-1} et $\Delta T = 10$ s (valeur maximale de la période d'émission des messages qui dépend de la dynamique des bateaux [133]). L'indice λ_m vaut donc dans ce cas 10,20 ce qui justifie l'utilisation d'un filtre IMM. En plus, il peut arriver que plusieurs messages d'affilée ne soient pas reçus. Dans ce cas, ΔT devient supérieur à 10 s et λ_m dépasse 10,20.

Cette analyse théorique a été confirmée par une analyse pratique : les performances de ces deux filtres ont été comparées dans cette étude [97]. Il s'est avéré qu'il était préférable d'utiliser un filtre IMM : l'erreur d'estimation des vecteurs d'état est inférieure, ce qui rend le filtre plus sensible aux falsifications de position. C'est dans cette optique que nous

présentons dans la partie suivante le pistage mono-cible multi-modèles et en particulier le filtre IMM.

1.4 Pistage de cibles manœuvrantes : pistage mono-cible multi-modèles

Il existe plusieurs approches envisageables pour pister une cible manœuvrante. Dans cette partie, les plus populaires de ces approches sont présentées et expliquées succinctement. Parmi ces approches, celles appliquant un pistage mono-cible multi-modèles est présentée plus en détails avec l'explication du principe de fonctionnement du filtre IMM.

1.4.1 Méthodes

La dynamique d'un bateau peut être caractérisée par deux modes. Un mode pour lequel le bateau ne manœuvre pas (mode 1) et un mode pour lequel il manœuvre (mode 2). Durant la manœuvre, la dynamique du bateau est caractérisée par une entrée inconnue, responsable de la manœuvre, qui varie dans le temps. Pour traiter le pistage de cibles manœuvrantes, les approches possibles les plus classiques considèrent soit :

— la commande inconnue, mais déterministe. On cherchera alors à estimer la commande. Une cible manœuvrante peut être modélisée dans ce cas par l'équation générale suivante :

$$X_{n+1} = F_n X_n + G_n u_n + V_n \tag{1.27}$$

où u_n est l'entrée (la commande) imposée lors de la manœuvre de la cible. Deux solutions sont alors possibles. Soit, nous estimons $\widehat{u}(n)$ et nous nous en servons pour corriger l'état estimé. Il s'agit de l'approche IE (Input estimation) [30]. Soit, lorsqu'une manœuvre est détectée (en considérant l'évolution de l'innovation normalisée) une composante accélération est rajoutée au vecteur d'état pour améliorer le pistage de la cible durant toute la manœuvre. Il s'agit du filtre à dimension variable (VSD) [11].

— la commande inconnue, mais aléatoire. Deux approches sont alors possibles :
  — soit on conserve l'équation d'état (1.13) et on ajuste le bruit de modèle au cours du temps, en fonction de l'innovation normalisée, pour permettre et améliorer le pistage lors des manœuvres.
  — soit, on suppose que la commande ne peut prendre qu'un nombre fini de valeurs correspondant au nombre de modes m_n que peut suivre la cible pistée. Chaque mode est caractérisé par une équation de dynamique particulière (mouvement à CV, à CA (Constant acceleration), à CT (Coordinated turn), etc) avec un certain niveau de bruit d'état. Cette approche est une technique d'estimation

hybride, car on estime à la fois l'état des cibles (variable continue) et la probabilité d'occurrence des modes m_n qui sont des variables discrètes. Parmi les méthodes de ce type, on citera :

— la méthode statique MM (multi-modèles) [152] qui ne prend pas en compte la possibilité de basculement possible d'un mode à l'autre ;

— La méthode optimale FHT qui elle prend en compte toutes les possibilités de basculement d'un mode à l'autre, mais qui, en conséquence, devient inexploitable en temps réel, car elle nécessite le stockage de tous les historiques possibles de modes depuis l'instant initial ;

— Les méthodes sous-optimales de type GPB (Generalized pseudo bayesian) [2] et IMM [20, 18].

Parmi toutes ces méthodes, l'IMM offre les meilleurs compromis entre le coût de calcul et la précision pour le pistage de cibles manœuvrantes [13]. Il est plus précis que le GPB d'ordre 1 et presque aussi précis que le GPB d'ordre 2 [19]. En outre, il a été démontré que l'IMM était capable de maintenir l'erreur d'estimation de la position à un niveau inférieur à l'erreur de mesure brute pendant la période critique de la manœuvre (début et fin) et de fournir une amélioration significative (réduction du bruit) aux autres instants [19].

1.4.2 Principe du filtre IMM

Dans le cas où la dynamique peut suivre plusieurs modes, l'estimée optimale $\widehat{X}_{n|n}$ de l'estimateur FHT et sa matrice de covariance associée $\widehat{P}_{n|n}$ peuvent être obtenues par l'équation suivante tirée de [12] :

$$\widehat{X}_{n|n} = \sum_{i=1}^{r^n} \widehat{X}_{i,n|n} P(H_i^n|Z^n) \tag{1.28}$$

$$\widehat{P}_{n|n} = \sum_{i=1}^{r^n} [\widehat{P}_{i,n|n} + (\widehat{X}_{n|n} - \widehat{X}_{i,n|n})(\widehat{X}_{n|n} - \widehat{X}_{i,n|n})^T] P(H_i^n|Z^n) \tag{1.29}$$

avec r le nombre de modèles, H_i^n l'historique possible de changement de mode depuis l'instant initial jusqu'à l'instant n, $\widehat{X}_{i,n|n}$ l'estimée optimale de l'historique H_i^n, Z^n l'ensemble des mesures obtenues depuis l'instant initial jusqu'à l'instant n et $P(H_i^n|Z^n)$ la probabilité que la cible ait suivi l'historique H_i^n sachant Z^n.

On se rend compte que cet estimateur optimal nécessite le stockage de tous les historiques possibles de modes avec leurs estimés optimales. Le nombre d'historiques possibles croît exponentiellement avec le temps. Par exemple, dans le cas où le nombre de modèles est fixe égal à n depuis l'instant initial jusqu'à l'instant k, le nombre total d'historiques possibles vaut :

$$N_k = n^k \tag{1.30}$$

C'est pourquoi l'estimateur optimal FHT est inutilisable. À la place, nous utiliserons

des algorithmes sous-optimaux qui limitent la croissance exponentielle dans le temps du nombre d'historiques possibles. L'idée consiste, soit, à fusionner (*merging*) les historiques de modes considérés comme similaires au sens d'un certain critère, ou bien à éliminer tous les historiques (*pruning*) dont la vraisemblance, par exemple, reste en dessous d'un certain seuil (paramètre de réglage).

Le filtre IMM, par exemple, ne considère que les modèles courants m_n^i possibles et non tous les historiques possibles pour construire un estimateur récursif. Il se compose, comme présenté sur la Figure 1.3, d'un filtre pour chaque modèle, d'un évaluateur de probabilité de modèle, d'un mélangeur d'estimations à l'entrée des filtres et d'un combinateur d'estimations à la sortie des filtres. Les multiples modèles interagissent par le biais du mélangeur pour suivre une cible qui manœuvre sur une trajectoire arbitraire. Les estimations d'état sont mélangées en fonction des probabilités de modèle et des probabilités de changement de modèle. Ce filtre a été développé à l'origine par H.A.P. Blom en 1984 [20], puis a été reprise par H.A.P. Blom et Bar Shalom en 1988.

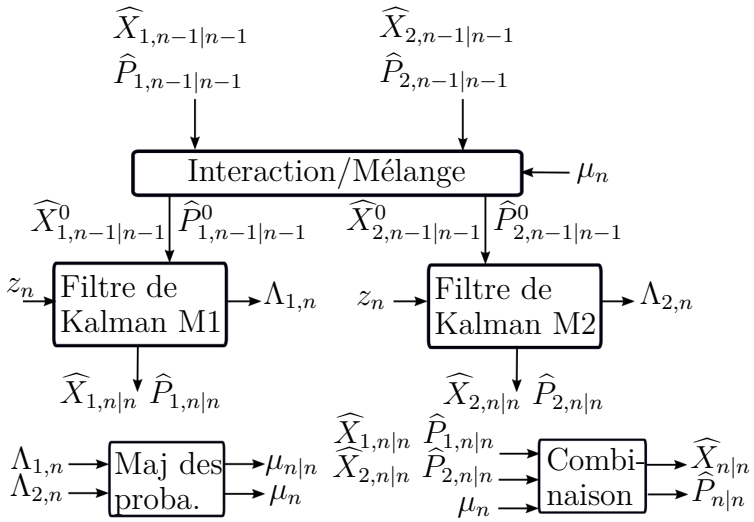

FIGURE 1.3 – Architecture du filtre IMM

1.5 Pistage de navires par filtrage IMM

Dans cette partie, nous présentons les équations du filtre IMM et la méthode à appliquer pour régler ses paramètres et adapter le filtre au pistage d'un navire utilisant un transpondeur AIS de classe A.

1.5.1 Modèles dynamiques

Le mouvement d'un bateau, comme le mouvement d'un avion dans un plan, peut être approximé par trois modèles [59]. Un modèle à vitesse constante (CV) appliqué lorsqu'il se

déplace selon une trajectoire rectiligne uniforme, un modèle à accélération constante (CA) lorsqu'il accélère ou décélère et un modèle à vitesse angulaire constante (CT) lorsqu'il effectue un virage. Dans le mode 1 (partie 1.3.2), comme nous avons déjà dit, le modèle CV permet d'approximer la dynamique du bateau. Dans le mode 2, lorsque le bateau manœuvre, les trois modèles CV, CA et CT sont applicables. Les modèles CT et CA utilisent plus de paramètres et nécessitent donc plus de calculs que le modèle CV. Par ailleurs, dans notre application, il se peut qu'il y ait un intervalle de temps important (> 10 s) entre la réception de deux messages consécutifs informant sur la position du navire. En effet, des déficiences techniques, la présence d'obstacles empêchant la communication et des mauvaises conditions environnementales peuvent provoquer la non-réception de messages. Pendant cet intervalle de temps, le bateau peut avoir changé plusieurs fois de comportement (changement de route, accélération ou décélération) sans qu'il y ait eu de messages transmis pour informer de cette manœuvre. Sans ces observations de position, les modèles CA et CT ne peuvent pas modéliser finement ces manœuvres, ce qui leur enlève leur gain de précision par rapport au modèle CV. Pour finir, le modèle CT est non linéaire et nécessite l'utilisation d'un EKF (Extended Kalman filter) pour gérer ces non-linéarités. Or, ce type de filtre est moins stable que le filtre de Kalman linéaire, et les variations de l'intervalle de temps entre les mesures peuvent affecter fortement ses performances. C'est pourquoi, seulement le modèle CV sera utilisé, comme dans [116], pour modéliser la dynamique du bateau pisté.

Ainsi, le filtre IMM que nous appliquons, modélise le mouvement des bateaux avec deux modèles dynamiques CV ayant chacun un bruit de modèle différent pour différencier le cas ou le bateau ne manœuvre pas, de celui où il manœuvre (virage, accélération, décélération). Nous fixons un rapport de 20 entre ces deux bruits de modèle comme dans [13]. Le bruit de modèle caractérisant le mode manœuvrant du bateau est déterminé en considérant que le changement de vitesse durant l'intervalle de temps ΔT est de l'ordre de grandeur de :

$$\sqrt{Q_{22}} \propto \sqrt{\tilde{q}\Delta T} \tag{1.31}$$

L'accélération maximale d'un bateau pisté (bateaux utilisant un AIS de classe A) est de $a_{max} = 1$ kn s^{-1}. De plus, dans des conditions normales d'utilisation de l'AIS l'intervalle de temps maximal entre deux messages est de $\Delta T = 10$ s. Ainsi, la variation de vitesse maximale est de $\Delta sog = 10$ kn. Ces valeurs nous permettent de fixer approximativement $\tilde{q}$ qui sera fixée plus précisément par des simulations de MC (Monte-Carlo).

$$\tilde{q} \propto \frac{(\Delta sog)^2}{\Delta T} \tag{1.32}$$

Pour le calcul de $\tilde{q}$, nous avons dû convertir les kn en m.s^{-1} et les m en degré, ce qui demande de considérer le rayon de la Terre et la latitude de l'endroit où l'on applique l'algorithme de pistage. Cette latitude est la latitude de l'École Navale et vaut 48,2827°.

C'est à cet endroit que nous avons effectué une partie de nos collectes de données pour les expérimentations. Par ailleurs, les simulations de MC, présentées dans la partie 1.8.1, ont montré que la valeur de $\tilde{q}$ optimale est $\tilde{q} = 2\frac{(\Delta sog)^2}{\Delta T}$, car, pour cette valeur, le RMSE (Root-mean-square error) (équation (1.56)) était minimal.

1.5.2 Équations

Comme expliqué, l'IMM applique deux filtres de Kalman exécutés en parallèle, qui ont les mêmes équations d'état et d'observation (équations présentées en 1.3.4), mais un bruit de modèle d'amplitude différente. Les estimées de ces deux filtres sont mélangées en entrée de l'IMM grâce à la matrice de transition de modes (Π) (aussi appelé matrice de transition de modèles), contenant les probabilités conditionnelles de basculer d'un mode à un autre. Ces probabilités sont calculées avec la méthode du temps de séjour [73]. Les résultats finaux ne sont pas très sensibles à ces valeurs.

$$\Pi = \begin{pmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{21} & \Pi_{22} \end{pmatrix} = \begin{pmatrix} 0.90 & 0.10 \\ 0.10 & 0.90 \end{pmatrix} \tag{1.33}$$

Le filtre IMM applique successivement les équations suivantes prises dans [166], [19] et [13]. Dans ces équations, $i, j \in \{1; 2\}$ et $r = 2$.

Interaction/mélange

Probabilité prédite que le mode M_j soit en vigueur à l'instant n [166] :

$$\mu_{j,n}^{-} = P(M_{j,n}|Z^{n-1}) = \sum_{i=1}^{r} \Pi_{ij}\mu_{i,n-1} \tag{1.34}$$

Probabilités de mélange : probabilité que la dynamique de la cible appartienne au mode M_i à l'instant $n-1$ sachant qu'elle appartient au mode M_j à l'instant n :

$$\mu_{i|j,n-1} = P(M_{i,n-1}|M_{j,n}, Z^{n-1}) = \frac{\Pi_{ij}\mu_{i,n-1}}{\mu_{j,n}^{-}} \tag{1.35}$$

Estimateur initial du mode M_j, $\widehat{X}^0_{j,n-1|n-1}$, obtenu après mélange des estimées de chaque mode à l'instant $n-1$:

$$\widehat{X}^0_{j,n-1|n-1} = \sum_{i=1}^{r} \mu_{i|j,n-1}\widehat{X}_{i,n-1|n-1} \tag{1.36}$$

Matrice de covariance associée à $\widehat{X}^0_{j,n-1|n-1}$.

$$\widehat{P}^0_{j,n-1|n-1} = \sum_{i=1}^{r} \mu_{i|j,n-1}(\widehat{P}_{i,n-1|n-1} + (\widehat{X}_{i,n-1|n-1} - \widehat{X}^0_{j,n-1|n-1})(\widehat{X}_{i,n-1|n-1} - \widehat{X}^0_{j,n-1|n-1})^T) \tag{1.37}$$

$\widehat{X}^0_{j,n-1|n-1}$ et $\widehat{P}^0_{j,n-1|n-1}$ sont les moments d'ordre 1 et 2 de la densité de probabilité $p(X_{n-1}|M_{j,n}, Z^{n-1})$ qui a pour équation d'après la loi des probabilités totales :

$$\begin{aligned} p(X_{n-1}|M_{j,n}, Z^{n-1}) &= \sum_{i=1}^{r} P(M_{i,n-1}|M_{j,n}, Z^{n-1}) p(X_{n-1}|M_{i,n-1}, Z^{n-1}) \\ &= \sum_{i=1}^{r} \mu_{i|j,n-1} p(X_{n-1}|M_{i,n-1}, Z^{n-1}) \end{aligned}$$

Application des r filtres de Kalman

les mêmes équations que celles de la partie 1.3.4 sont appliquées pour obtenir $\widehat{X}_{j,n|n-1}$, $\widehat{P}_{j,n|n}$, $S_{j,n}$, $\tilde{Y}_{j,n}$, $\widehat{X}_{j,n|n}$ et $\widehat{P}_{j,n|n}$ à partir des variables $\widehat{X}^0_{j,n-1|n-1}$ et $\widehat{P}^0_{j,n-1|n-1}$ sortant du mélangeur. Il faut tout de même noter que les équations (1.18) et (1.19) ne s'appliquent pas aux estimées $\widehat{X}_{n-1|n-1}$ et $\widehat{P}_{n-1|n-1}$ de l'instant $n-1$, mais aux estimées issues du mélange : $\widehat{X}^0_{j,n-1|n-1}$ et $\widehat{P}^0_{j,n-1|n-1}$.
$\widehat{X}_{j,n|n-1}$ et $\widehat{P}_{j,n|n-1}$ sont les moments d'ordre 1 et 2 de la densité de probabilité $p(X_n|M_{j,n}, Z^{n-1})$ et $\widehat{X}_{j,n|n}$ et $\widehat{P}_{j,n|n}$ sont les moments d'ordre 1 et 2 de la densité de probabilité $p(X_n|M_{j,n}, Z^n)$.

Mise à jour des probabilités des modes

Mise à jour de la probabilité que le mode M_j soit en vigueur à l'instant n :

$$\mu_{j,n} = P(M_{j,n}|Z^n) = \frac{\Lambda_{j,n}\bar{\mu_{j,n}}}{\sum_{i=1}^{r} \Lambda_{i,n}\bar{\mu_{j,n}}} \tag{1.38}$$

$\Lambda_{j,n}$ est la fonction de vraisemblance d'observer Z_n si la cible suit le mode j :

$$\Lambda_{j,n} = p(Z_n|M_{j,n}, Z^{n-1}) = \frac{\exp(-l^2_{j,n}/2)}{\sqrt{(2\pi)S_{j,n}}} \tag{1.39}$$

avec :

- $l^2_{j,n} = \tilde{Z}^T_{j,n} S^{-1}_{j,n} \tilde{Z}_{j,n}$ l'innovation normalisée du mode M_j à l'instant n
- $\tilde{Z}_{j,n} = Z_n - H\widehat{X}_{j,n|n}$ l'innovation du mode M_j à l'instant n

et

Combinaison des vecteurs d'état et covariances estimés

Estimations globales du vecteur d'état et de sa matrice de covariance :

$$\widehat{X}_{n|n} = \sum_{i=1}^{r} \mu_{i,n}\widehat{X}_{i,n|n} \tag{1.40}$$

$$\widehat{P}_{n|n} = \sum_{i=1}^{r} \mu_{i,n}\left(\widehat{P}_{i,n|n} + (\widehat{X}_{n|n} - \widehat{X}_{i,n|n})(\widehat{X}_{n|n} - \widehat{X}_{i,n|n})^T\right) \tag{1.41}$$

$\widehat{X}_{n|n}$ et $\widehat{P}_{n|n}$ sont les moments d'ordre 1 et 2 de la densité de probabilité $p(X_n|Z^n)$ qui a pour équation d'après la loi des probabilités totales :

$$\begin{aligned} p(X_n|Z^n) &= \sum_{i=1}^{r} P(M_{i,n}|Z^n)p(X_n|M_{i,n}, Z^n) \\ &= \sum_{i=1}^{r} \mu_{i,n} p(X_n|M_{i,n}, Z^n) \end{aligned}$$

1.5.3 Initialisation

L'initialisation est effectuée en appliquant la méthode de différenciation à deux points [13] pour les deux modes :

$$\widehat{X}_{1|1} = \begin{pmatrix} Z_1 \\ \frac{Z_1 - Z_0}{\Delta T(1)} \end{pmatrix} ; \widehat{P}_{1|1} = \begin{pmatrix} R & \frac{R}{\Delta T_1} \\ \frac{R}{\Delta T_1} & \frac{2R}{\Delta T_1^2} \end{pmatrix} \tag{1.42}$$

$\mu_{1,1} = 0{,}8$ et $\mu_{2,1} = 0{,}2$, représentant respectivement la probabilité initiale que le bateau soit dans le mode 1 et dans le mode 2. Ces probabilités sont déterminées statistiquement sur 10 000 messages enregistrés.

1.6 Détection des falsifications de position

Dans cette partie, nous présentons l'algorithme de la stratégie qui détecte les falsifications de position transmise dans les messages AIS. Cet algorithme applique le filtre IMM, que nous avons présenté dans la partie précédente, pour pister les bateaux et vérifier leur position.

1.6.1 Test de conformité

Certaines des mesures reçues sont falsifiées ou de mauvaises qualités. Ces mesures ne doivent pas être prises en considération, car l'une des meilleures façons de ruiner les performances d'un filtre de Kalman est d'y introduire des mesures erronées [15]. Une technique de tri des mesures, appelée test de validation ou test de conformité, est donc appliquée. Ce test de conformité calcule le carré de l'innovation normalisée (l_n) avec l'équation (1.43) à partir de l'innovation $\tilde{Z}_n$ et de sa matrice de covariance S_n, et est ensuite comparée à un seuil γ, pour valider ou rejeter la mesure, comme présenté par l'équation (1.44). Une explication du test de conformité et des probabilités de fausse alarme et de non détection est présentée en Annexe C.

$$l_n = \tilde{Z_n}^T S_n^{-1} \tilde{Z_n} \tag{1.43}$$

$$T(Z_n) = \begin{cases} H_0 : & Z_n \text{ n'a pas été falsifiée} \\ 1 & \text{si } l_n \leq \gamma \ Z_n \text{ validée} \\ 0 & \text{sinon} \quad Z_n \text{ rejetée} \end{cases} \tag{1.44}$$

Pour calculer le carré de l'innovation normalisée, nous avons besoin de connaître la prédiction du vecteur d'état ($\widehat{X}_{n|n-1}$), sa matrice de covariance ($\widehat{P}_{n|n-1}$), l'innovation ($\tilde{Z}_n$) et sa matrice de covariance (S_n). Ces variables s'obtiennent par la combinaison des vecteurs d'état et matrices prédits sur chaque mode par le filtre IMM et par l'application des équations suivantes :

$$\widehat{X}_{n|n-1} = \sum_{i=1}^{r} \mu_{i,n}^{-} \widehat{X}_{i,n|n-1}; \tag{1.45}$$

$$\widehat{P}_{n|n-1} = \sum_{i=1}^{r} \mu_{i,n}^{-} \left(\widehat{P}_{i,n|n-1} + (\widehat{X}_{n|n-1} - \widehat{X}_{i,n|n-1})(\widehat{X}_{n|n-1} - \widehat{X}_{i,n|n-1})^T \right) \tag{1.46}$$

$$\tilde{Z}_n = Z_n - H\widehat{X}_{n|n-1}; \tag{1.47}$$

$$S_n = R + H\widehat{P}_{n|n-1}H^T \tag{1.48}$$

$\widehat{X}_{n|n-1}$ et $\widehat{P}_{n|n-1}$ sont les moments d'ordre 1 et 2 de la densité de probabilité $p(X_n|Z^{n-1})$ qui a pour expression d'après la formule des probabilités totales :

$$\begin{aligned} p(X_n|Z^{n-1}) &= \sum_{i=1}^{r} P(M_{i,n}|Z^{n-1}) p(X_n|M_{i,n}, Z^{n-1}) \\ &= \sum_{i=1}^{r} \mu_{i,n}^{-} p(X_n|M_{i,n}, Z^{n-1}) \end{aligned}$$

La densité de probabilité $p(X_n|Z^{n-1})$ est supposée suivre une distribution normale de moyenne $\widehat{X}_{n|n-1}$ et de variance $\widehat{P}_{n|n-1}$ alors qu'elle résulte du mélange entre deux densités de probabilité suivant une distribution gaussienne ($p(X_n|M_{1,n}, Z^{n-1})$ et $p(X_n|M_{2,n}, Z^{n-1})$). C'est une approximation couramment faite pour les filtres IMM dont la justification, vérifiée dans notre cas d'application, est donnée par Bar-Shalom dans [14].

$\tilde{Z}_n$ suit une distribution normale, car les bruits des modèles et de mesure sont supposés gaussiens. Ainsi, l_n suit une loi du khi-deux à 1 degré de liberté. Pour avoir une probabilité de fausse alarme tel que $\alpha = 0{,}001$, nous fixons le seuil $\gamma = 10{,}82$. H_0 acceptée signifie que la mesure testée provient du transpondeur suivi. H_0 rejetée signifie que l'innovation ne suit pas une distribution normale avec une moyenne nulle et une variance égale à S_n : la mesure testée ne provient pas du transpondeur suivi. L'explication peut être que le transpondeur testé a falsifié sa position, que le GPS a effectué une erreur de mesure, ou

autre chose. L'utilisation d'un tel test pour détecter les mesures erronées de position a déjà été effectuée et s'est avérée efficace dans [138].

1.6.2 Algorithme

Nous présentons ci-dessous l'Algorithme (1) qui est appliqué par la stratégie 1 pour détecter des falsifications de position sur la latitude ou la longitude. Cet algorithme est présenté en régime permanent à l'instant discret n. Il considère en entrée le temps d'arrivée du message (Toa_n) et la mesure de position sur la latitude et la longitude (Z_n). En sortie, il renvoie un message d'alerte indiquant si l'hypothèse H_0 a été acceptée ou non. Plusieurs acronymes sont utilisés dans la description de l'Algorithme faisant référence aux différentes étapes de l'IMM :

- **IMMm** représente l'étape d'interaction et de mélange du filtre IMM ;
- **Kp** représente l'étape de prédiction du filtre de Kalman ;
- **Ke** représente l'étape d'estimation du filtre de Kalman ;
- **IMMc** représente l'étape de combinaison du filtre de IMM. Nous appliquons cette étape à deux endroits, un premier pour combiner les variables prédites et un deuxième pour combiner les variables estimées ;
- **IMMmaj** représente l'étape de mise à jour des probabilités des modes.

Il faut noter, comme nous l'avons expliqué en (1.6.1), que si le test du khi-deux n'est pas vérifié, la mesure est rejetée. Dans ce cas, comme montré sur l'Algorithme (1), les étapes **Ke**, **IMMmaj** et **IMMc** ne sont pas appliquées. Les estimées globales de chaque mode prennent alors la valeur de leur prédiction. Il faut noter que lorsque l'hypothèse H_0 a été rejetée durant cinq mesures d'affilée, le filtre est recalé par application de l'équation (1.49) sur les positions transmises après ces cinq positions rejetées. En effet, il peut arriver qu'après une forte falsification de position (plusieurs centaines de mètre ou quelques kilomètre), le filtre perde la piste du bateau ayant falsifié ses positions, comme cela est présenté sur les Figures 1.17 et 1.18 de la partie 1.9.2, ce recalage permet de pister à nouveau les positions du bateau.

$$\widehat{X}_{n|n} = \begin{pmatrix} Z_n \\ \\ \frac{Z_n - Z_{n-1}}{\Delta T_n} \end{pmatrix} ; \widehat{P}_{n|n} = \begin{pmatrix} R & \frac{R}{\Delta T_n} \\ \\ \frac{R}{\Delta T_n} & \frac{2R}{\Delta T_n^2} \end{pmatrix} \tag{1.49}$$

1.7 Vérification de la cohérence de la vitesse avec l'évolution des positions

Sachant que la vitesse intervient directement dans le mode d'accès TDMA, qui sera considéré par la stratégie 2 présentée dans le prochain chapitre, il est nécessaire de vérifier

Algorithm 1 Détection des falsifications de position

Data: Toa_n, Z_n
Result: H_0
At each timestep n>2 :
$\Delta T_n = Toa_n - Toa_{(n-1)}$
for *lat. and lon.* **do**
$\widehat{X}^0_{1,n|n-1}, \widehat{P}^0_{1,n|n-1}, \mu^-_{1,n}, \widehat{X}^0_{2,n|n-1}, \widehat{P}^0_{2,n|n-1}, \mu^-_{2,n} = \mathbf{IMMm}(\widehat{X}_{1,n-1|n-1}, \widehat{P}_{1,n-1|n-1}, \mu_{1,n-1}, \widehat{X}_{2,n-1|n-1}, \widehat{P}_{2,n-1|n-1}, \mu_{2,n-1})$
$\widehat{X}_{1,n|n-1}, \widehat{P}_{1,n|n-1}, S_{1,n}, \tilde{Z}_{1,n} = \mathbf{Kp}(\widehat{X}^0_{1,n|n-1}, \widehat{P}^0_{1,n|n-1}, Z_n)$
$\widehat{X}_{2,n|n-1}, \widehat{P}_{2,n|n-1}, S_{2,n}, \tilde{Z}_{2,n} = \mathbf{Kp}(\widehat{X}^0_{2,n|n-1}, \widehat{P}^0_{2,n|n-1}, Z_n)$
$\widehat{X}_{n|n-1}, \widehat{P}_{n|n-1}, S_n, \tilde{Z}_n = \mathbf{IMMc}(\widehat{X}_{1,n|n-1}, \widehat{X}_{2,n|n-1}, \widehat{P}_{1,n|n-1}, \widehat{P}_{2,n|n-1}, \mu^-_{1,n}, \mu^-_{2,n}, Z_n)$
$l_n = \tilde{Z_n}^T S_n^{-1} \tilde{Z}_n$
if $l_n < \gamma$ **then**
$\widehat{X}_{1,n|n}, \widehat{P}_{1,n|n} = \mathbf{Ke}(\widehat{X}_{1,n|n-1}, \widehat{P}_{1,n|n-1}, \tilde{Z}_{1,n})$
$\widehat{X}_{2,n|n}, \widehat{P}_{2,n|n} = \mathbf{Ke}(\widehat{X}_{2,n|n-1}, \widehat{P}_{2,n|n-1}, \tilde{Z}_{2,n})$
$d^2_{1,n} = \tilde{Z}^T_{1,n} S^{-1}_{1,n} \tilde{Z}_{1,n}$
$d^2_{2,n} = \tilde{Z}^T_{2,n} S^{-1}_{2,n} \tilde{Z}_{2,n}$
$\mu_{1,n}, \mu_{2,n} = \mathbf{IMMmaj}(d_{1,n}, \mu^-_{1,n}, d_{2,n}, \mu^-_{2,n})$
$\widehat{X}_{n|n}, \widehat{P}_{n|n} = \mathbf{IMMc}(\widehat{X}_{1,n|n-1}, \widehat{P}_{1,n|n-1}, \mu_{1,n}, \widehat{X}_{2,n|n-1}, \widehat{P}_{2,n|n-1}, \mu_{2,n})$
$NbErr = 0$
$H_0 = 1$
else
$\widehat{X}_{1,n|n} = \widehat{X}_{1,n|n-1}; \widehat{P}_{1,n|n} = \widehat{P}_{1,n|n-1}$
$\widehat{X}_{2,n|n} = \widehat{X}_{2,n|n-1}; \widehat{P}_{2,n|n} = \widehat{P}_{2,n|n-1}$
$\mu_{1,n} = \mu^-_{1,n}$
$\mu_{2,n} = \mu^-_{2,n}$
$\widehat{X}_{n|n} = \widehat{X}_{n|n-1}$
$\widehat{P}_{n|n} = \widehat{P}_{n|n-1}$
$NbErr = NbErr + 1$
$H_0 = 0$
end
if $NbErr = 5$ **then**
$\widehat{X}_{1,n|n}, \widehat{X}_{2,n|n} = \left(Z_n; \frac{Z_n - Z_{n-1}}{\Delta T_n}\right)$
$\widehat{P}_{1,n|n}, \widehat{P}_{2,n|n} = \begin{pmatrix} R & \frac{R}{\Delta T_n} \\ \frac{R}{\Delta T_n} & \frac{2R}{\Delta T_n^2} \end{pmatrix}$
$NbErr = 0$
end
end

la fiabilité des vitesses transmises dans les messages. Vu que la fiabilité de la position est déjà contrôlée par le filtre IMM, présenté dans la partie précédente, il ne reste plus qu'à assurer la cohérence entre l'évolution de la position et la vitesse du bateau pour assurer la fiabilité de cette dernière. C'est pourquoi un algorithme vérifiant justement cette cohérence est présenté dans cette même partie.

1.7.1 Calcul de la vitesse

Pour assurer la cohérence entre la vitesse et l'évolution de la position, un test de conformité est appliqué entre la vitesse calculée, à partir des vitesses angulaires présentes dans les vecteurs d'état estimés sur la latitude et la longitude, et la vitesse transmise dans les messages. L'équation permettant de calculer la vitesse, appelée aussi SOG (Speed over ground), a pour expression :

$$sog_c(n) = \sqrt{\widehat{X}^2_{\lambda(n|n)}(2)R_N^2 + \widehat{X}^2_{\phi(n|n)}(2)R_E^2 cos^2(\widehat{X}_{\lambda(n|n)}(1))} \tag{1.50}$$

avec $\widehat{X}_{\lambda(n|n)}(1)$ et $\widehat{X}_{\lambda(n|n)}(2)$ la première et la deuxième coordonnée du vecteur d'état estimé sur la latitude. $\widehat{X}_{\phi(n|n)}(2)$ est la deuxième coordonnée du vecteur d'état estimé sur la longitude. La deuxième coordonnée correspond à la dérivée première par rapport au temps. R_e et R_p représentent respectivement les rayons équatorial et polaire, et apparaissent sur la Figure 1.1 présentant le système de coordonnées WGS84.

L'incertitude sur le calcul de $sog_c(n)$ est déterminée grâce aux matrices de covariance estimées sur la latitude et la longitude. En plus, nous utilisons la loi de la propagation de l'incertitude d'une combinaison non linéaire $Z = f(X, Y)$ [77]. Sachant que les variables $\widehat{X}^2_{\lambda(n|n)}(2)$ et $\widehat{X}^2_{\phi(n|n)}(2)$ sont indépendantes, car la pseudo accélération peut-être négligée comme nous l'avons montré en (1.3.1) et en considérant la variable $\widehat{X}_{\lambda(n|n)}(1)$ comme constante, l'incertitude sur la vitesse a pour expression :

$$\sigma_Z^2 = \left(\frac{\partial f(X,Y)}{\partial X}\right)^2 \sigma_X^2 + \left(\frac{\partial f(X,Y)}{\partial Y}\right)^2 \sigma_Y^2 \tag{1.51}$$

avec $Z = sog_c(n)$, $X = \widehat{X}_{\lambda(n|n)}(2)$, $Y = \widehat{X}_{\phi(n|n)}(2)$ et $Z = f(X, Y)$ qui est l'équation (1.50). En développant cette équation, nous obtenons :

$$\sigma^2_{sog_c}(n) = \left(\frac{R_N^2 \widehat{X}_{\lambda(n|n)}(2)}{sog_c}\right)^2 \sigma^2_{\widehat{X}_{\lambda(n|n)}(2)} + \left(\frac{R_E^2 cos^2(\widehat{X}_{\lambda(n|n)}(1))\widehat{X}_{\phi(n|n)}(2)}{sog_c}\right)^2 \sigma^2_{\widehat{X}_{\phi(n|n)}(2)} \tag{1.52}$$

Les écarts types $\sigma_{\widehat{X}_{\lambda(n|n)}(2)}$ et $\sigma_{\widehat{X}_{\phi(n|n)}(2)}$ s'obtiennent à partir des matrices de covariance estimées sur la latitude et la longitude $\widehat{P}_{\lambda(n|n)}$ et $\widehat{P}_{\phi(n|n)}$.

1.7.2 Vérification de la vitesse émise

Un test de conformité, similaire à (1.44), est appliqué à la vitesse transmise dans les messages. Ce test utilise le carré de l'innovation normalisée avec l'équation :

$$l_{sog}(n) = \tilde{Z}_{sog}(n)^T S_{sog}^{-1}(n) \tilde{Z}_{sog}(n) \tag{1.53}$$

L'innovation sur la vitesse est calculée par l'équation :

$$\tilde{Z}_{sog}(n) = Z_{sog}(n) - sog_c(n) \tag{1.54}$$

avec $sog_c(n)$ la vitesse calculée avec l'équation (1.50) et $Z_{sog}(n)$ la mesure de la vitesse envoyée par les messages AIS.

Sachant que les variables $Z_{sog}(n)$ et $sog_c(n)$ sont indépendantes, la variance de l'innovation a pour équation :

$$S_{sog}(n) = \sigma^2_{Z_{sog}} + \sigma^2_{sog_c}(n) \tag{1.55}$$

avec $\sigma_{Z_{sog}} = 0{,}3$ kn d'après [162].

Comme $|\tilde{Z}_{sog}(n)|$ ne suit pas une distribution normale, l_{sog} ne suit pas une loi du khi-deux. Nous fixons donc le seuil γ du test de conformité sur les mesures par simulation de Monte Carlo (partie 1.8.3) pour avoir une probabilité de fausse alarme $\alpha < 0{,}01$. Un seuil fixé à $\gamma = 9{,}0$ répond à cette contrainte.

1.8 Simulations de Monte Carlo

Des simulations de MC sont exécutées sur la position et la vitesse pour régler le bruit de modèle $\tilde{q}$ et le seuil du test de conformité sur les mesures de vitesse. En plus, ces simulations permettent d'évaluer la sensibilité des algorithmes détectant les falsifications de position et de vitesse. Pendant ce réglage, afin de tester et régler l'algorithme de pistage en fonction de ses conditions limites d'utilisation, nous simulons la trajectoire d'un bateau qui accélère au maximum de ce qui est envisageable pour un bateau utilisant un AIS de classe A.

1.8.1 Évaluation du suivi de position

La sensibilité de l'algorithme détectant les falsifications de position est évaluée à partir de simulations de MC comprenant 10 000 exécutions. Chaque exécution de MC applique un même scénario qui dure environ 440 s et contient 64 mesures de position. Un exemple d'une trajectoire produite par une exécution de MC est présentée sur les Figures 1.4, 1.5 et 1.6. Le bateau part d'une position initiale (48,2827,−4,4167) avec une vitesse initiale v = 2 kn. La position initiale correspond à la position de l'école Navale. Durant la première

phase (phase 1 (p1)) de la trajectoire, qui dure 200 s, le bateau suit une trajectoire à course constante et vitesse quasi-constante (l'accélération est un bruit blanc centré d'écart-type $\sigma \propto 0{,}02$ kn s^{-1}). Ensuite, le bateau accélère durant 40 s avec une accélération égale à son accélération maximale ($a_{max} = 1$ kn s^{-1}). Il s'agit de la phase 2 (p2) de la trajectoire. La vitesse du bateau passe de 2 kn à 42 kn ce qui se rapproche de la vitesse maximale d'un bateau. Puis, durant 40 s, le bateau reprend une trajectoire rectiligne à vitesse quasi-constante à 42 kn (l'accélération est un bruit blanc centré d'écart type $\sigma = \propto 0{,}02$ kn s^{-1})). Il s'agit de la phase 3 (p3) de la trajectoire. Enfin, durant 160 s, pour modéliser la non-réception de messages, comme cela arrive régulièrement dans la réalité [80], aléatoirement, un nombre de messages compris entre 0 et 8 ne sont pas reçus. Il s'agit de la phase 4 (p4) de la trajectoire. L'intervalle de temps, entre chaque mesure de position, est de $\Delta T_n = 10$ s $\pm$ 20 % pour les phases 1 et 2, et de $\Delta T_n = 2$ s $\pm$ 20 % pour les phases 3 et 4. Les valeurs de ΔT_n utilisées dépendent de la vitesse du bateau comme défini par la norme AIS [133].

Il faut noter que durant la phase 2, ΔT_n devrait normalement passer successivement à 6 s puis à 2 s à mesure que la vitesse augmente, mais, parce qu'en pratique il y a un léger retard avant que la période d'émission de l'AIS change et se règle sur la période spécifiée par la norme, nous avons fixé ΔT à 10 s durant toute cette phase. Par ailleurs, considérer $\Delta T = 10$ s durant la phase d'accélération, au lieu de $\Delta T = 6$ s, augmente l'incertitude sur le modèle et permet ainsi de tester le filtre dans ses conditions limites. Une simulation de MC pour une route de 0° (COG = 0°) et une autre pour une route de 45° (COG = 45°) sont effectuées. La Figure 1.2 rappelle ce que représente la route dans le plan (M,$\vec{e_\lambda}$,$\vec{e_\phi}$). Lorsque COG = 0°, le bateau ne se déplace que suivant la latitude, ce qui permet de se placer dans le cas où l'erreur sur le modèle, due à l'accélération du bateau, est maximale. Dans ce cas, l'indice de manœuvre vaut $\lambda_m = 10{,}20$. Lorsque $COG = 45°$, l'accélération se répartit entre la latitude et la longitude, l'indice de manœuvre vaut $\lambda_m = 7{,}2$. Chaque mesure de latitude et de longitude est affectée d'un bruit blanc gaussien d'écart-type $\sigma_v \propto 5$ m.

La Figure 1.7 présente l'évolution du RMSE de la position estimée (RMSE IMM) par rapport à la position vraie. Le RMSE est calculé par l'équation (1.56) présentée ci-dessous. Nous constatons que cette erreur sur la position est toujours inférieure à l'imprécision sur la mesure de position (RMSE mes), dont la valeur moyenne vaut $\sqrt{2} \times 5 = 7{,}07$ m, sauf aux instants où l'erreur sur le modèle est maximale (lorsque le bateau commence et finit de manœuvrer). Durant ces instants, le RMSE de la position estimée est légèrement supérieur à celui de la position mesurée. Ce résultat montre la réduction de bruit qu'offre l'IMM. Les Figures 1.8 et 1.9 présentent les probabilités de chaque mode sur la latitude et la longitude. Ces Figures montrent le changement de modèle qui s'opère automatiquement par le filtre IMM pour s'adapter au mieux aux changements de dynamique de la cible.

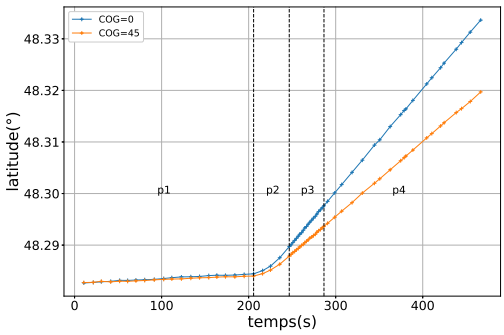

FIGURE 1.4 – Évolution de la latitude simulée au cours du temps pour COG = 0° et COG = 45°.

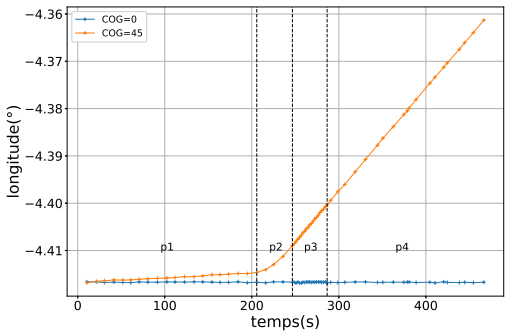

FIGURE 1.5 – Évolution de la longitude simulée au cours du temps pour COG = 0° et COG = 45°.

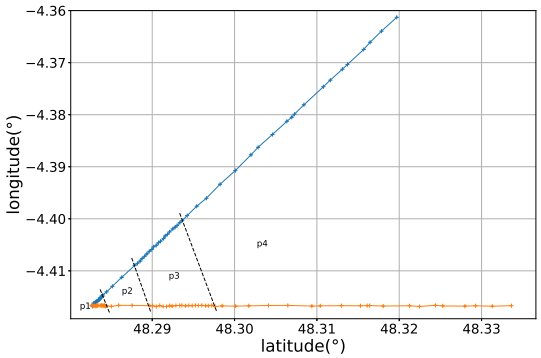

FIGURE 1.6 – Évolution de la position simulée au cours du temps pour COG = 0° et COG = 45°.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}\left(\tilde{Z}_{\lambda,n}^2 + \tilde{Z}_{\phi,n}^2\right)}{n}} \tag{1.56}$$

avec $\tilde{Z}_{\lambda,n} = \hat{Z}_{\lambda,n} - Z_{\lambda,n}$, $\hat{Z}_{\lambda,n}$ la latitude estimée et $Z_{\lambda,n}$ la latitude vraie, $\tilde{Z}_{\phi,n} = \hat{Z}_{\phi,n} - Z_{\phi,n}$, $\hat{Z}_{\phi,n}$ la longitude estimée et $Z_{\phi,n}$ la longitude vraie.

Pour finir, les variations du seuil du test de validation sur la latitude et la longitude sont affichées sur les Figures 1.10 et 1.11. Il faut noter qu'il s'agit du seuil de validation sur l'innovation et non pas de celui sur l'innovation normalisée, comme présenté dans la partie 1.6.1. La seuil sur l'innovation a pour valeur $\sqrt{\gamma \times S_n} = 3.29\sqrt{S_n}$. La valeur maximale obtenue de ce seuil vaut 250 m et sa valeur minimale 35 m. Ainsi, dans le cas le moins favorable, toute falsification de position supérieure à 250 m est détectée. Pour cette valeur, la probabilité de non-détection ou risque de deuxième espèce β est très faible. En effet, potentiellement, les falsifications peuvent atteindre 40 km sur la latitude et la longitude à cause de la portée de l'AIS. En considérant une répartition uniforme, comme dans [166], de la falsification de position, au maximum $\beta = \frac{250}{40000} = 0{,}006$ pour $\alpha = 0{,}001$.

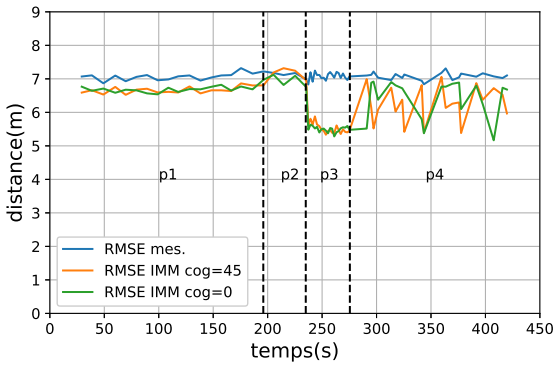

FIGURE 1.7 – Évolution du RMSE de la position estimée par l'IMM pour COG = 0° et COG = 45° en fonction du temps

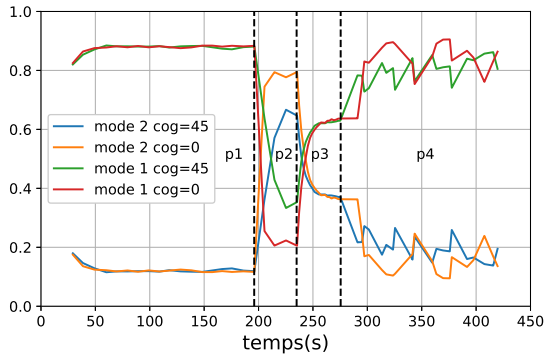

FIGURE 1.8 – Évolution des probabilités des modes après mise à jour en fonction du temps pour la latitude

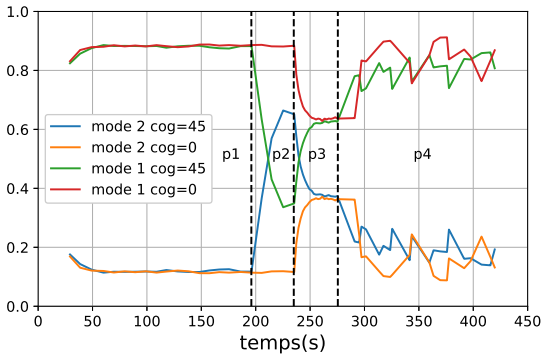

FIGURE 1.9 – Évolution des probabilités des modes après mise à jour en fonction du temps pour la longitude

1.8.2 Comparaison avec les méthodes existantes

Nous comparons la sensibilité obtenue par notre stratégie de détection des falsifications de position avec les autres stratégies citées plus haut dans l'état de l'art présenté dans l'Introduction :

— la méthode [117] calcule la position des bateaux en considérant la différence des temps d'arrivée des messages AIS entre l'émission et la réception sur plusieurs récepteurs AIS espacés les uns des autres. Cette méthode a une sensibilité de quelques centaines de mètres dans des conditions optimales et plusieurs kilomètres ailleurs ;

— la méthode [72] propose d'utiliser un ou plusieurs radars pour connaître la position des bateaux. La sensibilité de cette méthode dépend du nombre de radars utilisés. Plus il y a de radars, plus la sensibilité est élevée. Dans le cas du suivi d'un bateau qui ne manœuvre pas et qui émet ses messages avec une période $\Delta T = 10$ s, notre méthode est aussi précise que si trois radars étaient utilisés. Si $\Delta T = 2$ s notre méthode devient aussi efficace que si 10 radars étaient utilisés. Dans le cas du pistage d'un bateau qui manœuvre, notre stratégie a la même efficacité que cette

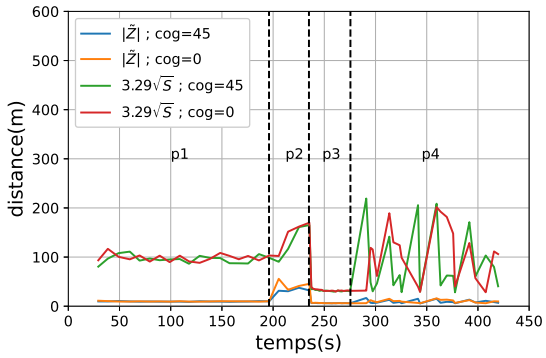

FIGURE 1.10 – Évolution du seuil de validation et de l'innovation pour la latitude en fonction du temps

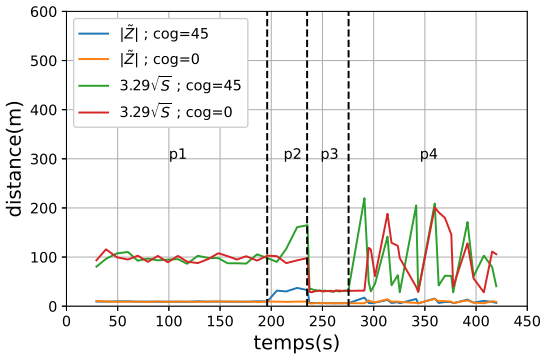

FIGURE 1.11 – Évolution du seuil de validation et de l'innovation pour la longitude en fonction du temps

stratégie lorsqu'un seul radar est utilisé ;

— la méthode [55] qui utilise l'effet Doppler pour estimer la position a une sensibilité de plus de 10 km ;

— la méthode [138] propose de suivre les bateaux avec un filtre IMM pour détecter les falsifications de position. C'est la solution la plus aboutie dans la littérature. Son filtre IMM utilise un filtre de Kalman étendu, modélisant la dynamique du bateau lorsqu'il manœuvre, et un filtre de Kalman, modélisant la dynamique du bateau lorsqu'il ne manœuvre pas. Ces deux filtres de Kalman sont exécutés en parallèle et le système de coordonnées utilisé est cartésien. L'article affirme qu'une sensibilité aux falsifications de 21 m est obtenue, cependant les calculs ont été faits en fixant l'erreur de modèle sur la vitesse à 0,07 m s^{-1} (0,14 kn) pour $\Delta T = 1$ s, alors que nous l'avons fixé à 0,5 m s^{-1} (0,97 kn). Cela sous-entend qu'ils ne peuvent suivre que des bateaux avec une très grosse inertie. Par ailleurs, trop peu de détails sont donnés sur leurs simulations pour que l'on puisse comparer les performances de leur algorithme de pistage avec les nôtres.

1.8.3 Évaluation du suivi de vitesse

Nous réalisons aussi des simulations de MC comprenant 10 000 exécutions pour évaluer la sensibilité de l'algorithme détectant les falsifications de vitesse. Ces simulations permettent aussi de fixer la valeur du seuil validant les mesures. Ce seuil doit être associé à une probabilité de fausse alarme α de 0,01. La probabilité de fausse alarme est fixée à 0,01 et non à 0,001 comme pour le test sur la position afin de diminuer la valeur de la probabilité de non détection (β).

La même simulation de Monte Carlo que celle appliquée à la position est effectuée. La Figure 1.12 illustre le résultat obtenu. Nous constatons que la route suivie par le bateau n'a pas d'influence sur la sélectivité du test. Pour les simulations effectuées, le seuil sur la vitesse vaut en moyenne au maximum 8,5 kn et au minimum 4,2 kn. Sachant que la

vitesse maximale d'un bateau pisté est de 45 kn et en supposant, comme pour la position, une répartition uniforme des falsifications sur la vitesse, l'erreur de deuxième espèce vaut respectivement $\beta = 0{,}19$ et $\beta = 0{,}1$. Ces valeurs ne sont pas suffisantes pour que le test soit déclaré puissant ($\beta < 0{,}05$). Néanmoins, ce test n'est pas pour autant inutile, il permet de détecter certaines falsifications de vitesse de quelques nœuds et vient en appui du test sur les mesures de position.

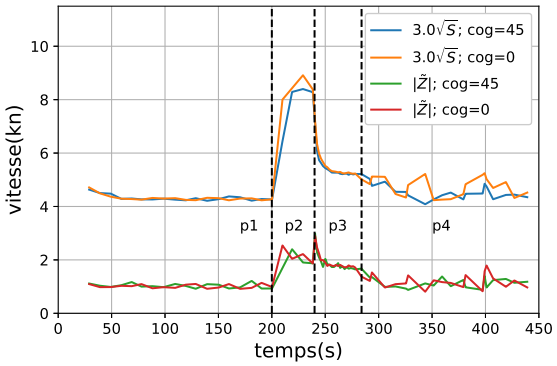

FIGURE 1.12 – Évolution du seuil de validation et de l'innovation pour la vitesse en fonction du temps

1.9 Expérimentation sur des données réelles

Dans cette partie, nous testons, sur 100 000 messages enregistrés dans la rade de Brest, la stratégie développée dans ce chapitre, qui applique l'algorithme détectant les falsifications de position et l'algorithme détectant les falsifications de vitesse.

1.9.1 Mesures utilisées

Les données collectées pour tester les deux algorithmes développés dans ce chapitre ont été obtenues suite à une collaboration avec le Cerema (Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement) de Brest lors de l'édition 2002 de l'Ocean Hackathon à Brest. Le Cerema est un établissement public à caractère administratif apportant son expertise scientifique et technique à l'élaboration, à la mise en œuvre et à l'évaluation des politiques publiques en matière d'aménagement durable et d'urbanisme, notamment dans les domaines de la sécurité routière et maritime, de la mer et du littoral. L'affiche de l'Hackathon est présentée sur la Figure 1.13. Pour cette édition, avec l'équipe du Cerema, nous avons développé une application permettant de détecter les falsifications de messages AIS. Dans le développement, j'apportais les algorithmes que j'avais développés durant ma thèse et le Cerema amenait des données expérimentales. Nous n'avons pas remporté l'Hackathon, mais cette expérience a permis de rencontrer

FIGURE 1.13 – Présentation de l'affiche de l'Hackathon 2022

des personnes impliquées dans la sécurité maritime au niveau local et de resserrer la collaboration entre l'Irenav et le Cerema.

Par ailleurs, grâce à cette collaboration, j'ai pu obtenir du Cerema un très grand jeu de données pour tester mes algorithmes. En effet, le Cerema a collecté, grâce à une station de base AIS, durant une campagne de mesures de 9 h, 100 000 messages provenant de 73 bateaux. La station de base a extrait les trames NMEA (National marine electronics association) des messages reçus qui contiennent les informations listées en Annexe D. Parmi ces informations nous avons utilisé celles concernant la position, la vitesse et l'identité. L'instant d'arrivée des messages était aussi fourni avec une précision à la nanoseconde près. Nous représentons sur les Figures 1.14 et 1.15 la position des bateaux ayant émis des messages et la position du Cerema qui est l'endroit où était positionnée l'antenne pour recevoir les messages.

1.9.2 Résultats de l'expérimentation

Les algorithmes détectant les falsifications de position et de vitesse sont testés sur les messages collectés par le Cerema. Ces algorithmes appliquent des tests de conformité aux mesures (information) de latitude, longitude et vitesse transmises dans les messages. Ces tests ont rejeté 29 fois la mesure sur la latitude, 57 fois celle sur la longitude et 132 fois celle sur le SOG. Les rejets peuvent correspondre soit à des fausses alarmes ou soit à des vraies alarmes. Néanmoins, le nombre de rejets observés est inférieur, pour les trois tests, au nombre de rejets acceptés par la valeur de la probabilité de fausse alarme (α) choisie. Pour rappel, les tests sur la latitude et la longitude acceptent une probabilité de fausse alarme de 0,1 % et celui sur le SOG accepte une probabilité de fausse alarme de 1 %. Toutes les alarmes sont représentées sur les Figures 1.16, 1.17 et 1.18.

Par ailleurs, pour montrer l'utilité de l'algorithme 1, un scénario de falsification a été

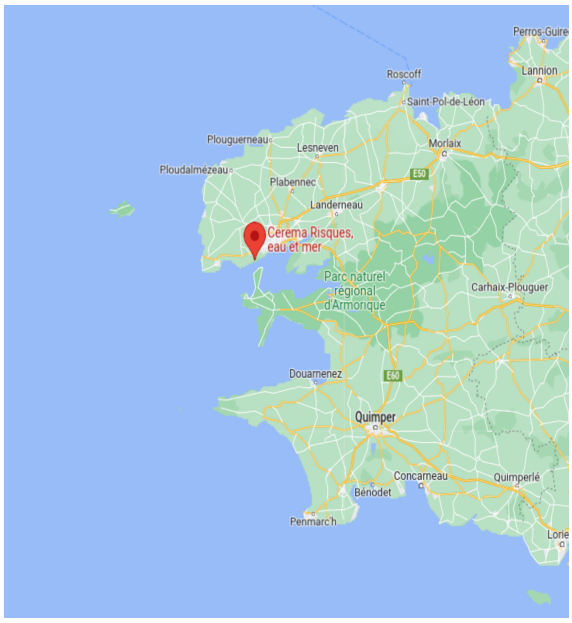

FIGURE 1.14 – Localisation du centre du Cerema à Brest.

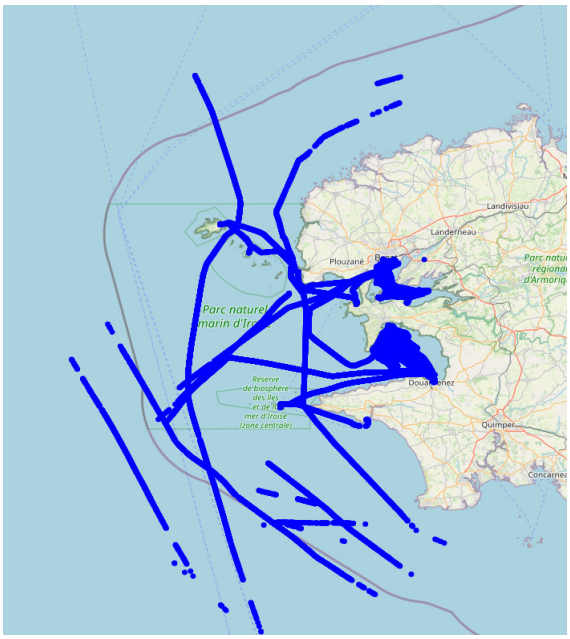

FIGURE 1.15 – Ensemble des positions enregistrées.

FIGURE 1.16 – Ensemble des alertes enregistrées sur la vitesse.

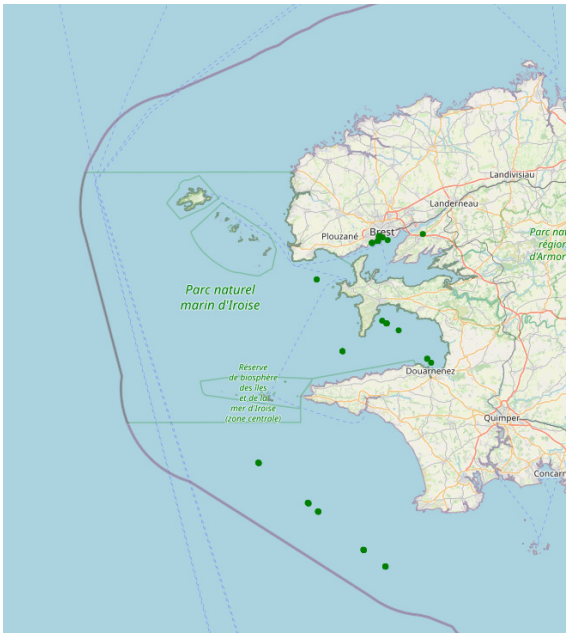

FIGURE 1.17 – Ensemble des alertes enregistrées sur la latitude.

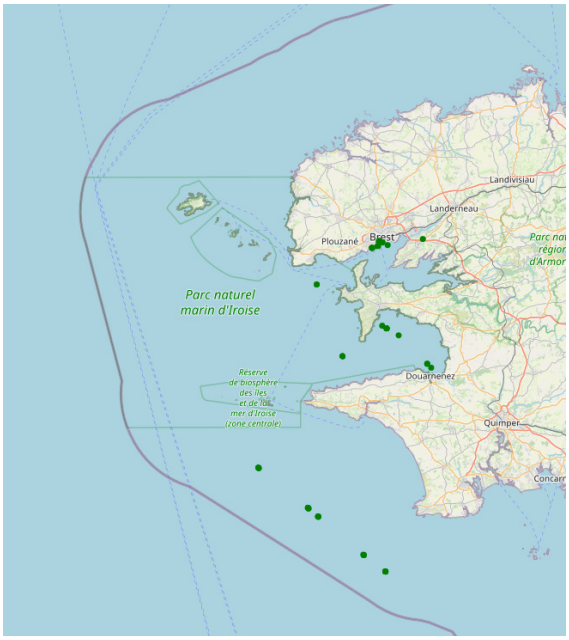

FIGURE 1.18 – Ensemble des alertes enregistrées sur la longitude.

créé à partir de données réelles concernant un des navires suivis. Dans ce scénario, inspiré de [72], un navire falsifie ses coordonnées en latitude et longitude pendant 10 min, à partir de la $174^{ème}$ seconde d'acquisition, en ajoutant 800 m à la longitude et latitude mesurées. Ce type de falsification peut être utilisé par des bateaux de pêche pour pêcher dans des zones protégées, telles que Natura 2000, sans révéler leurs véritables positions. La période d'émission des messages est de 6 s $\pm$ 20 %. L'évolution de l'innovation sur la longitude et la vitesse au cours du temps, et l'évolution de leur seuil sont présentées respectivement sur les Figures 1.19 et 1.20.

L'algorithme 1 détecte les falsifications sur la longitude : les innovations étaient supérieures de plusieurs centaines de mètres au seuil, lorsque le navire a commencé à falsifier ($174^{ème}$ seconde) et lorsqu'il a cessé de falsifier ($760^{ème}$ seconde). La même observation est faite sur la latitude. Durant ces falsifications, le seuil sur la longitude augmente. En effet, sachant que la mesure est rejetée, les équations d'estimations ne sont pas appliquées pour réduire l'incertitude sur la position et la vitesse de la cible. Il faut noter que la longitude estimée est recalée après les cinq premières erreurs rencontrées, c'est pourquoi, de nouveau, l'innovation devient correcte. Concernant la vitesse, aucune erreur n'est détectée. Le seuil de la vitesse a une valeur moyenne de 6,82 kn, si nous ne prenons pas en considération les valeurs induites par les falsifications. Cette valeur est conforme à la valeur trouvée par simulation dont les résultats sont présentés sur la Figure 1.12.

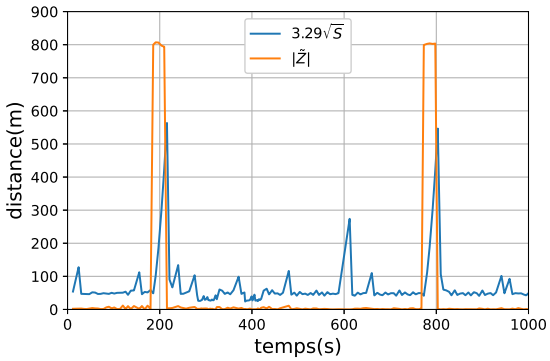

FIGURE 1.19 – Évolution de l'innovation et du seuil sur la longitude mesurée en fonction du temps.

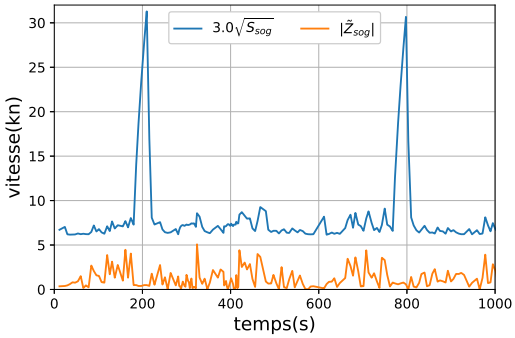

FIGURE 1.20 – Évolution de l'innovation et du seuil sur la vitesse mesurée en fonction du temps.

1.10 Conclusion

Dans ce chapitre, nous avons proposé une stratégie qui détecte les falsifications de la position et de la vitesse transmises dans les messages AIS. Cette stratégie applique un algorithme pistant la position des bateaux grâce à un filtre IMM. Ce filtre utilise le système de coordonnées sphériques du GPS pour décrire le mouvement des bateaux et réprésenter leurs positions, ce qui est une première dans la littérature scientifique pour ce type d'application. Pour une probabilité de fausse alarme $\alpha = 0{,}1$ %, la sensibilité de la stratégie aux falsifications de position est, dans le pire des cas, de 250 m, et, dans le meilleur des cas, de 35 m. Pour ces valeurs, les probabilités de non détection sont très faibles et valent respectivement $\beta = 0{,}6$ % et $\beta = 0{,}09$ %. La sensibilité aux falsifications de cette stratégie est meilleure que celles proposées par les autres stratégies disponibles dans la littérature. Par ailleurs, concernant la vitesse, pour une probabilité de fausse alarme $\alpha =$ 1 %, la probabilité de non détection vaut, au maximum, 19 %. Cette dernière vérification permet d'assurer la fiabilité de la vitesse reçue dans les messages. Cette vérification est importante, car la vitesse intervient dans la stratégie 2, que nous présentons dans le prochain chapitre, pour contrôler le respect du mode d'accès TDMA.

Chapitre 2

Détection des faux messages et des bateaux fantômes

2.1 Introduction

Dans ce chapitre, nous présentons la stratégie 2 détectant les faux messages et les bateaux fantômes. Cette stratégie vérifie que chaque bateau respecte le mode d'accès TDMA lorsqu'il transmet ses messages. Ce mode d'accès définit la période d'émission des messages et le procédé de réservation des intervalles de temps (TS (Time slot)), durant lesquels les messages sont transmis. Chaque bateau émet des messages avec une période cohérente à sa vitesse de navigation et applique le procédé de réservation TDMA pour pré-annoncer les TS qu'il utilisera pour transmettre ses prochains messages. Cette stratégie vise à détecter les messages créés *ex nihilo* qui se caractérisent, dans la réalité, par l'apparition de bateaux fantômes. Les communications AIS n'étant pas sécurisées, il est très facile d'émettre, avec une radio logicielle, de faux messages. L'efficacité et la robustesse de notre stratégie sont testées sur des données réelles provenant d'une campagne de mesures effectuées près de Brest, ainsi que sur des données de l'OTAN reconnues comme ayant été falsifiées.

2.2 Mode d'accès TDMA

Pour rappel, un transpondeur AIS est composé d'un émetteur de données VHF et est connecté à plusieurs équipements dont le GPS fait partie. Le transponder reçoit du GPS les informations concernant la position, la vitesse et la route du navire, et les transmet, avec les informations statiques et celles liées au trajet, sous forme numérique, via deux canaux VHF dédiés à l'AIS. Pour éviter que des transpondeurs émettent leurs messages en même temps, ce qui entraînerait des interférences et des pertes de données, le mode d'accès TDMA est appliqué et organise les communications. Ce mode d'accès répartit dans le temps l'accès, par les transpondeurs, aux canaux de communication. Ainsi, chaque transpondeur revendique des intervalles de temps durant lesquels il transmettra ses messages. Dans ce qui suit, nous présentons les caractéristiques techniques et le principe de ce mode d'accès.

2.2.1 Caractéristiques techniques

Toutes les caractéristiques techniques et opérationnelles de l'AIS ont été définies par l'ITU et sont contenues dans une norme dont la dernière mise à jour date de 2014 [133]. Cette norme définit, en particulier, le mode d'accès TDMA qui organise les communications entre bateaux pour prévenir l'apparition d'interférences. Ce mode d'accès fait partie de la couche liaison du modèle OSI (Open systems interconnection). Il divise le temps en trames d'une minute dont chacune contient 2250 TS comme présenté sur la Figure 2.1. Chaque TS dure 26,7 ms et permet de transmettre un message. Les TS sont numérotés de 0 à 2249 et chaque transpondeur AIS se synchronise grâce à la minute UTC (Coordinated universal time) qui sert de référence de temps commune.

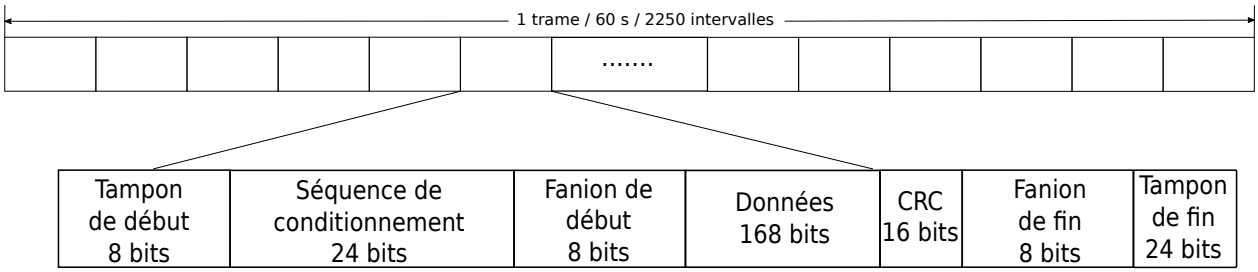

FIGURE 2.1 – Structure d'un message AIS et d'une trame [133].

La structure d'un message AIS est présentée sur la Figure 2.1. Le protocole HDLC (High-level data link control) est appliqué pour délimiter le début et la fin d'un message. La structure d'un message est composée d'un tampon de début, d'une séquence de conditionnement, d'un fanion de début, des données, du CRC (Contrôle de redondance cyclique), du fanion de fin et du tampon de fin comme illustré sur la Figure 2.1. Les fanions de début et de fin permettent respectivement de détecter le début et la fin des données contenant les informations du message. La séquence de conditionnement permet au récepteur de se synchroniser au message. Le tampon de début permet de laisser le temps à l'émetteur de passer en pleine puissance lors de l'émission. Le tampon de fin permet de contenir les bits rajoutés par le bourrage de zéro, et de laisser du temps pour compenser le retard dû au temps de propagation et au jitter. La vérification CRC utilise le polynôme 16 bits défini par la norme ISO/IEC 13239 :2002 [1]. Les données d'un message dépendent de l'identité du message envoyé ; il en existe 27. Les messages qui nous concernent dans ce travail sont ceux de report de position pour les transpondeurs de classe A. Ils ont une identité égale à 1, 2 ou 3 et représentent 80 % des messages émis [81]. La description des différentes informations contenues dans chaque message dont l'identité du message fait partie est faite dans l'annexe (D).

2.2.2 Fonctionnement

Les transmissions de paquets se font alternativement sur deux canaux VHF : le canal A (161,975 MHz) et le canal B (162,025 MHz). À chaque canal est allouée une trame identique à celle présentée sur la Figure 2.1. Le mode d'accès TDMA utilise quatre configurations différentes en fonction de la dynamique du bateau et du mode de fonctionnement de l'AIS pour organiser les communications. Ces configurations indiquent de quelle manière un transpondeur AIS sélectionne les TS durant lesquels il transmettra ses messages. Ces quatres configurations peuvent être de type SOTDMA (Self-organized time division multiple access), ITDMA (Incremental time division multiple access), RATDMA (Random acces time division multiple access) et FATDMA (Fixed acces time division multiple access).

La configuration de base est le SOTDMA. Elle est utilisée tant que la période d'émission des messages reste constante, c'est-à-dire tant que le bateau ne manœuvre pas. Si le bateau manœuvre (accélère, décélère, change de route) la période d'émission des messages doit être modifiée ; plusieurs messages non répétitifs sont alors transmis pour indiquer ce changement de période ; pour cela, les autres configurations sont utilisées.

Parmi ces autres configurations, il y a la configuration ITDMA. Cette configuration permet à une station AIS d'annoncer, à l'avance, les TS qu'elle utilisera pour transmettre des messages à caractère non répétitif. Il existe aussi la configuration RATDMA qui accède de manière aléatoire aux canaux de communication. Elle est appliquée lorsqu'un navire a besoin d'utiliser un TS qui n'a pas été pré-annoncé. Cela arrive lors du démarrage de l'AIS, ou au tout début de la manœuvre. Le mode d'accès FATDMA n'est utilisé que par les stations côtières pour transmettre des messages répétitifs. Les messages transmis par ce dernier type de configuration ne sont pas considérés dans nos travaux, car ils proviennent de stations de base utilisées par les autorités de contrôle, et non pas par des stations mobiles utilisées par les bateaux.

Lorsqu'un bateau démarre son AIS, il va écouter pendant une minute les communications de messages AIS, sans émettre de message, pour connaître tous les TS réservés. Ensuite, connaissant les TS déjà réservés, il transmettra ses messages durant les TS libres pour éviter qu'ils interfèrent avec les messages émis par les autres navires. L'écoute du réseau AIS ne couvre que les messages émis par des bateaux à portée du récepteur, ce qui correspond à une cellule de rayon 25 à 40 km. Les TS qui paraîtront libres dans cette cellule seront peut-être occupés dans une autre cellule d'un autre AIS.

2.3 Vérification de la période d'émission des messages

Nous présentons dans cette section notre premier algorithme (Algorithme 1) qui vérifie le respect du mode d'accès TDMA. Cet algorithme vérifie que la période d'émission des

messages d'un bateau est cohérente avec sa vitesse.

2.3.1 Conditions d'application

La période d'émission des messages dépend de la vitesse des bateaux et de leur comportement (manœuvre ou non). Cela est présenté dans le Tableau 2.1 tiré de la norme AIS [133]. Lorsque le bateau ne manœuvre pas, l'état de communication utilisé est le SOTDMA, comme présenté sur le Tableau D.5 de l'annexe (D) ; l'*identité* (`ID`) des messages est donc égale à 1 ou 2. Lorsque le bateau est à l'encre ou au mouillage, le *Statut de navigation* (`Navs`) est égal à 1 ou 5. Lorsque le bateau manœuvre, la période d'émission des messages doit changer et être diminuée ; de nouveaux messages sont émis par le bateau en plus des messages qu'il émettait déjà ; pour ces nouveaux messages émis, les modes d'accès ITDMA et RATDMA sont utilisés avec l'état de communication ITDMA. Cela est présenté dans le Tableau D.5 de l'annexe (D). Pour l'état de communication ITDMA, l'`ID` des messages vaut 3. Ainsi, lorsqu'un bateau commence à manœuvrer, deux états de communication peuvent être utilisés : l'état de communication SOTDMA pour les messages transmis à la période d'émission d'avant la manœuvre, et l'état de communication ITDMA pour les messages transmis en plus du fait de la manœuvre et de la nouvelle période d'émission à respecter. L'`ID` des messages peut alors être 1, 2 ou 3, mais il n'est pas possible d'avoir deux messages consécutifs avec une `ID` égale à 1 ou 2. La manœuvre (changement de route) sera donc détectée si deux messages consécutifs n'ont pas une `ID` égale à 1 ou 2 et si le `Navs` est différent de 1 ou 5. Toutes ces conditions sont récapitulées dans le Tableau 2.1.

Tableau 2.1 – Période d'émission des messages

Dynamique du bateau	**Période**	**Conditions**
Bateau à l'encre ou au mouillage se déplaçant à moins de 3 kn [1]	3 min	`Navs`== 1 OU `Navs`== 5
Bateau à l'encre ou au mouillage se déplaçant à plus de 3 kn	10 s	`Navs`== 1 OU `Navs`== 5
Bateau se déplaçant à 0−14 kn	10 s	$\texttt{ID}_{n-1}$==$\texttt{ID}_n$ == 1 OU 2
Bateau se déplaçant à 0−14 kn et changeant de route	$3 + \frac{1}{3}$ s	Autres cas
Bateau se déplaçant à 14−23 kn	6 s	$\texttt{ID}_{n-1}$==$\texttt{ID}_n$ == 1 OU 2
Bateau se déplaçant à 14−23 kn et changeant de route	2 s	Autres cas
Bateau se déplaçant à plus de 23 kn	2 s	$\texttt{ID}_{n-1}$==$\texttt{ID}_n$ == 1 OU 2
Bateau se déplaçant à plus de 23 kn et changeant de route	2 s	Autres cas

2.3.2 Algorithme 1

Pour chaque message reçu, l'Algorithme 1 vérifie l'hypothèse $H_{0,RI}$ suivante :

— $H_{0,RI}$: *Le bateau respecte le mode d'accès TDMA en ce qui concerne la période d'émission des messages.*

1. kn = nœud = 1 miles nautique/heure = 1,852 km h^{-1}.

Pour vérifier cette hypothèse, l'Algorithme 1 considère la vitesse, l'identité et le statut de navigation transmis dans le message reçu. En plus, l'intervalle de temps entre ce message reçu et le dernier message transmis par ce même bateau est aussi considéré afin de connaître sa période d'émission des messages instantanés. Si ces données considérées sont conforment à la norme du mode d'accès TDMA, dont nous avons rappelé les conditions dans le Tableau 2.1, l'hypothèse $H_{0,RI}$ est acceptée ($H_{0,RI} = 1$). Sinon, l'hypothèse $H_{0,RI}$ est rejetée ($H_{0,RI} = 0$).

La précision de la période d'émission des messages est de ±20 % lorsque le navire ne manœuvre pas, comme expliqué dans la prochaine partie 2.4.1. Lorsque le navire manœuvre (changement de route), cette précision est de ±90 %. La valeur de 90 % n'est pas spécifiée par la norme mais est issue d'un calcul statistique sur 6600 messages afin d'avoir une probabilité de fausse alarme $\alpha = 5$ %. Ainsi, si un bateau se déplace à 16 kn et ne manœuvre pas, la période d'émission de ses messages doit se trouver dans l'intervalle [4,8; 7,2] et s'il manœuvre, la période doit se trouver dans l'intervalle [0,2; 4,8]. Cet algorithme est sensible aux conditions environnementales qui, si elles sont mauvaises, peuvent provoquer de fausses alarmes. En effet, dans ce cas, plusieurs messages peuvent ne pas être reçus, ce qui augmente la période entre deux messages reçus, et rend cette période non conforme à la norme. Cela arrive relativement souvent.

2.4 Vérification du respect de la réservation des TS

Pour prévenir des interférences, le mode d'accès TDMA impose aussi à chaque bateau de réserver en avance les TS durant lesquels ses messages seront transmis. La réservation des TS s'effectue en insérant, dans les messages, des données spécifiques (`STO`, `Offset`, `Keep flag` et `Slot Increment`). Grâce à ces données, expliquées ci-dessous, tous les transpondeurs situés dans une même zone de navigation connaissent les prochains TS réservés et disponibles.

C'est le respect de ce procédé de réservation par les bateaux que nous contrôlons dans le second algorithme présenté dans ce chapitre. Cet algorithme vérifie que chaque message reçu avait été préalablement réservé. Avant de présenter cet algorithme, nous commençons par présenter, ci-dessous, le fonctionnement des procédés de réservation des modes d'accès SOTDMA et ITDMA. Le mode d'accès RATDMA n'est pas présenté, car il sélectionne ses TS de manière aléatoire sans appliquer de procédé de réservation. Nous présentons, en plus, sur la Figure 2.2, un exemple de réservation de TS sur les deux canaux A et B dans un même espace maritime durant une trame d'une minute. Sur cette Figure, on observe 15 rangées de 2×150 TS (une pour chaque canal), ce qui correspond aux 2250 TS utilisables sur chacun des canaux par un bateau pour transmettre ses messages. Les TS apparaissant en blanc sont ceux qui n'ont pas été utilisés. Ils sont nombreux ce qui montre que le trafic maritime dans la zone observée n'est pas très dense.

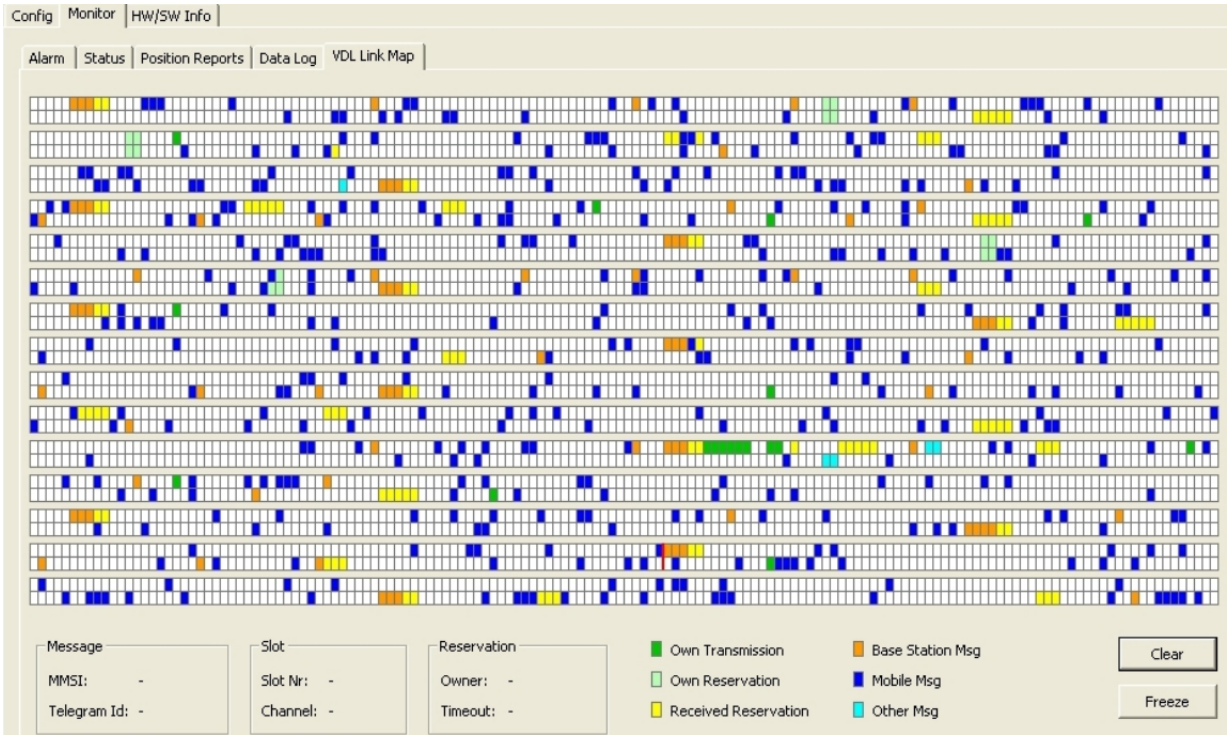

FIGURE 2.2 – Présentation de tous les TS réservés parmi tous les TS disponibles sur les deux canaux durant une trame d'une minute

2.4.1 Mode d'accès SOTDMA

Une vue générale du procédé de réservation des TS par le mode d'accès SOTDMA est illustrée sur la Figure 2.3. Sur cette Figure apparaît une partie des 2250 TS qui composent les canaux A et B. L'incrément nominal NI (Nominal increment) correspond à la période d'émission des messages. Les NS (Nominal slot), représentés en noir sur la Figure 2.3, sont définis en tenant compte du NI et du NSS (Nominal start slot). Le NSS est, pour chaque canal, le premier TS utilisé par un navire pour transmettre un message sur le canal et sert de point de référence pour fixer NS en considérant la période d'émission des messages. Les NS sont les centres des intervalles de sélection (SI (Selection interval)) dans lesquels les TS de transmission nominaux (NTS (Nominal transmission slot)), représentés en gris foncé sur la Figure 2.3, sont sélectionnés selon une loi uniforme. Leur largeur est égale à 0,2×NI et est représentée en gris clair sur la Figure. Les NTS définissent les TS pendant lesquels les messages sont transmis. Les paramètres NSS, NS, SI et NI sont maintenus constants tant que la période d'émission reste constante. Cependant, si la période est modifiée, alors ces paramètres sont également modifiés.

2.4.2 Réservation des TS par le mode d'accès SOTDMA

Le procédé de réservation associé au mode d'accès SOTDMA utilise les données `STO` et `Offset`. Ces données sont contenues dans les informations concernant l'état de commu-

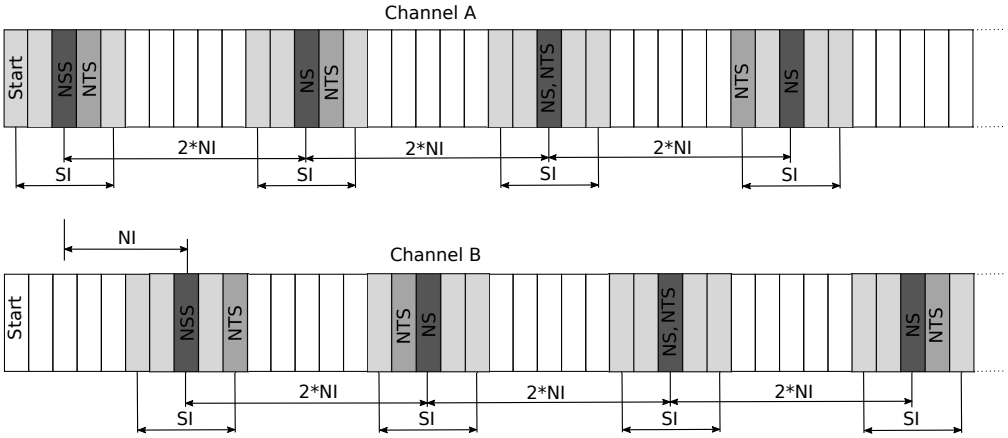

FIGURE 2.3 – Diagramme du mode d'accès SOTDMA [133].

nication des messages comme rappelé sur le Tableau de l'annexe (D). Avec ces données, le procédé suivant est appliqué pour réserver les prochains NTS :

— Lorsqu'un NTS est sélectionné, un nombre compris entre 3 et 7, et appelé `slot time-out` (`STO`), est généré de manière aléatoire. Il spécifie le nombre de trames restantes pour lesquelles ce NTS sera utilisé sur le même canal ;
— Pour ces trames restantes, le `STO` sera décrémenté de 1 à chaque nouvelle transmission de message durant ce NTS ;
— Lorsque la valeur de `STO` devient égal à 0, un nouveau NTS doit être sélectionné et réservé pour les prochaines trames. Il est sélectionné aléatoirement dans le SI de la trame suivante avec le même NS. Un `Offset` est alors intégré au message transmis et indique le nombre de TS situés entre le NTS actuel et le nouveau réservé.

Des exemples de ce procédé de réservation sont présentés dans les tableaux 2.2 et 2.3.

2.4.3 Réservation des TS par le mode d'accès ITDMA

Le procédé de réservation associé au mode d'accès ITDMA utilise les données `Keep flag` et `Slot increment` (`Slot incr.`) contenues dans l'information concernant l'état de communication des messages (Tableau D). Avec ces données, le procédé suivant est appliqué pour réserver les prochains NTS :

— Lorsque le `Keep flag` est fixé à 1, il informe que le même NTS reste alloué pour une trame supplémentaire sur le même canal. Lorsqu'il est fixé à 0, il informe que le même NTS sera libre pour la trame suivante ;
— Le `Slot Incr.` indique le nombre de TS séparant le NTS actuel au nouveau NTS réservé selon la formule : `Slot incr.`$=\text{NTS}_{nouveau}-\text{NTS}_{actuel}$. Le nouveau NTS est réservé sur le même canal.

2.4.4 Enregistrement des TS réservés

L'Algorithme 2 utilise les informations intervenant dans le procédé de réservation des TS pour connaître les prochains TS réservés. La représentation graphique de ce procédé est donnée par l'organigramme de la Figure 2.4. Les NTS (TS réservés) sont insérés dans une liste qui est créée pour chaque canal, (`ListNTS_A` pour le canal A et `ListNTS_B` pour le canal B). Ces listes sont mises à jour à chaque commencement de nouvelle trame. Cette étape soustrait 2500 à toutes les valeurs NTS des deux listes, et après, supprime chaque NTS ayant une valeur négative. Cette suppression permet de ne conserver que les NTS réservés pour cette trame ou les suivantes. Sur la Figure 2.4, seul le procédé à suivre dans le cas où un message a été reçu sur le canal A est représenté. Le même procédé est appliqué si un message est reçu sur le canal B. Dans ce cas, le NTS est ajouté à la liste `ListNTS_B`.

Le numéro du TS est déterminé grâce au temps d'arrivée des messages (TOA (Time of arrival)) et l'équation suivante :

$$\mathrm{TS} = \mathrm{round}\left(\mathrm{mod}\left(\mathrm{TOA}_s, 60\right)\frac{2250}{60}\right) \tag{2.1}$$

où mod est le reste de la division et TOA_s est le TOA du message exprimé en secondes avec une précision à la ms.

Ces numéros de TS calculés doivent être synchronisés avec les numéros de TS du mode d'accès TDMA synchronisés sur la minute UTC. Ceci peut être fait en utilisant les données `Slot number` qui sont encapsulées dans chaque message AIS ayant un état de communication SOTDMA et un `STO` avec une valeur paire. Des exemples qui illustrent le procédé de réservation de TS sont présentés dans les tableaux 2.2 et 2.3 de la sous-section 2.6.1.

2.4.5 Algorithme 2

Pour chaque message reçu, l'Algorithme 2 vérifie l'hypothèse $H_{0,TS}$ suivante :

— $H_{0,TS}$: *le bateau respecte le mode d'accès TDMA en ce qui concerne le procédé de réservation des TS.*

Si le TS a été préalablement réservé, le test accepte l'hypothèse $H_{0,TS}$ ($H_{0,TS} = 1$), sinon il rejette l'hypothèse. L'algorithme comporte une phase d'initialisation pour chaque bateau qui dure une minute. Après une minute, assez de messages ont été reçus pour connaître les prochains TS réservés. Les messages provenant du mode d'accès RATDMA ne sont pas pris en compte comme nous l'avons dit. Ces messages sont, soit le premier message transmis, durant une action de manœuvre, ou après le démarrage, soit caractérisés par la donnée `repeat_indicator` qui est égale à 1.

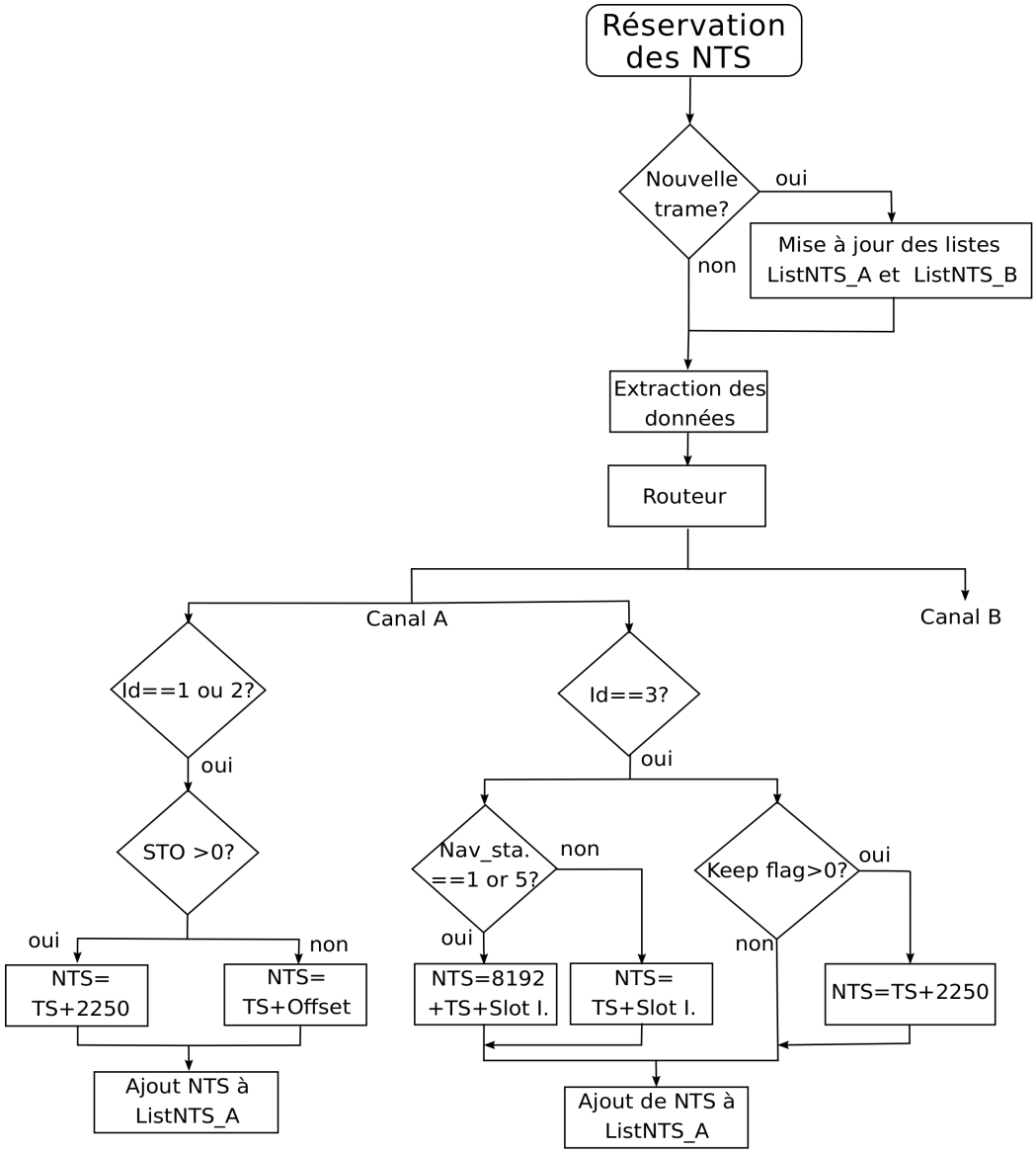

FIGURE 2.4 – Organigramme du procédé de réservation des NTS.

2.5 Présentation de la stratégie 2

La stratégie 2, détectant les faux messages et les bateaux fantômes, se compose des deux algorithmes présentés dans ce chapitre. Ces algorithmes sont exécutés en parallèle, et vérifient le respect du mode d'accès TDMA par les bateaux. L'architecture de cette stratégie est présentée sur la Figure 2.5. L'Algorithme 1 vérifie que la période d'émission des messages de chaque bateau respecte la norme AIS et l'Algorithme 2 vérifie que chaque bateau émet un message durant un TS qu'il a préalablement réservé.

En général, la stratégie renvoie beaucoup d'alarmes, que ce soit pour l'Algorithme 1 ou 2. Cela peut être dû à des problèmes techniques des transpondeurs ou à de mauvaises conditions environnementales qui causent régulièrement la non-réception de messages. Un prétraitement est donc appliqué à ces alarmes afin de faire un premier tri, et d'alerter seulement si une situation anormale est détectée. Ce prétraitement calcule le pourcentage

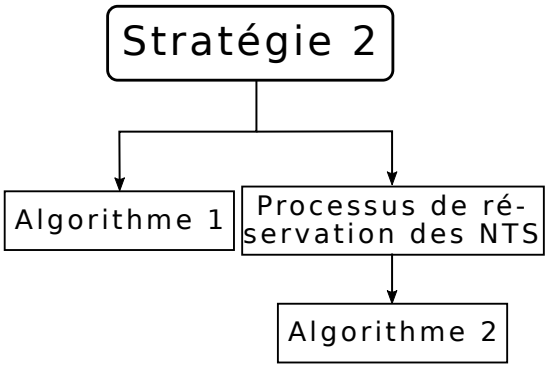

FIGURE 2.5 – Architecture de la stratégie 2.

d'alarmes détectées en fonction du nombre de messages reçus. Le pourcentage d'alarmes est une moyenne mobile calculée sur les 15 dernières trames (1 trame = 1 min) lorsque, au moins, le nombre de messages reçus est supérieur au nombre de messages reçus pendant trois trames complètes. Cette dernière condition est imposée pour s'assurer d'avoir reçu un nombre de messages suffisant pour qu'ils soient représentatifs du comportement du bateau.

2.6 Expérimentation

Nous testons les stratégies développées sur des données réelles provenant de campagnes de mesures, ainsi que sur des données reconnues par l'OTAN comme étant falsifiées [119].

2.6.1 Données provenant d'un seul bateau

Pour faciliter la compréhension du procédé de réservation des TS, nous présentons, sur la Figure 2.6, les numéros des TS utilisés par un seul navire pour transmettre ses messages durant 30 min. Le navire utilise les modes d'accès SOTDMA, ITDMA et RATDMA. Visuellement, nous pouvons discerner deux périodes d'émission sur la Figure 2.6 : une période égale à 225 TS et l'autre égale à 75 TS. L'histogramme présenté sur la Figure 2.6 donne aussi une idée des valeurs du SI. Par ailleurs, les données intervenant dans le procédé de réservation et émises durant deux trames consécutives par ce même navire sont reportées dans les tableaux 2.2 et 2.3. Lorsque l'`ID` est égal à 1, le mode d'accès SOTDMA est appliqué pour réserver le prochain TS. Lorsque l'`ID` est égal à 3, le mode d'accès ITDMA est appliqué pour réserver le prochain TS. Certaines données ont une valeur égale à "-1" pour indiquer qu'elles n'ont pas été transmises dans le message associé. En effet, lorsque le procédé de réservation du mode d'accès SOTDMA est utilisé, les données du procédé de réservation du mode d'accès ITDMA ne sont pas transmises. Les valeurs de RI (Reporting Interval), qui correspondent à la période d'émission des messages exprimée en nombre de TS, ont été calculées après réception des messages, et ne sont pas transmises

dans les messages.

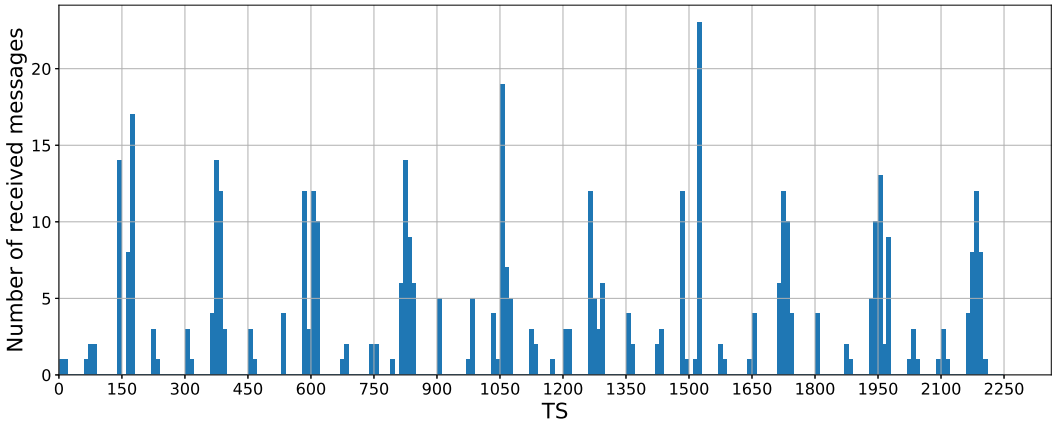

FIGURE 2.6 – Histogramme des TS utilisés par un navire durant 30 minutes d'enregistrement.

En utilisant les valeurs de TS, `STO` et `Offset`, reçues pendant la trame 1, et en appliquant le procédé de réservation du mode d'accès SOTDMA présenté sur la Figure 2.4, les TS 140, 375, 589, 847, 1051, 1290, 1525, 1956 et 2180 de la trame 2 sont réservés. Il apparaît que les messages reçus durant les TS 1971 et 2044 ont appliqué le mode d'accès RATDMA, car les TS n'avaient pas été réservés et ils marquent le début de la manœuvre du navire sur chaque canal. Comme mentionné ci-dessus, ce mode d'accès envoie des messages de manière aléatoire, les TS utilisés ne peuvent donc pas être réservés préalablement. Les `Slot Incr.` (137 et 281) reçus lors de ces deux messages, permettent de réserver les TS $2108 = 1971 + 137$ et $75 = 2044 + 281 - 2250$. En appliquant les mêmes règles, chaque message avec un `ID` égal à 3 a été réservé avant d'être reçu pendant la deuxième trame. La valeur RI (Reporting interval) est modifiée temporairement pendant la durée de la manœuvre. La manœuvre est caractérisée par les messages avec une `ID` égale à 3. Une erreur est détectée sur la trame 2. L'Algorithme 2 détecte que le message reçu durant le TS 1956 a un RI trop élevé. Cette erreur a été produite par la non-réception du message pendant le TS 1728.

TABLEAU 2.2 – Données transmises par le mode d'accès TDMA pour réserver les TS durant la trame 1.

ID	1	1	1	1	1	1	1	1	1	3	3	3	1
TS	140	375	589	847	1051	1290	1525	1728	1956	1971	2044	2108	2185
STO	7	1	3	5	4	4	7	1	3	-1	-1	-1	0
Offset	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	2245
Slot Incr.	-1	-1	-1	-1	-1	-1	-1	-1	-1	137	281	148	-1
SOG (kn)	18,4	18,3	18,3	18,3	18,3	18,4	18,5	18,5	18,5	18,5	18,6	18,5	18,4
Keep flag	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	-1
Repeat	0	0	0	0	0	0	0	0	0	0	0	0	0
Navs	0	0	0	0	0	0	0	0	0	0	0	0	0
RI	205	235	214	258	203	239	124	204	228	15	73	64	77
channel	A	B	A	B	A	B	A	B	A	A	B	A	B

Tableau 2.3 – Données transmises par le mode d'accès TDMA pour réserver les TS durant la trame 2.

ID	3	3	1	3	3	1	3	3	1	3	3	1	3	1	1	1	1	1
TS	6	75	140	226	307	375	456	535	589	681	749	847	983	1051	1290	1525	1956	2180
STO	-1	-1	6	-1	-1	0	-1	-1	2	-1	-1	4	-1	3	3	6	2	3
Offset	-1	-1	-1	-1	-1	2247	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Slot Incr.	301	151	-1	309	149	-1	293	146	-1	302	0	-1	0	-1	-1	-1	-1	-1
SOG (kn)	18,6	18,4	18,4	18,5	18,5	18,5	18,6	18,6	18,6	18,7	18,7	18,7	18,7	18,8	18,7	18,8	18,6	18,6
keep flag	0	0	-1	0	0	-1	0	0	-1	0	0	-1	0	-1	-1	-1	-1	-1
Navs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
repeat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RI	71	69	65	86	81	68	81	79	54	92	68	98	135	68	239	235	432	224
channel	A	B	A	B	A	B	A	B	A	B	A	B	B	A	B	A	A	B

2.6.2 Données réelles reconnues comme falsifiées

Nous évaluons l'efficacité de la stratégie sur des données AIS réelles pour lesquelles il existe des preuves irréfutables que les positions ont été falsifiées [119]. Ces données concernent un navire de guerre navigant en mer Noire. D'après les positions transmises, le navire a quitté Odessa et a navigué directement vers Sébastopol en s'approchant à moins de deux milles nautiques de l'entrée du port. Cependant, grâce au visionnage de l'enregistrement d'une webcam située à Odessa, on sait que le navire n'a jamais quitté le port [119].

Les données AIS ont été collectées par satellite. Sachant qu'un satellite n'est pas toujours présent au-dessus d'une même zone maritime, beaucoup de messages ne sont pas reçus et par conséquent le délai entre chaque message est généralement très long (entre 500 s et 1500 s). Pour ce délai, il n'est pas intéressant d'appliquer notre stratégie en utilisant les données collectées, car sinon le nombre d'alarmes $H_{0,RI}$ et $H_{0,TS}$ serait trop grand.

Néanmoins, l'observation des données du procédé de réservation, et en particulier les valeurs de `STO` et `Offset`, affichées sur les Figures 2.8 et 2.7, montre que le procédé de réservation n'est pas respecté. En effet, à partir du $16^{ième}$ message, même si le `STO` reste égal à 0, l'`Offset` reste également égal à 0. Cela est contraire à la norme, car dans ce cas, aucun TS n'est réservé pour transmettre les prochains messages. En effet, lorsque l'on regarde ces mêmes données présentes sur la Figure 2.4 et transmises par un autre bateau présent dans le même espace maritime et ne falsifiant pas ses données, on observe un fonctionnement conforme à la norme : lorsque le `STO` est égal à 0 alors `Offset` a une valeur différente de 0. Le $16^{ème}$ message correspond au départ d'Odessa en direction de Sébastopol et au début des falsifications. Ainsi, si la stratégie 2 avait été appliquée aux messages reçus par un transpondeur AIS situé sur un navire ou une station de base côtière proche du bateau suspect, ces erreurs auraient été détectées. La trajectoire du navire de guerre qui falsifie ses positions est affichée sur la Figure 2.11.

Le fait que le `STO` et l'`Offset` restent égals à 0 pendant toute la trajectoire falsifiée montre que ces données AIS ont été fixées sans tenir compte du procédé de réservation SOTDMA. Il s'agit vraisemblablement d'un cas d'attaque du type bateau fantôme pour

lequel les messages sont crêés *ex nihilo* et transmis sur le réseau par radio logicielle. On peut raisonnablement penser qu'il ne s'agit pas d'un cas particulier et que, souvent, lorsque des messages AIS créés *ex nihilo* sont transmis, il ne respecte pas du tout le mode d'accès TDMA ; le pourcentage d'alerte est alors égal à 100 %. Ce fait démontre la pertinence de notre stratégie de détection des bateaux fantômes par contrôle de la conformité des messages transmis avec le mode d'accès TDMA.

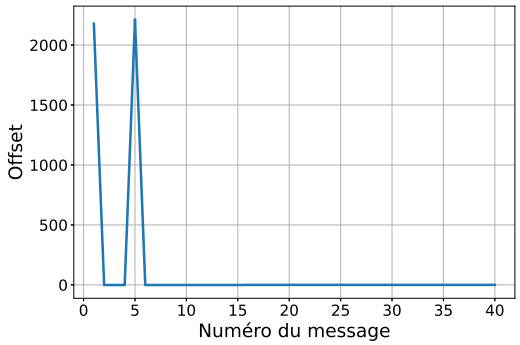

FIGURE 2.7 – Offset fonction du temps pour le bateau falsifiant ses données.

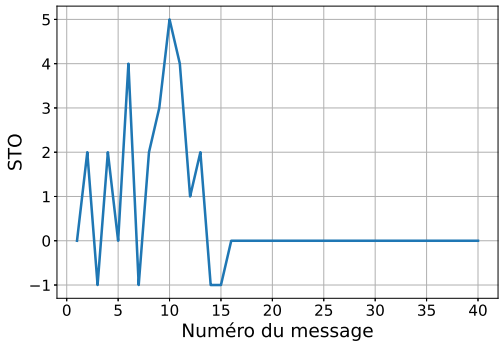

FIGURE 2.8 – STO fonction du temps pour le bateau falsifiant ses données.

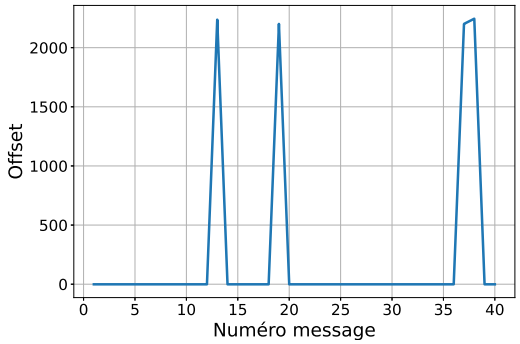

FIGURE 2.9 – Offset fonction du temps pour un bateau ne falsifiant pas ses données.

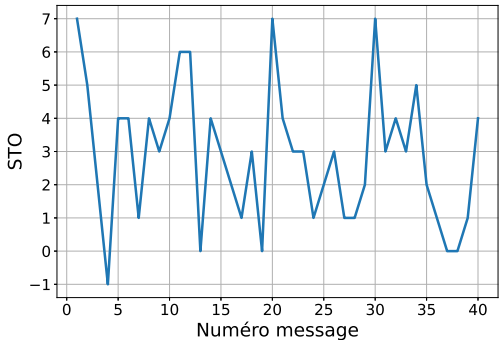

FIGURE 2.10 – STO fonction du temps pour un bateau ne falsifiant pas ses données.

2.6.3 Données réelles provenant de plusieurs bateaux

Pour finir, la stratégie 2 est appliquée aux données des 100 000 messages provenant de 73 navires et enregistrés durant 9 heures près de Brest (France) par le Céréma. Ce sont ces mêmes données que nous avons utilisées pour tester la stratégie 1 dans la partie 1.9. La localisation du Céréma est représentée par une croix rouge sur la Figure 2.12. La stratégie a rejeté 11 506 fois l'hypothèse $H_{0,RI}$ et 6397 fois l'hypothèse $H_{0,TS}$. Pour rappel, les hypothèses $H_{0,RI}$ et $H_{0,TS}$ sont acceptées si le message a respecté le mode d'accès TDMA, respectivement, en ce qui concerne la période d'émission (RI) et le procédé de réservation des TS. Le pourcentage de rejet de l'hypothèse $H_{0,RI}$ dépasse 60 % 1368 fois

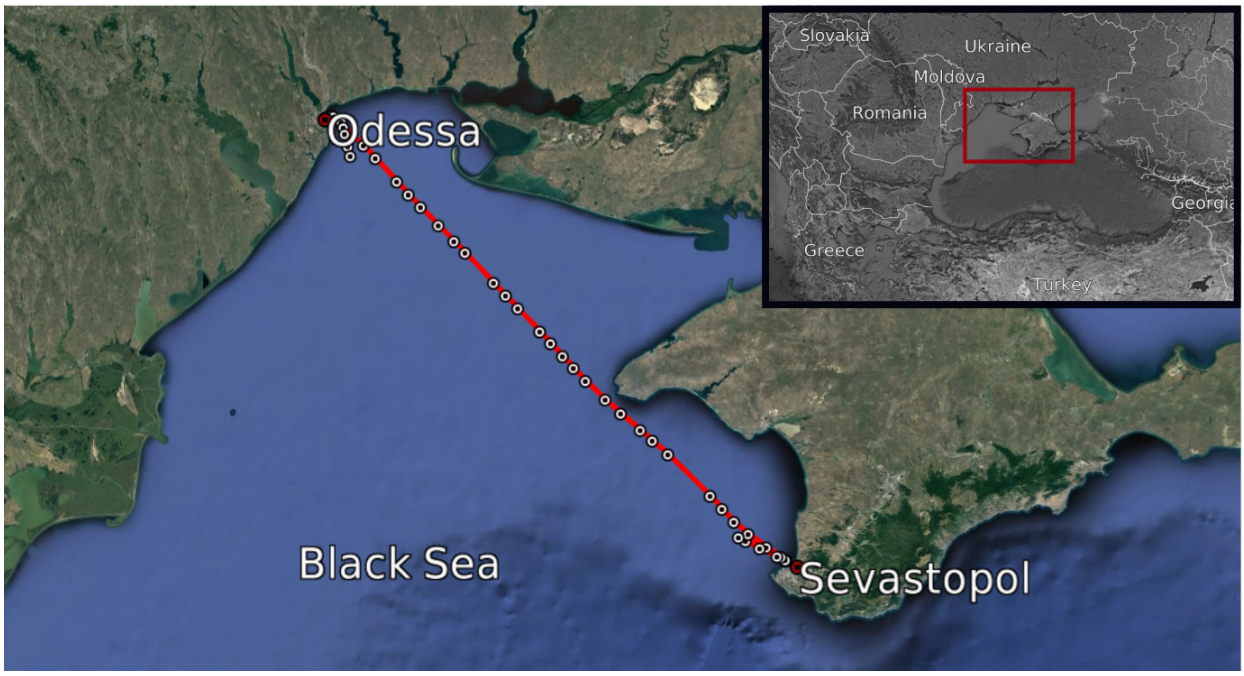

FIGURE 2.11 – Trajectoire du bateau ayant falsifié ses positions

et le pourcentage de rejet de l'hypothèse $H_{0,TS}$ dépasse 60 % 532 fois. Pour rappel, ce pourcentage d'erreur est une moyenne mobile sur les 15 dernières trames (partie 2.5). Les pourcentages de rejet de ces deux tests sont représentés sur les Figures 2.13 et 2.14 avec les positions des bateaux correspondants. Tous ces rejets d'hypothèses proviennent certainement de messages manqués. Les pourcentages dépassent les 80 % seulement durant de courtes durées de quelques minutes au maximum. Néanmoins, le nombre de rejet est élevé en raison des conditions environnementales mauvaises. En effet, l'expérience a été menée dans la rade de Brest et dans cet environnement le canal de communication n'est pas aussi ouvert qu'en pleine mer. Plusieurs messages n'ont pas été reçus, ce qui a provoqué l'apparition d'alertes correspondant aux rejets des hypothèses $H_{0,RI}$ et $H_{0,TS}$. En effet, la non-réception d'un message peut empêcher le navire de réserver un TS servant à transmettre un de ses prochains messages.

Néanmoins, en considérant ces résultats expérimentaux, nous pouvons fixer un seuil à 80 % pour toutes les alertes renvoyées par les deux algorithmes de la stratégie 2. Si les pourcentages d'alerte dépassent ce seuil durant plusieurs minutes, le navire correspondant doit être considéré comme suspect. Par exemple, nous avons montré, dans la partie précédente (2.6.2), que pour un cas réel de bateau fantôme, le pourcentage d'alerte, pour l'Algorithme 2, était de 100 % durant toute la durée des falsifications (plusieurs heures). Cette séparation entre fausses alertes (déficiences techniques) et vraies alertes (falsification et spoofing) peut paraître simple, mais est efficace. Elle peut être améliorée, en mettant en œuvre des algorithmes issus de la théorie de la détection ou de l'apprentissage automatique.

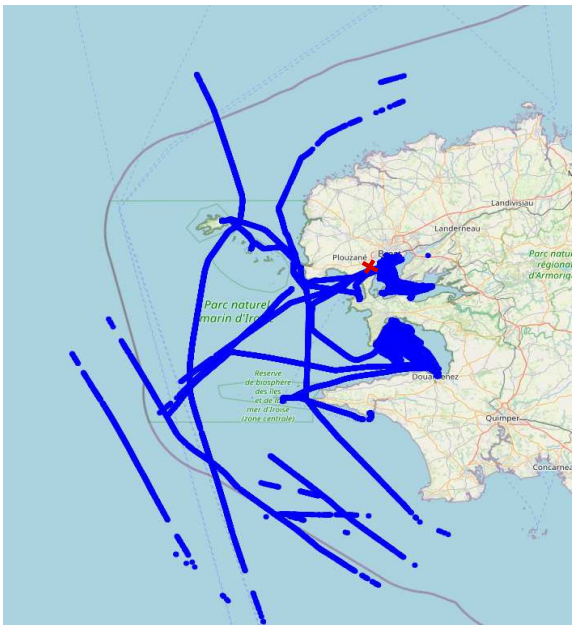

FIGURE 2.12 – Ensemble des positions enregistrées.

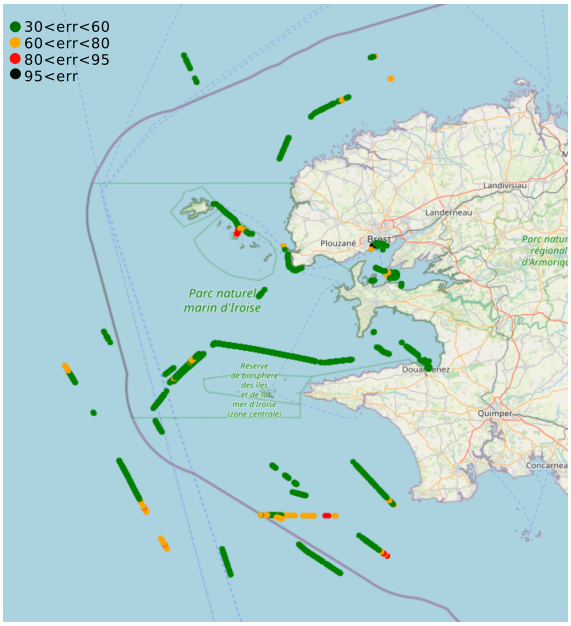

FIGURE 2.13 – Représentation des pourcentages d'alertes pour $H_{0,RI}$

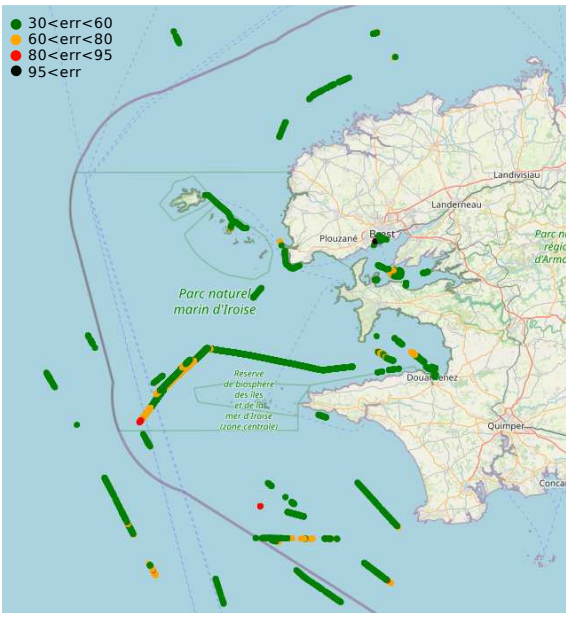

FIGURE 2.14 – Représentation des pourcentages d'alertes pour H_{TS}

2.7 Discussion

Bien que nous ayons montré l'utilité de la stratégie 2 pour détecter les faux messages et les bateaux fantômes, cette stratégie peut être contournée dans certaines conditions. En effet, si de fausses trajectoires sont injectées directement dans un transpondeur AIS, via l'entrée GPS par exemple, pour être émises sur le réseau, les messages transmis respecteront le mode d'accès TDMA, et la stratégie 2 ne détectera aucune falsification. Pour détecter une falsification dans un tel cas de figure il faut utiliser le fait que l'émetteur AIS est forcément non co-localisé avec la position du navire fantôme. Dès lors, la trajectoire relative entre le récepteur AIS, utilisé pour détecter les falsifications, et le navire fantôme, d'une part, et la trajectoire relative entre le récepteur AIS et l'émetteur réel, d'autre part, sont notablement différentes et par là même induisent des différences dans les variations de niveau, de Doppler ou d'azimut d'émission (goniométrie). Ces incohérences constituent des perspectives de travaux futurs en la matière pour détecter ce type de falsification. Néanmoins, les messages falsifiés peuvent aussi être émis par radio logicielle, comme dans [10], et alors, ils doivent respecter le mode d'accès TDMA pour ne pas être détectés par la stratégie mise en place. Or, cela arrive peu souvent, car il est difficile d'implémenter ce mode d'accès. Par ailleurs, d'autres techniques seront appliquées conjointement à cette stratégie, ce qui permettra d'augmenter sa robustesse comme nous allons le montrer dans le chapitre 4. Pour finir, afin d'améliorer la séparation entre fausses et vraies alarmes, des algorithmes issus de la théorie de la détection pourraient être utilisés. Justement, pour faciliter leur intégration à notre stratégie, nous mettons en libre accès son code Matlab [93].

2.8 Conclusion

Dans ce chapitre, nous avons proposé une stratégie qui permet de détecter les faux messages et les bateaux fantômes en contrôlant le respect, par les bateaux, du mode d'accès TDMA défini par la norme de l'AIS. Cette stratégie vérifie, que chaque bateau, d'une part, émet ses messages à une période qui est cohérente avec sa vitesse, et d'autre part, applique le procédé de réservation TDMA pour pré-annoncer les TS qu'il utilisera pour transmettre ses prochains messages. En raison de déficiences techniques ou de l'impact néfaste des conditions environnementales sur les communications, de nombreux messages n'arrivent pas à destination, ce qui implique un nombre élevé d'alertes. Ainsi, un premier tri calculant le pourcentage d'erreur a été proposé, et permet de différencier les bateaux suspects des autres. L'application de cette stratégie, à des données réelles, dont certaines ont été reconnues par l'OTAN comme étant falsifiées, a montré que ce tri était pertinent pour détecter les falsifications de messages AIS et les bateaux fantômes.

Chapitre 3

Détection des falsifications d'identité

3.1 Introduction

Les navires utilisant l'AIS sont authentifiés par leur identifiant MMSI qu'ils transmettent dans leurs messages. Cependant, ce numéro peut facilement être falsifié et remplacé par un numéro MMSI associé ou non à un autre bateau sur le réseau AIS. L'intérêt de cette action est de passer inaperçu lors d'une action frauduleuse ou d'usurper l'identité d'un autre navire. La stratégie présentée dans ce chapitre cherche à détecter ce type de falsification. Pour cela, chaque bateau est identifié par une signature radiométrique, caractérisant une imperfection matérielle de son transpondeur qui est unique. Ainsi, si un bateau falsifie son numéro MMSI, la signature radiométrique de son transpondeur restera la même et permettra de l'identifier et de détecter cette falsification. La signature radiométrique utilisée correspond à l'offset sur la fréquence porteuse et est extraite à partir du signal AIS en bande de base. Ainsi, dans ce chapitre, en plus des informations contenues dans les messages, nous considérons certaines caractéristiques des signaux AIS.

3.2 État de l'art

L'approche traditionnelle pour lutter contre les falsifications d'identité consiste à appliquer des méthodes cryptographiques permettant de chiffrer les messages. Ce type d'approche a été étudié et plusieurs solutions ont été proposées pour être appliquées aux messages AIS [9, 132]. Cependant, ces méthodes nécessitent certaines modifications matérielles et logicielles de l'AIS existant comme nous l'avons expliqué dans l'Introduction lorsque nous présentions l'état de l'art.

Un autre type d'approche consiste à extraire des signatures radiométriques des signaux émis par les émetteurs pour les identifier matériellement [164]. Cette approche a d'abord été développée par l'armée américaine dans les années 60 pendant la guerre du Vietnam pour différentier les radars amis des radars ennemis [147]. Ces signatures peuvent concerner les caractéristiques radiofréquences de l'émetteur [23] ou les caractéristiques du canal de communication entre l'émetteur et le récepteur [163]. Les caractéristiques du canal

peuvent être l'indicateur de la force du signal reçu [165], la densité spectrale de puissance du signal reçu [151], la réponse impulsionnelle du canal de communication [88] et sa réponse fréquentielle [163]. L'identification basée sur ce type de caractéristiques est efficace si ces caractéristiques sont stables dans le temps [169]. Dans notre application, l'émetteur et le récepteur sont en mouvement, et les conditions environnementales changent durant les communications. C'est pourquoi, pour identifier les émetteurs, nous n'utilisons pas les caractéristiques du canal de communication.

Parmi les caractéristiques radiofréquences de l'émetteur, appelées aussi signatures radiométriques, certaines sont calculées durant le régime transitoire du signal (coefficients d'ondelettes, temps de montée, temps de descente, temps avant et après modulation) et d'autres durant son régime permanent (bruit de phase, offset sur la fréquence porteuse, déséquilibre I/Q ...) [164, 23, 58]. Cependant, les signatures radiométriques calculées durant le régime transitoire sont difficile à extraire et peu discriminantes car calculées durant un intervalle de temps très court [164, 23]. Par exemple, l'article [123], qui est le seul utilisant des signatures radiométriques pour identifier des transpondeurs AIS, affirme que l'identification utilisant ce type de signature ne permet pas de différencier tous les transpondeurs. C'est pourquoi, nous considérons des signatures calculées durant le régime permanent du signal émis pour détecter les falsifications d'identité.

Pour le moment, aucune signature extraite durant le régime permanent du signal n'est utilisée pour identifier les transpondeurs AIS. Parmi ce type de signatures, l'offset sur la fréquence porteuse, appelé CFO en anglais, présente des performances d'identification intéressantes [136]. Il caractérise les imperfections des oscillateurs en émission et en réception qui ont pour conséquence que la fréquence porteuse n'est pas exactement à 161,975 MHz ou 162,025 MHz et qu'une composante fréquentielle parasite apparaît dans le signal en bande de base. Dans certaines applications, le CFO peut dériver dans le temps, à cause des conditions environnementales (température, tension d'alimentation, effet Doppler) [172, 148]. Cette dérive doit être considérée avec attention, car elle peut fortement détériorer les performances d'identification des émetteurs [136]. Hou et al. propose une solution en pistant cette dérive grâce à un KF (Kalman filter) [58]. Dans cet article, le CFO est modélisé par un processus aléatoire auto-régressif d'ordre 1 suivant un processus stationnaire. A chaque itération du filtre, le coefficient d'auto-corrélation est calculé par un calcul complexe de corrélation. Or, le CFO peut ne pas pouvoir être considéré comme suivant un processus stationnaire pendant le suivi [136, 78]. Sa valeur moyenne et son écart-type peuvent varier au cours du temps avec les variations des conditions environnementales et notamment celles concernant la température comme c'est le cas dans notre application [172, 136]. C'est pourquoi, nous proposons un autre modèle, plus simple, pour suivre avec un KF les variations de CFO. De plus, notre KF utilise des matrices de covariance adaptatives pour rendre notre stratégie robuste aux variations de bruits et de dérives des mesures de CFO qui dépendent du transpondeur pisté. En effet,

la qualité des oscillateurs et les conditions environnementales ne sont pas les mêmes pour tous les transpondeurs. À notre connaissance, c'est la première fois dans la littérature que le CFO est suivi par un tel KF.

3.3 Pistage de CFO des transpondeurs AIS

Puisque le CFO peut dériver au cours du temps, son évolution est pistée par un filtre de Kalman. L'équation dynamique du filtre, le calcul des bruits de modèle et de mesure et l'équation d'initialisation appliquée sont présentées dans cette partie.

3.3.1 Caractéristiques du CFO

Pour rappel, les signaux AIS sont transmis, après modulation GMSK (Gaussian minimum-shift keying), en VHF sur deux fréquences porteuses (161,975 MHz et 162,025 MHz). Le cahier des charges de l'AIS accepte des erreurs de ± 500 Hz (± 3 ppm) sur les fréquences porteuses en raison d'imperfections matérielles que l'on appelle CFO [133]. Ces erreurs sont dues à la température (principalement), à la tension d'alimentation et aux facteurs de vieillissement [172]. Lors de la transposition en fréquence du signal AIS reçu, au CFO de l'émetteur est soustrait celui du récepteur ce qui induit une composante fréquentielle parasite dans le signal en bande de base. Cette composante fréquentielle est le CFO relatif du transpondeur de l'émetteur par rapport à celui du récepteur que l'on appellera, par abus de langage, CFO de l'émetteur. Le CFO varie d'un transpondeur à l'autre, car chaque transpondeur a des imperfections matérielles uniques. Ainsi, en connaissant le CFO, le transpondeur de chaque bateau sera identifié.

3.3.2 Modèle dynamique appliqué

En fonction de l'oscillateur utilisé le CFO peut varier au cours du temps. Cette variation est appelée dérive et est due principalement aux variations de température et à l'effet Doppler. Par exemple, l'oscillateur utilisé par l'USRP (Universal software radio peripheral) e310 [22] avec lequel nous démodulons les signaux AIS reçus est un oscillateur TCXO (Temperature compensated X (crystal) oscillator). Pour ce type d'oscillateur, la dérive peut varier entre ± 0.7 ppm $= \pm 113$ Hz à cause des variations de température [78], ce qui est susceptible de se produire durant nos expérimentations car elles durent plusieurs heures d'affilées. À titre de comparaison, la dérive causée par l'effet Doppler est beaucoup plus faible : elle vaut ± 10 Hz si le bateau se déplace à 40 kn.

La dérive du CFO n'est pas constante au court du temps et peut augmenter ou diminuer après plusieurs minutes de mesures. Néanmoins, on peut supposer que durant un court laps de temps, de quelques minutes, elle est presque constante. Ainsi, un polynôme du premier ordre est appliqué pour modéliser l'évolution du CFO, et considère la première

dérivée temporelle du CFO (la dérive) comme constante. Il s'agit du même modèle que celui à vitesse constante (CV) qui a été appliqué pour suivre la position. Les mêmes équations, présentées dans la partie (1.3.4) peuvent donc être utilisées. Le filtre de Kalman en pistant le CFO s'ajustera automatiquement à la dérive du CFO si les bruits de modèle et de mesure du filtre de Kalman sont bien réglés.

Soit $n \in \mathbb{N}$ l'indice de temps discret, l'équation dynamique d'évolution du CFO a pour expression :

$$X_{n+1} = F_n X_n + V_n \tag{3.1}$$

où

$$X_n = \begin{pmatrix} x_n \\ \dot{x}_n \end{pmatrix} ; \; F_n = \begin{pmatrix} 1 & \Delta T_n \\ 0 & 1 \end{pmatrix}$$

X_n est le vecteur d'état contenant le CFO et sa dérivée première par rapport au temps.

Comme pour le suivi de position, le modèle de bruit utilisé est une discrétisation du modèle d'accélération à bruit blanc continu (CWNA (Continuous white noise acceleration)) où :

$$\mathbb{E}[V_n] = 0 \tag{3.2}$$

$$\mathbb{E}[V_n V_k^T] = Q_n \delta_{nk} = \begin{pmatrix} \frac{\Delta T_n^3}{3} & \frac{\Delta T_n^2}{2} \\ \frac{\Delta T_n^2}{2} & \Delta T_n \end{pmatrix} \tilde{q} \delta_{nk} \tag{3.3}$$

avec $(n, k) \in \mathbb{N}^2$, $\mathbb{E}[.]$ est l'opérateur d'espérance et δ est le symbole de Kronecker.

3.3.3 Calcul du CFO

Pour chaque message reçu, le CFO est calculé à partir de son signal AIS en bande de base, qui est reçu en quadrature (I/Q) et a pour expression : $x(t) = I(t) + jQ(t)$. Le calcul applique au produit défini par l'équation (3.4) une transformée de Fourier (FFT) et sélectionne la fréquence pour laquelle la norme de la FFT est maximale :

$$r(t) \times s^*(t) = K(t)||s(t)||^2 e^{j2\pi\Delta f t} + n'(t) \tag{3.4}$$

où $r(t)$ est la séquence d'apprentissage reçue : $r(t) = K(t)s(t)e^{j2\pi\Delta f t} + n(t)$, $K(t)$ l'atténuation du signal, $s(t)$ la séquence d'apprentissage de référence avec $||s(t)|| = 1$, Δf le CFO, $n(t)$ le bruit du signal reçu, s^* le conjugué de $s(t)$ et $n'(t)$ un bruit additif.

Les signaux sont échantillonnés à la fréquence $F_s = 192$ kHz. La séquence d'apprentissage $r(t)$ est extraite du signal en bande de base $x(t)$ grâce à un calcul de corrélation avec la séquence d'apprentissage de référence $s(t)$. L'atténuation $K(t)$ est supposée constante, car le signal AIS est à bande étroite, c'est-à-dire que la largeur de bande du signal est inférieure à la bande de cohérence du canal. Par ailleurs, les signaux sont transmis dans

un environnement marin, ce qui limite leur nombre de réflexions.

La FFT est appliquée à 192 000 points pour avoir une précision fréquentielle de 1 Hz ($\frac{F_s}{N}$). Pour ce nombre de points, l'écart-type de l'erreur d'estimation du CFO varie avec le SNR comme présenté sur la Figure 3.1. Pour connaître cette évolution, une simulation de Monte Carlo a été appliquée. Cette simulation, crée, pour chaque SNR (en dB) se trouvant dans l'intervalle $[0, 20]$, 500 séquences d'apprentissage $r(t)$ avec un offset en fréquence de $\Delta f = 400$ Hz et un bruit blanc $n(t)$ dont l'amplitude est réglée par la valeur du SNR testé. A partir de ce signal, nous appliquons la formule présentée plus haut pour estimer Δf. Pour chaque SNR testé, avec les 500 valeurs de Δf estimées, nous calculons la moyenne et l'écart-type de l'erreur d'estimation. Ces valeurs sont présentées sur les Figures 3.2 et 3.1.

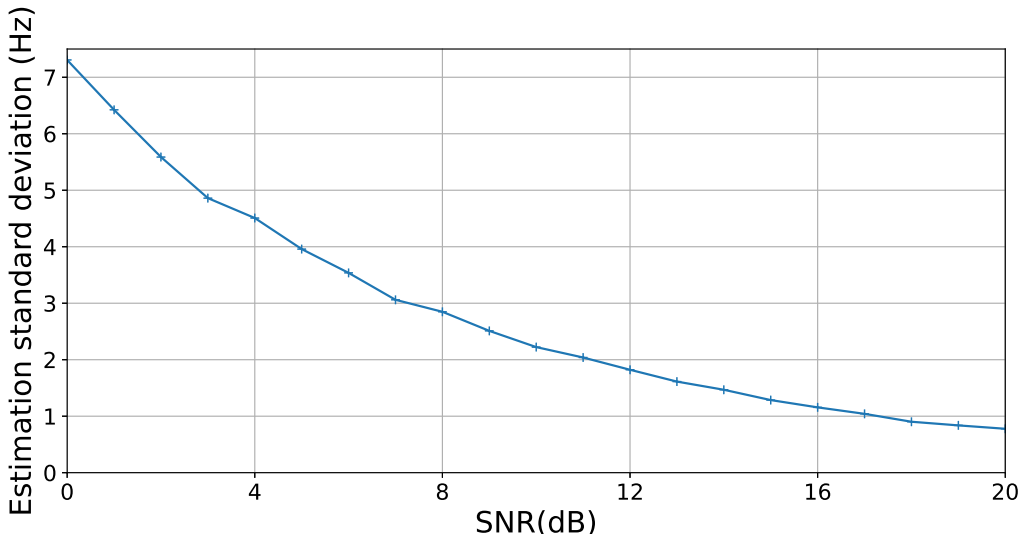

FIGURE 3.1 – Évolution de l'écart-type de l'erreur d'estimation du CFO en fonction du SNR pour une FFT fait sur 192 000 points.

L'équation d'observation du CFO a la forme suivante :

$$Z_n = HX_n + w_n \tag{3.5}$$

Avec $H = \begin{pmatrix} 1 & 0 \end{pmatrix}$ la matrice d'observation et w_n le bruit d'observation qui est un bruit blanc gaussien comme le montre la Figure 3.3 qui présente l'histogramme de l'erreur d'estimation pour tous les SNR. Ainsi la matrice de covariance du bruit de mesure a pour expression :

$$\mathbb{E}[w_n w_k] = R\delta_{nk} = \sigma_w^2 \delta_{nk} \tag{3.6}$$

Les bruits d'observation et de modèle sont indépendants :

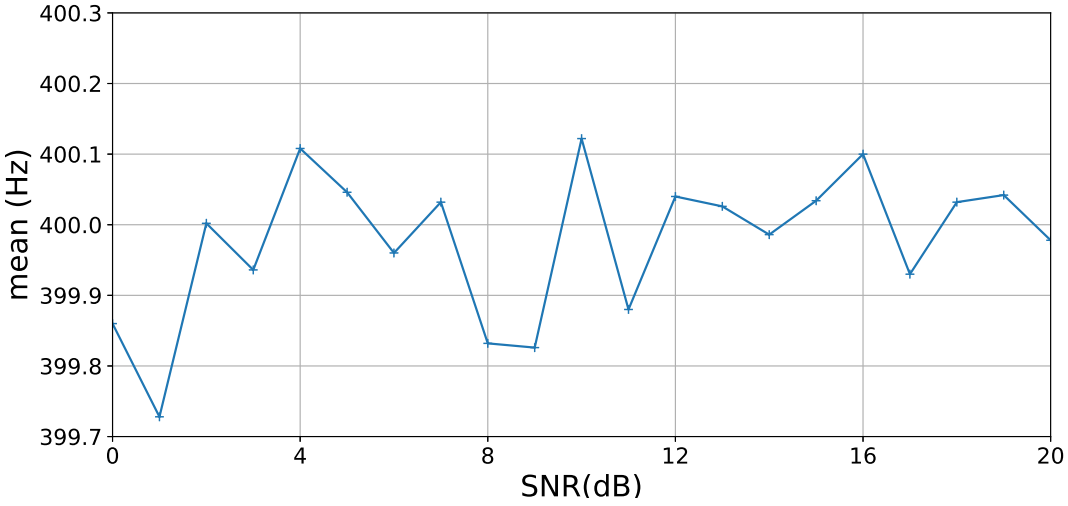

FIGURE 3.2 – Évolution de la moyenne de l'erreur d'estimation du CFO en fonction du SNR pour une FFT fait sur 192 000 points.

3.3.4 Initialisation et modélisation du bruit

Comme pour la position, l'initialisation est effectuée en appliquant la méthode de différentiation à deux points [13].

$$\widehat{X}_{1|1} = \begin{pmatrix} Z_1 \\ \\ \frac{Z_1 - Z_0}{\Delta T_1} \end{pmatrix} ; \widehat{P}_{1|1} = \begin{pmatrix} R & \frac{R}{\Delta T_1} \\ \\ \frac{R}{\Delta T_1} & \frac{2R}{\Delta T_1^2} \end{pmatrix} \tag{3.7}$$

Le bruit d'observation dépend du SNR des signaux reçus comme expliqué dans la partie précédente. Quant au bruit de modèle, il dépend de la variation de température et de l'effet Doppler qui impactent différemment chaque transpondeur. C'est pourquoi, pour chaque CFO suivi, les matrices de covariance du bruit d'observation et du bruit de modèle, sont ajustées automatiquement au cours du temps, grâce aux équations (3.8) et (3.9) présentées ci-dessous et introduites dans [5].

$$R_n = \alpha_R R_{n-1} + (1 - \alpha_R)(\epsilon_{n-1}\epsilon_{n-1}^T + H\widehat{P}_{n|n-1}H^T) \tag{3.8}$$

$$Q_n = \alpha_Q Q_{n-1} + (1 - \alpha_Q)(K_{n-1}\tilde{Z}_{n-1}\tilde{Z}_{n-1}^T K_{n-1}^T) \tag{3.9}$$

où α_R et α_Q sont des facteurs d'oubli ($\alpha_R = 0{,}95$ et $\alpha_R = 0{,}55$), ϵ_n est le résidu calculé par (3.10) et $\tilde{Z}_n$ l'innovation calculée par le filtre de Kalman (1.21).

$$\epsilon_n = Z_n - H\widehat{X}_{n|n} \tag{3.10}$$

Q_n dépend de ΔT_n qui varie au cours du temps en fonction de la période d'émission des messages. C'est pourquoi, nous considérons $\tilde{q}_n^2$, qui lui est indépendant de ΔT_n, et

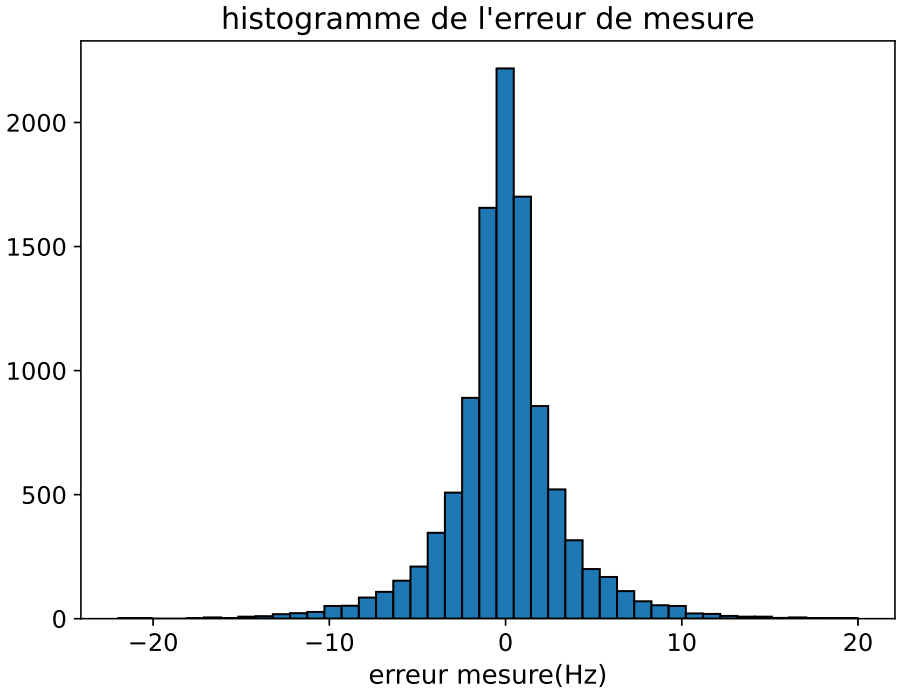

FIGURE 3.3 – Histogramme de l'erreur d'estimation du CFO

l'équation (3.11) à la place de (3.9) ;

$$\tilde{q}_n = \alpha_Q \tilde{q}_{n-1} + (1-\alpha_Q)\frac{(K_{n-1}\tilde{Z}_{n-1}\tilde{Z}_{n-1}^T K_{n-1}^T)(2,2)}{\Delta T_{n-1}} \tag{3.11}$$

$\tilde{q}_n$ permet ensuite de calculer Q_n

R_n est initialisé en fixant $R_1 = \sigma_w^2$ avec $\sigma_w = 7$ Hz. 7 Hz est la valeur maximale de l'écart-type sur l'erreur de mesure que l'on peut rencontrer comme le montre la Figure 3.1. Q_n est initialisée en considérant que la dérive maximale du CFO sur une période ΔT est de l'ordre de $Q_n(1,1) = \sqrt{\frac{\Delta T^3}{3}}\tilde{q}$. En observant les Figure 3.4 et 3.5, nous constatons qu'au maximum, pour $\Delta T = 1000$ s, nous avons une dérive de 20 Hz, donc $Q_n(1,1) = 20$ Hz.

3.4 Détection des falsifications d'identité

Pour détecter les falsification d'identité, l'innovation du CFO, calculée par le filtre de Kalman, est considérée par un test de conformité.

3.4.1 Test de conformité

De même que pour la position, un test du khi-deux est appliqué (3.13) pour détecter les mesures absurdes en considérant l'innovation $\tilde{Z}_n$ et sa variance S_n calculées par le filtre de Kalman. $\tilde{Z}_n$ suit une loi normale, car les bruits de modèle et d'observation sont

gaussiens. L'hypothèse $H_{0,cfo}$ vraie signifie que la mesure testée provient du transpondeur suivi et $H_{0,cfo}$ faux signifie que la mesure testée ne provient pas du transpondeur suivi.

$$l_n = \tilde{Z}_n^T S_n^{-1} \tilde{Z}_n \tag{3.12}$$

$$T(Z_n) = \begin{cases} H_{0,cfo} : & \tilde{Z}_n \text{ suit une distribution normale de moyenne nulle et de variance } S_n \\ 1 & \text{si } l_n \leq \gamma \ Z_n \text{ validée} \\ 0 & \text{sinon} \quad Z_n \text{ rejetée} \end{cases} \tag{3.13}$$

Nous supposons, comme dans [58], que l'ensemble des valeurs de CFO suit une distribution normale centrée avec un écart-type $\sigma = 166$ Hz. L'écart-type est fixé à 166 Hz parce que les valeurs de CFO varient entre ± 500 Hz [133] et nous fixons $3\sigma = 500$ Hz. Ainsi, la différence entre le CFO du transpondeur usurpé et le CFO du transpondeur qui usurpe suit également une distribution normale centrée (addition de deux distributions normales) : $p_{\tilde{Z}_n|H_{1,cfo}} \sim \mathcal{N}(0, \sqrt{2} \times 166)$. Pour avoir une probabilité de fausse alarme tel que $\alpha = 0{,}01$, nous fixons le seuil $\gamma = 6{,}64$. Le rejet d'une mesure peut signifier que l'identité du transpondeur a été usurpée.

3.4.2 Algorithme

L'algorithme de la stratégie 3 est décrit ci-dessous. Il est présenté en régime permanent à l'instant discret n. Il considère, en entrée, le temps d'arrivée du message ($Toa(n)$) et la mesure du CFO (Z_n) dont la méthode de calcul a été présentée en (3.3.3). En sortie, il renvoie un message d'alerte indiquant si l'hypothèse $H_{0,cfo}$ a été acceptée ou non. Plusieurs acronymes sont utilisés dans la description de l'algorithme faisant référence aux différentes étapes du filtre de Kalman :

— **Kp** représente l'étape de prédiction du filtre de Kalman ;
— **Ke** représente l'étape d'estimation du filtre de Kalman ;

Par ailleurs, lorsque cinq erreurs d'affilée sont rencontrées, le filtre est recalé sur les dernières mesures avec l'équation suivante (3.14).

$$\widehat{X}_{n|n} = \begin{pmatrix} Z_n \\ \\ \frac{Zn - Z_{n-1}}{\Delta T(n)} \end{pmatrix} ; \ \widehat{P}_{n|n} = \begin{pmatrix} R & \frac{R}{\Delta T_n} \\ \\ \frac{R}{\Delta T_n} & \frac{2R}{\Delta T_n^2} \end{pmatrix} \tag{3.14}$$

3.5 Observation du CFO sur des signaux réels

Des signaux AIS ont été enregistrés dans la rade de Brest durant plus de huit heures pour observer l'évolution du CFO au cours du temps. Ces observations ont été utilisées

Algorithm 2 Détection des falsifications d'identité

Data: $Toa(n), Z_n$
Result: $H_{0,cfo}$
At each timestep n :
$\Delta T(n) = Toa(n) - Toa(n-1)$
$\epsilon_{n-1} = Z_{n-1} - H\widehat{X}_{n-1|n-1}$
$\tilde{q}_n = \alpha_Q \tilde{q}_{n-1} + (1-\alpha_Q)\frac{(K_{n-1}\tilde{Z}_{n-1}\tilde{Z}_{n-1}^T K_{n-1}^T)(2,2)}{\Delta T_{n-1}}$
$Q_n = \begin{pmatrix} \frac{\Delta T_n^3}{3} & \frac{\Delta T_n^2}{2} \\ \frac{\Delta T_n^2}{2} & \Delta T_n \end{pmatrix} \tilde{q}$
$\widehat{X}_{n|n-1}, \widehat{P}_{n|n-1} = \mathbf{Kp}(\Delta T, \widehat{X}_{n-1|n-1}, \widehat{P}_{n-1|n-1}, Q_n)$
$R_n = \alpha_R R_{n-1} + (1-\alpha_R)(\epsilon_{n-1}\epsilon_{n-1}^T + H\widehat{P}_{n|n-1}H^T)$
$\tilde{Z}_n = Z_n - H\widehat{X}_{n|n-1}$
$S_n = R_n + H\widehat{P}_{n|n-1}H^T$
$l_n = \tilde{Z}_n^T S_n^{-1} \tilde{Z}_n$
if $l_n < \chi_\alpha$ **then**
  $\widehat{X}_{n|n}, \widehat{P}_{n|n} = \mathbf{Ke}(Z_n, \widehat{X}_{n|n-1}, \widehat{P}_{n|n-1})$
  $NbErr = 0$
else
  $\widehat{X}_{n|n} = \widehat{X}_{n|n-1}$; $\widehat{P}_{n|n} = \widehat{P}_{n|n-1}$
  $NbErr = NbErr + 1$
end
if $NbErr = 5$ **then**
  $\widehat{X}_{n|n} = \left(Z_n; \frac{Z_n - Z_{n-1}}{\Delta T_n}\right)$
  $\widehat{P}_{n|n} = \begin{pmatrix} R & \frac{R}{\Delta T_n} \\ \frac{R}{\Delta T_n} & \frac{2R}{\Delta T_n^2} \end{pmatrix}$
  $NbErr = 0$
end
if $l_{CFO}(n) > \chi_\alpha$ **then**
  $H_{0,cfo} = 0$
else
  $H_{0,cfo} = 1$
end

pour régler le filtre de Kalman et permettent de donner une première idée de l'efficacité de l'algorithme détectant les falsifications d'identité.

3.5.1 Conditions d'enregistrement

Parce qu'il n'y a pas de signaux AIS publics, deux enregistrements de signaux AIS ont été réalisés pour récupérer des données et tester notre stratégie. Les deux enregistrements ont duré respectivement 292 min et 225 min ; un USRP e310 [22] utilisant l'interface Gnuradio a servi pour recueillir les signaux. Le filtre de réception utilisé est centré sur 162 MHz et la période d'échantillonnage du signal en bande de base est de 192 kHz. L'antenne est d'abord connectée à un amplificateur faible bruit [110], puis à l'USRP pour améliorer le SNR des signaux reçus. Pour réduire la dérive du CFO, l'USRP a été démarré deux heures avant le début des deux enregistrements afin de le placer dans des conditions stables, notamment en ce qui concerne sa température de fonctionnement. Les signaux AIS sont mis à disposition sur GitHub [91] afin de permettre à d'autres chercheurs ou ingénieurs d'avoir des données réelles pour développer leurs algorithmes.

3.5.2 Observations

Les signaux enregistrés contiennent 8455 messages envoyés par 31 navires. Pour chaque message, le CFO est extrait et nous présentons sur les Figures 3.4 et 3.5 l'évolution du CFO au cours du temps pour chaque bateau. L'observation des CFO montrent qu'il s'agit d'une signature radiométrique discriminante permettant d'identifier chaque bateau (représentés par leur numéro MMSI). Leurs valeurs sont comprises entre ± 500 Hz, ce qui est conforme à la norme AIS [133]. Une dérive du CFO apparaît sur les courbes en plus d'un bruit de mesure dont l'écart-type dépend du bateau suivi. Ces deux dernières remarques justifient l'utilisation d'un KF pistant le CFO et ajustant, dans le temps, sa modélisation du bruit de mesure et de modèle.

Par ailleurs, nous observons, pour chaque transpondeur, que le CFO reste confiné, au cours de l'enregistrement, dans un intervalle de largeur 200 Hz ce qui est cohérent avec la dérive en fréquence de l'oscillateur TCXO utilisé par notre récepteur USRP [78]. Ainsi, la valeur maximale d'innovation calculée par notre KF ne dépassera jamais 200 Hz en valeur absolue. Nous fixons donc, pour chaque test sur le CFO, la valeur maximale du seuil ($\sqrt{S_n \times \gamma}$) à 200 Hz, ce qui nous permet de diminuer la valeur β. En effet, lorsque plusieurs messages consécutifs ne sont pas reçus, la matrice de covariance S_n augmente beaucoup et $\sqrt{S_n \times \gamma}$ peut dépasser 200 Hz.

Pour ce seuil maximal, la probabilité de non détection β est calculé en utilisant la fonction de distribution cumulative d'une distribution normale. En considérant la loi $p_{\tilde{Z}_n|H_1}$ que nous avons rappelée dans la partie (3.4.1), nous obtenons $\beta = erf(\frac{200}{\sqrt{2}\times 166 \times \sqrt{2}}) = 0{,}60$. Cette valeur représente la valeur maximale de β que nous obtenons avec notre test. Cette

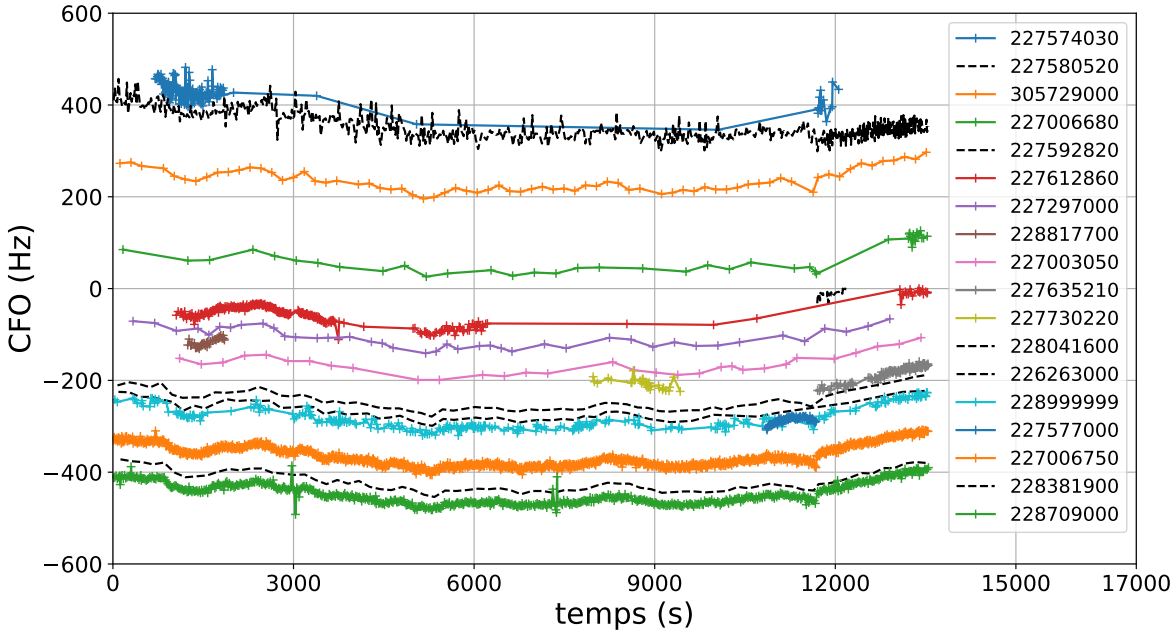

FIGURE 3.4 – Évolution du CFO dans le temps pour plusieurs transpondeurs AIS (premier enregistrement)

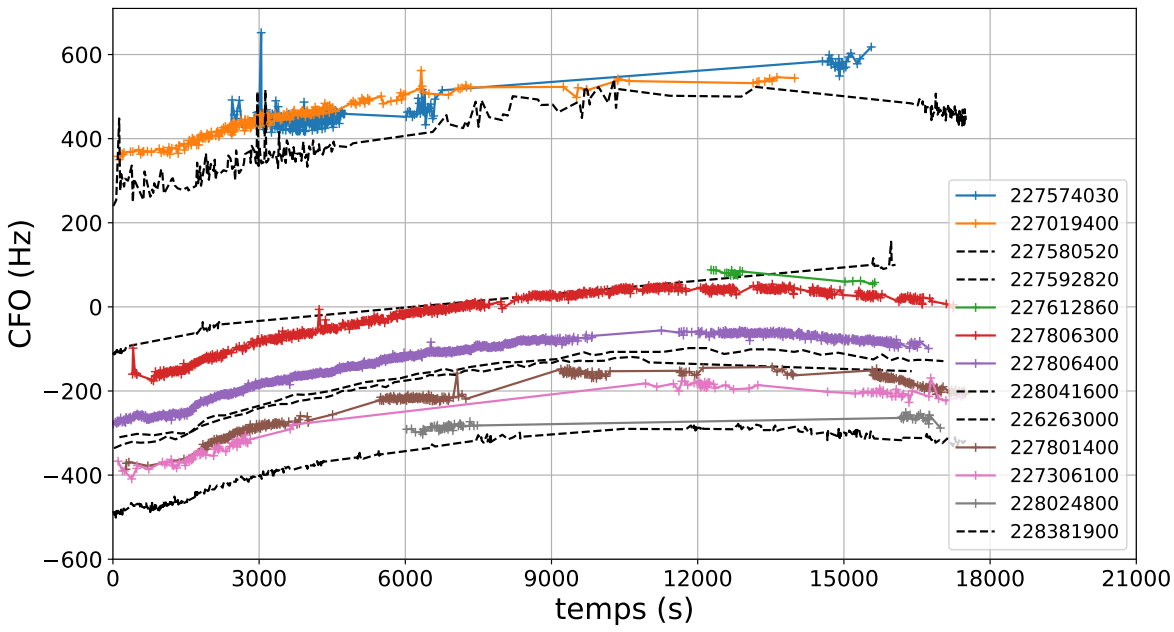

FIGURE 3.5 – Évolution du CFO dans le temps pour plusieurs transpondeurs AIS (deuxième enregistrement).

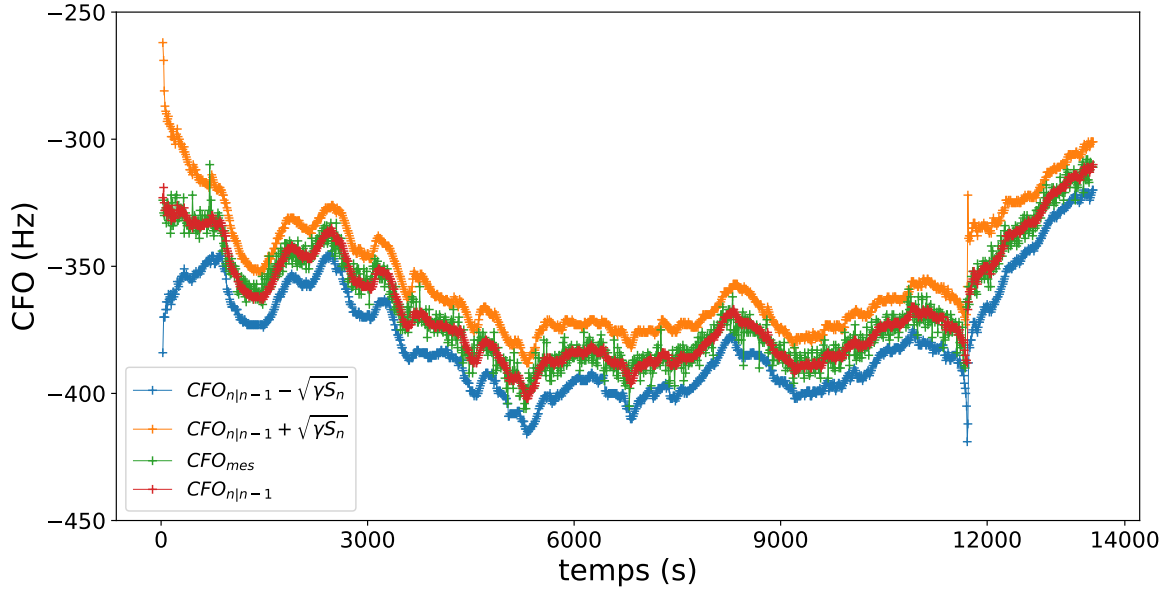

FIGURE 3.6 – Évolution du CFO du transpondeur d'un des bateaux pisté, en plus du CFO estimé par le KF et du seuil haut et bas du test de conformité

valeur est trop élevée pour rendre notre test puissant ($\beta < 0{,}05$). Néanmoins, il s'agit d'une valeur maximale et la plupart du temps, le seuil est inférieur à 200 Hz. En effet, comme présenté sur la Figure 3.6, qui montre l'évolution du CFO pour l'un des transpondeurs suivis, $\sqrt{S_n \times \gamma}$ est d'environ 15 Hz ($\beta = 0{,}0509$). La valeur de la probabilité de non détection devient acceptable, mais le test n'est toujours pas puissant car $\beta > 0{,}05$. C'est pourquoi, afin de diminuer la probabilité de non détection, cette stratégie est appliquée conjointement avec la stratégie pistant la position des transpondeurs. C'est ce que nous allons présenter dans le chapitre suivant.

3.6 Conclusion

Dans ce chapitre, une stratégie permettant de détecter les falsifications d'identité a été proposée. Cette stratégie extrait une signature radiométrique des signaux reçus pour chaque transpondeur afin de les caractériser matériellement et de les identifier indépendamment de l'identité (numéro MMSI) qu'ils transmettent dans leurs messages. La signature utilisée est l'offset en fréquence des porteuses appelé CFO. Puisque ce CFO dérive au cours du temps, un filtre de Kalman est utilisé pour suivre son évolution. Ce filtre utilise des bruits de modèle et de mesure adaptatifs afin de s'adapter automatiquement au niveau de bruit et à la dérive des oscillateurs des transpondeurs, et aux variations des conditions environnementales dans lesquelles les mesures sont extraites. La probabilité de fausse alarme est fixée à $\alpha = 1$ %, et, pour cette valeur, la probabilité de non détection peut atteindre, au maximum 60 %. Bien que l'expérimentation nous ait montré que cette

probabilité pouvait atteindre des valeurs plus raisonnables (5,1 %), ces valeurs restent élevées et empêchent notre test d'être puissant. C'est pourquoi, dans le prochain chapitre, ce test permettant de détecter les falsifications d'identité considère en plus, afin de diminuer cette probabilité de non détection, la position et les TS réservés.

Chapitre 4

Application conjointe des stratégies

4.1 Introduction

Dans ce chapitre, nous présentons une stratégie globale appliquant conjointement les trois stratégies que nous avons présentées dans les précédents chapitres. Cette application conjointe permet de compenser les faiblesses des stratégies prises individuellement et d'améliorer leur efficacité. Nous montrerons, en particulier, la diminution des probabilités de fausse alarme et de non détection qui en résulte pour le test détectant les falsifications d'identité.

4.2 Stratégie globale

Les trois stratégies sont appliquées conjointement pour détecter les falsifications de message AIS, et forment ensemble une stratégie globale. Les conditions d'application de cette stratégie et son architecture sont présentées dans cette partie.

4.2.1 Conditions d'application

Cette stratégie globale ne s'applique qu'aux messages de rapport de position (`ID`=1, 2 ou 3) transmis par les transpondeurs AIS de classe A, qui représentent 80 % des messages émis [81]. Ce type de message provient des stations mobiles et non pas des stations de base présentes sur la côte et utilisées pour surveiller le trafic maritime. Une description détaillée des informations contenues dans ce type de message est présentée en annexe (D).

La stratégie globale est intégrée à un récepteur AIS implémenté sur un FPGA. Ce système a vocation à être utilisé sur la surface terrestre et non pas dans des satellites. Un satellite ne reste pas fixé au dessus d'un point de la surface de la Terre, mais se déplace continuellement autour du globe. Ainsi, le temps séparant la réception de deux messages consécutifs d'un même bateau est long (entre 500 s et 1500 s), comme nous l'avons constaté pour les données provenant de l'OTAN dans la partie 2.6.2 du chapitre 2. Pour de tels délais, nos stratégies 1 et 3, basées sur le pistage de cible par application du filtre de Kalman et IMM, deviennent moins sensibles aux falsifications et la stratégie

2 devient inapplicable. La portée des émetteurs AIS à la surface de la Terre est de 25 à 40 km. Ainsi, seulement les informations transmises par les transpondeurs AIS se trouvant dans une zone dont le rayon est égale à cette distance seront contrôlées. Le nombre de transpondeurs contrôlés varie au cours du temps à mesure que des bateaux sortent ou entrent dans cette zone. Par exemple, sur la Figure 4.1, le nombre de bateaux contrôlés est de trois alors qu'il est de quatre sur la Figure 4.2. Sur ces deux figures le récepteur est placé sur la côte mais ce positionnement n'est pas obligatoire, un récepteur peut très bien être placé sur un bateau.

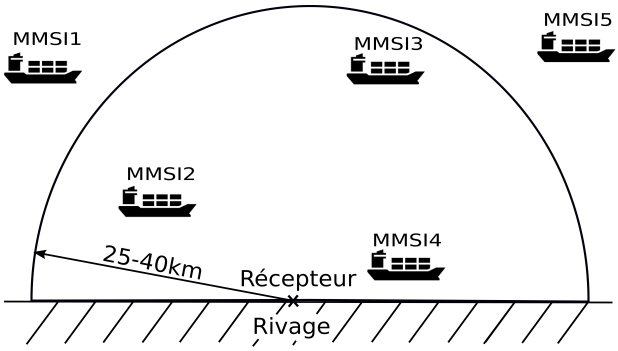

FIGURE 4.1 – Activité maritime à l'instant t0

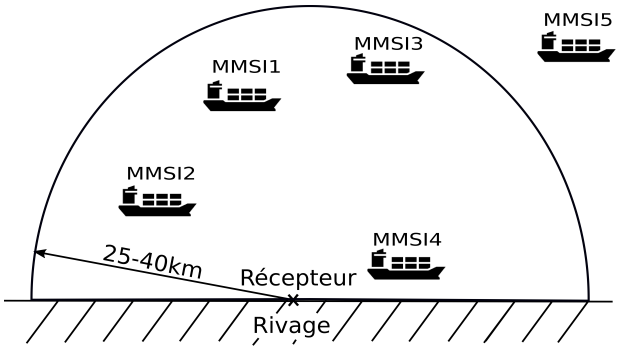

FIGURE 4.2 – Activité maritime à l'instant t1

Chaque bateau situé dans la zone est identifié grâce au numéro MMSI qu'il transmet dans ses messages. Ce numéro peut être falsifié comme nous l'avons montré dans l'Introduction lorsque nous présentions les vulnérabilités de l'AIS. C'est pourquoi, l'identité de chaque bateau se trouvant dans la zone surveillée doit être vérifiée. Cette vérification s'effectue en deux étapes, une première étape vérifie l'identité de chaque bateau entrant dans la zone surveillée, et une seconde étape s'assure, en pistant chaque bateau, que son identité ne change pas. Ces deux vérifications sont effectuées respectivement par l'étape d'*Identification* et l'étape de *Vérification* introduites par [21] et expliquées en détails dans la prochaine partie (4.4).

4.2.2 Architecture

L'architecture de la stratégie globale est présentée sur le schéma de la Figure 4.3. La stratégie commence par analyser l'évolution de la puissance du signal pour détecter l'arrivée d'un message. Ensuite, le CFO est calculé et le signal est démodulé pour extraire les données contenues dans le message. Après, le type du message est contrôlé pour s'assurer qu'il s'agisse d'un report de position. La liste des transpondeurs (bateaux) pistés est ensuite mise à jour en supprimant tous les transpondeurs qui n'ont pas envoyé de messages depuis plus de 420 s. Cette durée de 420 s est la même que pour le Saab R5 Supreme qui est un des AIS de classe A vendu sur le marché. Cette mise à jour permet d'éliminer les transpondeurs ne se trouvant plus dans la zone surveillée. Après, l'étape d'*Identification* ou l'étape de *Vérification* est appliquée. L'étape d'*Identification* vérifie l'identité et permet de confirmer l'entrée d'un nouveau bateau dans la zone surveillée. Si c'est le cas, le transpondeur associé à ce bateau est ajouté à la liste des transpondeurs déjà pistés. Durant l'étape de *Vérification*, les trois stratégies sont appliquées conjointement, comme présenté sur la Figure 4.4, pour détecter les falsifications de position, de vitesse et d'identité, en plus des faux messages. Le code Matlab de la stratégie avec des données réelles est disponible au lien [92].

4.3 Amélioration des performances de la stratégie détectant les falsifications d'identité

En plus de leur CFO, les transpondeurs des bateaux peuvent être caractérisés par leur positions et les TS qu'ils ont réservés et utilisés. C'est pourquoi, les stratégies 1 et 2 sont aussi utilisées, en appui de la stratégie 3, pour détecter les falsifications d'identité.

4.3.1 Apports des stratégies 1 et 2

Pour rappel, la stratégie 1 détecte les falsifications de position et de vitesse contenues dans les messages reçus. Pour cela, la position des navires est pistée par un filtre IMM. Or, la position et la vitesse caractérisent un bateau. Par exemple, si un utilisateur falsifie son MMSI et le remplace par le MMSI d'un bateau déjà pisté, alors deux bateaux émettrons des messages avec un même MMSI mais des positions différentes. Cela se traduira par des sauts de position et des alarmes émises par la stratégie 1. La falsification d'identité sera donc détectée. La vitesse, contrairement à la position, n'est pas assez discriminante pour caractériser un bateau, d'autant plus que lorsque le bateau manœuvre son estimation est imprécise. C'est pourquoi, elle n'est pas prise en considération pour vérifier l'identité des bateaux.

La stratégie 2 contrôle, pour chaque bateau, le respect du mode d'accès TDMA. Elle

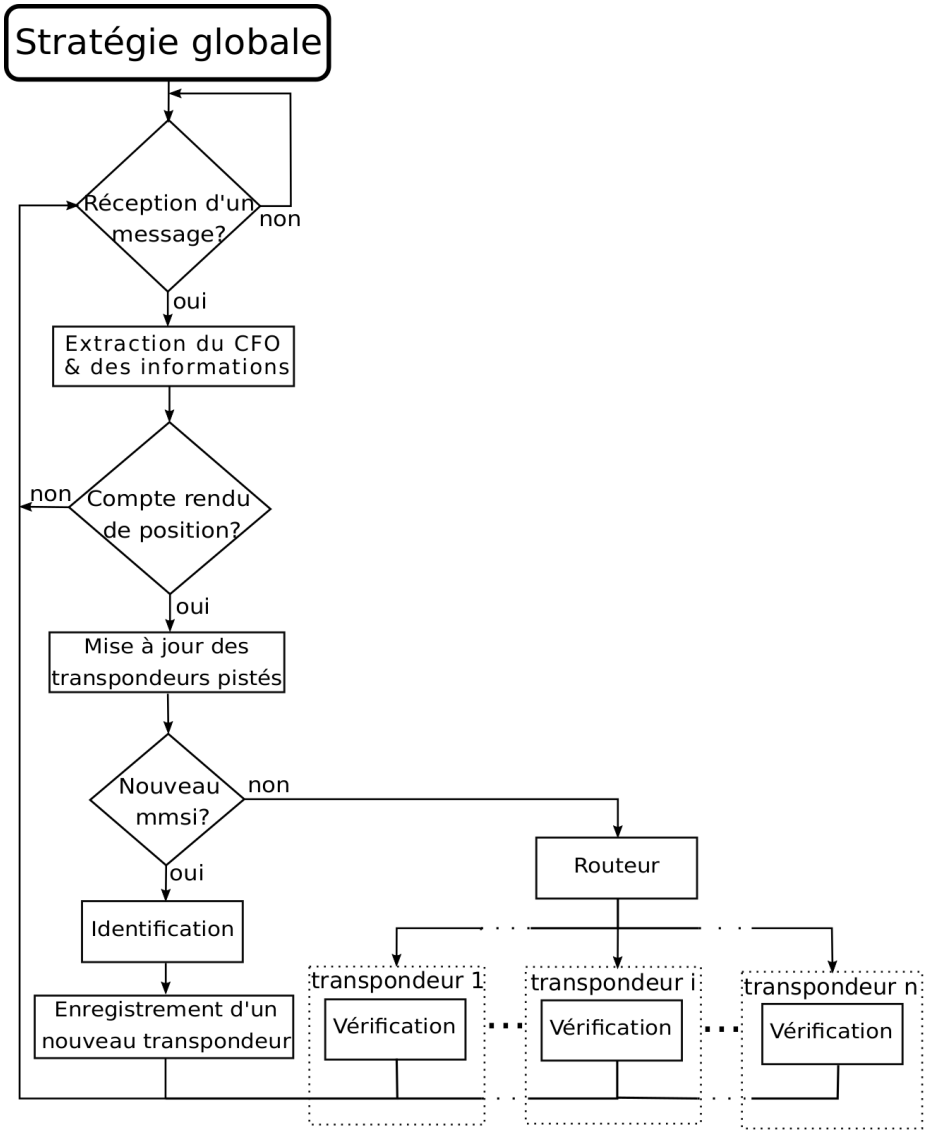

FIGURE 4.3 – Architecture de la stratégie globale

vérifie, en particulier, par l'application de l'Algorithme 2, que chaque bateau émet ses messages durant des TS qu'il a préalablement réservés. Ces TS réservés sont uniques pour chaque bateau et permettent de l'identifier. C'est pourquoi, la stratégie 2 peut aussi être utilisée pour détecter les falsifications d'identité. Par exemple, un bateau qui émettrait des messages durant des TS non réservés ou réservés par un bateau avec un autre numéro MMSI peut avoir falsifié son MMSI. Dans le premier cas, cette falsification a été effectuée depuis un bateau qui n'était pas suivi par la stratégie, et dans le deuxième cas, cette falsification a été effectuée depuis un bateau qui était déjà suivi par la stratégie.

Néanmoins, l'application de la stratégie 2 sur des données réelles dans le chapitre 2 a montré, que lorsque les conditions d'utilisation étaient mauvaises (haut SNR, antenne masquée par un obstacle, etc), beaucoup de messages étaient reçus sans avoir été réservés préalablement. Dans ce cas, il s'agit de fausses alarmes et l'alarme reçue n'indique pas une falsification d'identité. Toutefois, la stratégie 2 reste très utile, car elle permet d'infirmer

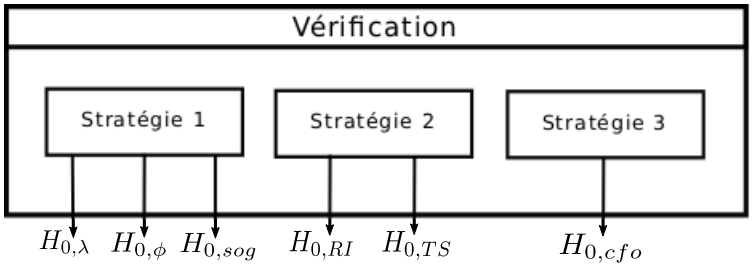

FIGURE 4.4 – Architecture du test d'identité appliqué durant l'étape de *Vérification*

ou de confirmer une falsification d'identité détectée. En effet, si un bateau est suspecté d'avoir falsifié son MMSI mais qu'il émet un message durant un TS réservé par le bateau associé à cet MMSI, alors l'alarme de falsification d'identité est rejetée : il s'agit d'une fausse alarme. Un bateau ne peut pas émettre durant le TS réservé par un autre bateau, car alors, les deux bateaux émettraient leur message en même temps, il y aurait des interférences entre leurs signaux et aucun message ne serait reçu. Par contre, si un bateau est suspecté d'avoir falsifié ou changé son MMSI, et qu'il émet des messages durant les TS réservés par le bateau identifié par son précédent MMSI, alors la falsification est confirmée. L'hypothèse $H_{0,TS}$ est donc utilisée pour détecter les falsifications d'identité.

Par ailleurs, la stratégie 2 contrôle aussi la période d'émission des messages des transpondeurs. Cependant, ce contrôle n'est d'aucune utilité pour détecter les falsifications d'identité, car la période d'émission des messages ne permet pas d'identifier un transpondeur. En effet, plusieurs bateaux peuvent émettre à la même période. Ainsi, les alarmes concernant l'hypothèse $H_{0,RI}$ ne sont pas considérées pour détecter les usurpations d'identité.

4.3.2 Équation du test d'identité

Les considérations de la position et du TS, en plus du CFO, sont regroupées dans un nouveau test vérifiant l'identité. Ce test exécute les stratégies 1 et 3 pour détecter les falsifications sur la latitude, la longitude et le CFO comme présenté sur la Figure 4.5. L'hypothèse $H_{0,id}$ de ce test d'identité est valide si l'hypothèse testée par chacun de ces trois tests est valide. La considération du test de la stratégie 2, permettant de détecter le non-respect du procédé de réservation des TS, permet seulement de confirmer ou d'infirmer le résultat du test d'identité, comme nous l'avons dit précédemment. L'hypothèse $H_{0,id}$ du test a pour équation :

$$\begin{cases} H_{0,id} : \text{"le message reçu provient du transpondeur testé"} \\ H_{0,id} = (H_{0,\phi} \bigcap H_{0,\lambda} \bigcap H_{0,cfo}) \bigcup H_{0,TS} \end{cases} \tag{4.1}$$

avec $H_{0,id}$ l'hypothèse du test sur l'identité, $H_{0,\phi}$ l'hypothèse du test de la stratégie 1 sur la longitude, $H_{0,\lambda}$ l'hypothèse du test de la stratégie 1 sur la latitude, $H_{0,cfo}$ l'hypothèse

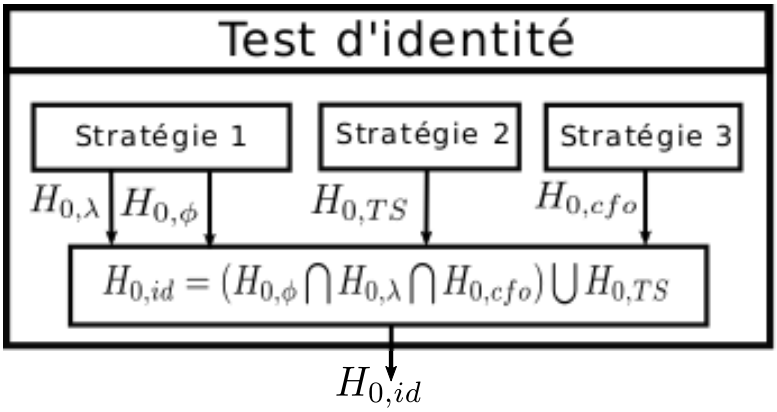

FIGURE 4.5 – Architecture du test sur l'identité

du test de la stratégie 3 sur le CFO et $H_{0,TS}$ l'hypothèse du test de la stratégie 2 sur la réservation du TS.

En utilisant l'équation du test d'identité (4.1), nous pouvons estimer grossièrement les probabilités de fausses alarmes et de non détection. En effet, nous connaissons déjà les valeurs de ces probabilités pour les tests sur la position et le CFO. Parmis ces tests, il est évident que les tests concernant la position, le CFO et le TS sont mutuellement indépendants. Cependant, les tests concernant la latitude et la longitude ne sont pas indépendants : la dynamique du navire et les conditions environnementales affectent d'une manière corrélée les innovations sur la latitude et la longitude contrôlées par ces tests introduits dans le chapitre 1. Ainsi, puisque les probabilités conditionnelles $P(H_{1,\lambda} \bigcup H_{1,\phi}|H_0)$ et $P(H_{0,\lambda} \bigcap H_{0,\phi}|H_1)$ sont inconnues, nous pouvons seulement majorer les valeurs de α_{id} et β_{id} du test d'identité pour donner une approximation. Nous obtenons les inéquations suivantes :

$$\begin{aligned} \alpha_{id} = P(H_1|H_0) &= P\left(\left(H_{1,\phi} \bigcup H_{1,\lambda} \bigcup H_{1,cfo}\right) \bigcap H_{1,TS}|H_0\right) \\ &\leq \left(P\left(H_{1,\phi}|H_0\right) + P(H_{1,\lambda}|H_0) + P(H_{1,cfo}|H_0)\right) \times P(H_{1,TS}|H_0) \\ &\leq \alpha_\phi + \alpha_\lambda + \alpha_{cfo} = 3\alpha \end{aligned} \tag{4.2}$$

Nous fixons, pour les tests sur la latitude, la longitude et le CFO, la probabilité de fausses alarmes $\alpha = 0{,}33$ %. Pour le test sur le TS, il est impossible de déterminer cette probabilité car elle dépend de paramètres difficilement évaluables (conditions environnementales, qualité de l'émetteur et du récepteur, ...). En majorant la probabilité α_{id}, nous obtenons

$\alpha_{id} \leq 1$ %. Pour la probabilité de non détection nous avons :

$$\begin{aligned}\beta_{id} = P(H_0|H_1) &= P\left(\left(\left(H_{0,\phi} \bigcap H_{0,\lambda} \bigcap H_{0,cfo}\right) \bigcup H_{0,TS}\right)|H_1\right) \\ &\leq min[P(H_{0,\phi}|H_1) \times P(H_{0,cfo}|H_1), P(H_{0,\lambda}|H_1) \times P(H_{0,cfo}|H_1)] + P(H_{0,TS}|H_1) \\ &\leq min[\beta_\phi \times \beta_{cfo}, \beta_\lambda \times \beta_{cfo}] + P(H_{0,TS}|H_1)\end{aligned} \tag{4.3}$$

La probabilité $P(H_{0,TS}|H_1)$ représente la probabilité qu'un message soit reçu du bateau qui usurpe l'identité durant le TS réservé par le bateau usurpé. Cela ne peut arriver que si le bateau usurpé n'a pas émis son message durant le TS qu'il avait réservé, car sinon il y aurait eu des interférences et aucun message n'aurait été reçu. Cet évènement est donc très rare. Il est difficile de calculer précisément sa probabilité, en tout cas, elle est négligeable.

4.4 Vérification et Identification

Il existe deux scénarios différents de falsification d'identité auxquels notre stratégie peut être confrontée. Le scénario 1 correspond à un transpondeur qui change son MMSI pour un MMSI d'un transpondeur déjà suivi par la stratégie. Ce type de scénario est présenté sur les Figures 4.6 et 4.7. Le navire avec le MMSI3 change son MMSI et prend le MMSI2. Ce scénario a été observé au large de Singapour [63]. Le scénario 2 correspond à un transpondeur qui change son MMSI pour un MMSI qui n'est pas associé à un transpondeur déjà suivi. Ce scénario, observé et présenté dans l'article [159], est également présenté sur les Figures 4.6 et 4.7, où le navire avec le MMSI4 falsifie son MMSI pour le MMSI6.

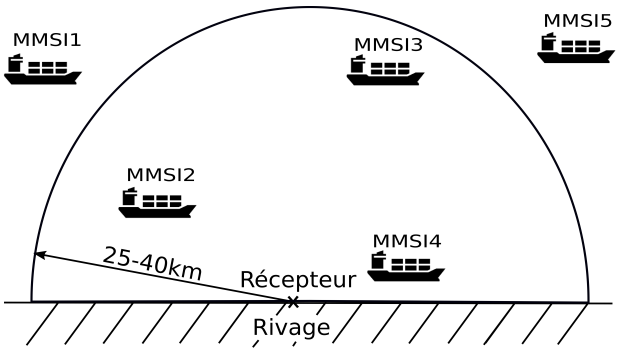

FIGURE 4.6 – Activité maritime à l'instant t0

Ces deux types de scénario sont détectés durant les étapes de *Vérification* et d'*Identification* de la stratégie globale (Figure 4.3). La *Vérification*, telle que définie dans [21], vérifie que le transpondeur est bien celui qu'il prétend être. Ce bloc est appliqué pour détecter l'usurpation d'identité du scénario 1. L'*Identification*, quant à elle, tente d'identifier le transpondeur ayant émis le message avec l'un des transpondeurs déjà suivis [21]. Elle détecte l'usurpation d'identité du scénario 2. Ce bloc est donc exécuté si le message reçu

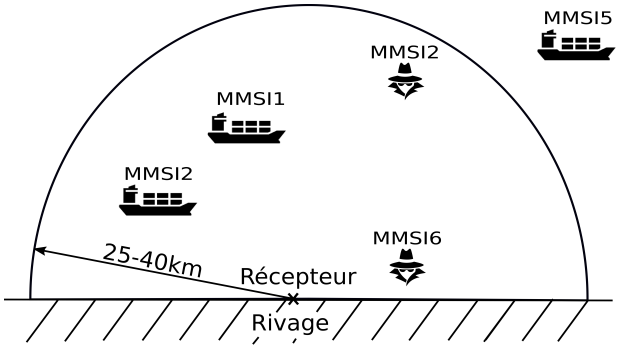

FIGURE 4.7 – Activité maritime à l'instant t1

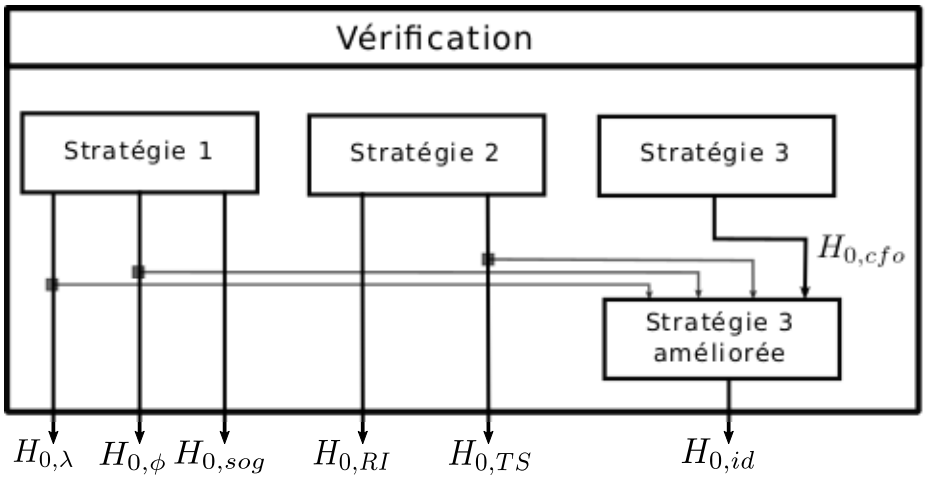

FIGURE 4.8 – Architecture de l'étape vérification avec le test d'identité

contient un MMSI qui n'a pas déjà été enregistré.

Ainsi, durant l'étape de *Vérification* le test d'identité est appliqué aux données extraites du message reçu pour vérifier leurs conformités avec la position (stratégie 1), le TS réservé (stratégie 2) et le CFO (stratégie 3) du transpondeur déjà pisté et ayant le même MMSI. C'est notamment pourquoi, un bloc *Routeur* est présent sur la Figure 4.3 pour envoyer les données du message reçu vers le transpondeur avec le même MMSI. Si le test d'identité est accepté ($H_{0,id} = 1$), le MMSI n'a pas été usurpé par un autre transpondeur. Dans le cas contraire, le MMSI a été usurpé. Durant l'étape d'*Identification*, ce test d'identité est appliqué pour tous les transpondeurs déjà suivis. Si pour un de ces transpondeurs, ce test est accepté, alors le message reçu provient de ce transpondeur qui a falsifié son MMSI pour celui contenu dans le message. Il faut noter, qu'en plus du test sur l'identité permettant de détecter les falsifications d'identité, l'étape de *Vérification* permet aussi de détecter les falsifications de position, de vitesse et les faux messages, comme présenté sur la Figure 4.8. Sur cette Figure, nous présentons une version améliorée de l'architecture de l'étape de *Vérification*, par rapport à celle que nous avions présentée sur la Figure 4.4, avec l'ajout du test sur l'identité.

4.5 Expérimentation

Pour évaluer les performances du test sur l'identité de la stratégie globale, une expérimentation sur des signaux AIS collectés dans la rade Brest est effectuée.

4.5.1 Conditions

La stratégie globale est appliquée aux signaux AIS présentés dans la partie (3.5.2) du chapitre 3. Ces signaux contiennent 8455 messages et concernent 31 navires. Nous supposons que chacun des messages provient d'un utilisateur bienveillant, et que toutes les falsifications de données (position, MMSI) et bateaux fantômes détectés sont involontaires (fausses alarmes). La raison est que l'enregistrement n'ayant pas eu lieu dans une zone de conflit, il n'y a aucune raison pour les bateaux de falsifier leurs données. Ainsi, chaque fois que l'hypothèse H_0 est rejetée par le test d'identité durant l'étape de *Vérification*, cela correspond à une fausse alarme ou un faux positif. On parle de faux positif pour une fausse alarme car le résultat du test est utilisé pour avertir (alarme, détection de virus, etc). De même, chaque fois que l'hypothèse H_0 est acceptée durant l'étape d'*Identification*, cela correspond à une non détection donc à un faux négatif.

4.5.2 Résultats

Pour le test d'identité du bloc *Vérification*, nous obtenons 30 faux positifs pour 8455 tests effectués : la probabilité de fausse alarme (α) est égale à 0,0035. Pour le test d'identité du bloc *Identification*, nous obtenons 16 faux négatifs pour 936 tests d'identification exécutés : la probabilité de non détection (β) est égale à 0,017. Par ailleurs, nous calculons ces deux probabilités α et β pour chacun des tests que contient le test d'identité. Et pour finir, nous calculons les valeurs de α et β dans le cas où le test d'identité ne prend pas en considération le résultat du test sur le TS. 78 faux positifs ($\alpha = 0{,}0092$) et 16 faux négatifs sont obtenus ($\beta = 0{,}017$). Les résultats sont présentés dans les tableaux 4.1 et 4.2.

Pour la latitude et la longitude, la valeur α est significativement inférieure à la valeur de α fixée théoriquement (0,0033). L'explication est la suivante, plusieurs navires étaient à l'ancre et ne bougeaient pas : leur innovation de position est restée égale à 0. Pour le CFO, la valeur de α est deux fois plus élevée que la théorie. En effet, le CFO dépend du SNR et peut parfois varier brusquement, ce qui induit une valeur d'innovation élevée. Sa valeur α peut être rendue moins sensible à l'environnement en faisant un test sur cinq mesures de CFO reçues consécutivement et non pas seulement sur la dernière mesure de CFO reçue. Les inégalités exprimées pour α_{id} et β_{id} en (4.2) et (4.3) sont respectées pour le test sans la considération du TS. On observe que les innovations sur la latitude et la longitude ne sont pas indépendantes, comme nous l'avions affirmé, car $\beta_{id} \approx \beta_{CFO} \times \beta_{\lambda}$. En effet, si

elles avaient été indépendantes on aurait du avoir : $\beta_{id} \approx \beta_{CFO} \times \beta_{\lambda} \times \beta_{\phi}$. Les valeurs de β pour la latitude et la longitude sont élevées, car l'environnement est un port et nous nous trouvons dans une zone côtière : certaines zones comptent de nombreux transpondeurs (comme dans le cas des ports), ce qui augmente la probabilité que deux transpondeurs aient une position proche. La valeur de β aurait été plus faible si l'enregistrement avait été effectué en pleine mer.

Néanmoins, dans cette application, la considération de la latitude et de la longitude, en plus du CFO dans le test d'identité, divise la valeur de β presque par trois. Cela montre l'intérêt d'utiliser plusieurs stratégies conjointement. Ainsi, le test d'identité devient puissant ($\beta_{id} < 0{,}05$) pour cette application, alors que cela n'aurait pas été le cas si l'on avait considéré seulement le CFO. Par ailleurs, la considération de la réservation du TS, en plus, dans le test d'identité, permet de diviser par 3 la valeur de α. β quant à lui n'est pas modifié.

Tableau 4.1 – Tests de validité sur le CFO, la latitude et la longitude en utilisant des données réelles.

	CFO		**latitude**		**longitude**	
	H_0 vraie	H_1 vraie	H_0 vraie	H_1 vraie	H_0 vraie	H_1 vraie
H_0 **acc.**	$0{,}9914(1-\alpha)$	$0{,}0438(\beta)$	0,9993	0,3825	0,9996	0,4359
H_1 **acc.**	$0{,}0086(\alpha)$	$0{,}9562(1-\beta)$	0,0007	0,6175	0,0004	0,5641

Tableau 4.2 – Résultats du test d'identité appliqué sur des données réelles en ne prenant pas en considération le TS et en le prenant en considération.

	test d'identité (sans TS)		**test d'identité (avec TS)**	
	H_0 vraie	H_1 vraie	H_0 vraie	H_1 vraie
H_0 **acc.**	0,9908	0,0171	0,9965	0,0171
H_1 **acc.**	0,0092	0,9829	0,0035	0,9829

Une version du code détectant les falsifications d'identité en appliquant les stratégies 1 et 3 est disponible en ligne [91]. Les signaux AIS ayant servi à l'expérimentation et dont les conditions d'enregistrement ont été présentées dans la partie 3.5.1 du chapitre 3 sont également mis à disposition. C'est la première fois que des signaux AIS bruts en bande de base sont mis en libres accès.

4.5.3 Limites de la stratégie

Premièrement, notre stratégie ne peut détecter les falsifications d'identité que dans la zone qu'elle contrôle, c'est-à-dire un cercle d'un rayon de 40km dont le récepteur appliquant la stratégie est le centre. En dehors de cette zone, aucune manipulation de l'AIS ne peut être détectée. Néanmoins, cette zone de contrôle peut être agrandie par l'utilisation de plusieurs antennes ou en plaçant l'antenne à une haute altitude. Cependant, dans ce

cas, la nouvelle zone de contrôle peut couvrir plusieurs cellules, du fait de la topologie cellulaire du réseau AIS pour empêcher les interférences. Les cellules font entre 25 et 40 km de rayon, comme présenté dans le chapitre 2. Or, le mode d'accès TDMA assure l'absence d'interférences à l'intérieur d'une cellule, mais il n'y aucune protection contre les interférences sur un espace couvrant plusieurs cellules. C'est pourquoi, il faudra adapter la stratégie 2, car plusieurs messages pourront être émis durant un même TS.

Par ailleurs, dans une zone contrôlée, un utilisateur peut arrêter d'émettre des messages durant 420 s pour être retiré des bateaux suivis par la stratégie, et ensuite émettre avec un nouveau MMSI falsifié. Si le nouveau MMSI n'est pas déjà enregistré par notre stratégie, il sera impossible pour notre stratégie de détecter cette falsification d'identité. Une solution serait de prendre en considération l'énergie des signaux transmis pour détecter si un arrêt d'émission de message par un des bateaux est justifié. Ce type de travail a été exploré dans [105] et il reste à l'adapter à notre stratégie.

4.6 Conclusion

Dans ce chapitre, nous avons développé une stratégie globale qui applique conjointement les trois stratégies que nous avions présentées dans les précédents chapitres. La stratégie globale détecte les falsifications de données (position, vitesse et identité) et les bateaux fantômes. Cette stratégie a été appliquée à des signaux bruts enregistrés dans la rade de Brest et a montré la complémentarité qu'il y avait entre ces trois stratégies pour détecter les falsifications d'identité. Alors, que les probabilités de fausse alarme et de non détection étaient, lorsqu'on considérait seulement le CFO, respectivement de $\alpha = 0{,}0086$ et de $\beta = 0{,}0438$, maintenant, en considérant en plus la position et les TS réservés, ces deux probabilités deviennent respectivement de $\alpha = 0{,}0035$ et de $\beta = 0{,}0171$.

Chapitre 5

Conception sur FPGA du récepteur AIS intelligent

5.1 Introduction

Dans les précédents chapitres, plusieurs stratégies ont été présentées pour détecter les falsifications de position, de vitesse et d'identité, et les bateaux fantômes. Dans ce chapitre, nous concevons le récepteur AIS intelligent qui embarque ces stratégies. Le chapitre 7 qui suivra, présentera, quant à lui, une méthode de vérification de ce récepteur.

5.1.1 Architecture du récepteur

L'architecture du récepteur AIS intelligent à concevoir est présentée sur la Figure 5.1. Ce récepteur possède :

- une antenne pour capter les signaux ;
- deux filtres passe-bande pour sélectionner le signal utile et supprimer le bruit ;
- un LNA (Low noise amplifier) pour amplifier les signaux ;
- un bloc transposant le signal en bande de base. Il retire 162 MHz au signal. La fréquence du signal devient donc soit -25 kHz (Canal A) ou, soit 25 kHz (Canal B) ;
- un ADC (Analog-to-digital converter) pour convertir les signaux analogiques en signaux numériques. Il échantillonne les signaux à la fréquence 192 kHz. Sachant que le débit est de 9600 bits s^{-1}, nous avons donc 20 échantillons du signal par bit, ce qui fait 5120 échantillons par message (256 bits par message (partie 2.1)) ;
- un bloc décodant les messages et extrayant les informations qu'ils contiennent en plus du temps d'arrivée des messages. L'architecture de ce bloc est expliquée dans la partie (5.3.2) ;
- un bloc appliquant les stratégies que nous avons développées et renvoyant des indications sur les falsifications rencontrées à l'interface utilisateur.

Parmi tous ces blocs, seulement les deux blocs encadrés en bleu sont développés et implémentés sur FPGA. Le choix du FPGA sera expliqué dans la prochaine sous-partie. Le FPGA reçoit en entrée des signaux en bande de base en quadrature (I/Q) à la fréquence ± 25 kHz et transmet en sortie, à l'interface utilisateur, les informations contenues dans

les messages décodés et les indications sur les falsifications rencontrées. Le début de la chaîne de traitement des signaux allant de l'antenne jusqu'à l'ADC n'a pas été conçue par manque de temps.

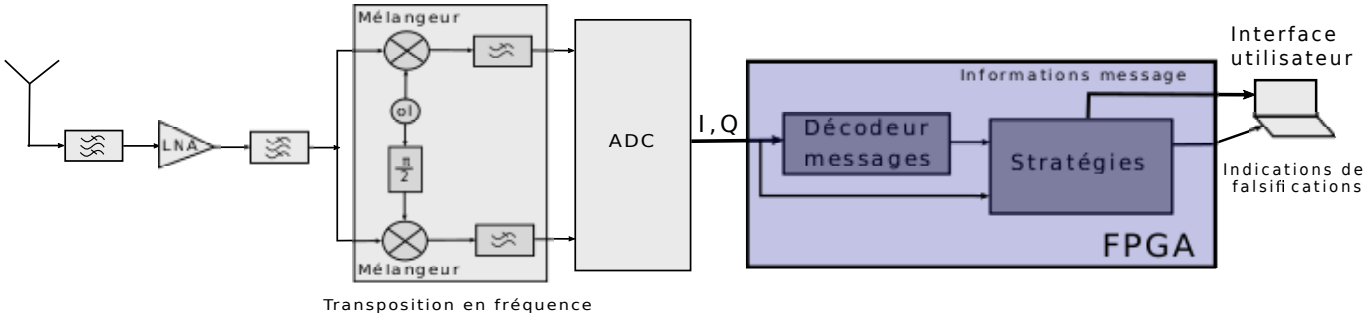

FIGURE 5.1 – Architecture du récepteur AIS intelligent à concevoir.

5.1.2 Méthode de conception appliquée

Habituellement, la conception de tels systèmes peut se limiter à l'utilisation de microcontrôleurs, car ce type de circuits intégrés est facilement programmable et peu coûteux. Cependant, l'architecture d'un tel microcontrôleur est fixée et guère propice à des extensions modulaires, comme il est souhaitable ici. C'est pourquoi, nous nous sommes orientés vers une implémentation sur FPGA. Ce circuit contient à la fois une grande quantité de ressources matérielles et est reconfigurable par essence : il peut être personnalisé pour s'adapter à n'importe quelle évolution fonctionnelle du système. Cette capacité d'adaptation de l'architecture est appropriée à notre approche de développement qui applique la méthode Agile [127]. Par cette méthode, le développement se fait de manière itérative et incrémentale : les performances du récepteur sont améliorées et son architecture développée et étendue à mesure que de nouvelles stratégies lui sont intégrées. Par ailleurs, le FPGA offre une puissance de calcul élevée, ce qui est essentiel, car les stratégies doivent être exécutées en temps réel. Notons enfin que notre conception, qui est donc fondée sur un FPGA, nous a également permis de dégager une méthode de vérification originale, qui sera exposée dans le chapitre suivant, et qui s'harmonise avec le développement du récepteur.

Néanmoins, le recours à de telles architectures FPGA reste un challenge technique. Jusqu'à présent, un des grands obstacles à l'utilisation d'un FPGA est la nécessité de l'implémenter avec un code écrit avec un bas niveau d'abstraction (niveau RTL). Ce type d'implémentation exige une bonne expertise en conception matérielle, particulièrement lorsque, comme dans notre cas d'application, des algorithmes complexes sont implémentés (filtre IMM). Or, bien souvent, les ingénieurs de traitement du signal, qui modélisent la chaîne de traitement des données du système, n'ont pas cette compétence et ne peuvent donc mener à bien la conception dans son ensemble.

Heureusement, la synthèse haut niveau (HLS), qui a pourtant connu une courbe d'adoption complexe chez les ingénieurs, permet désormais de dépasser cet obstacle. En utilisant la HLS, les ingénieurs peuvent, à partir d'une modélisation de tout ou partie du système, synthétiser automatiquement le circuit numérique correspondant pour qu'il soit implémenté sur FPGA ou ASIC (Application-specific integrated circuit). Le langage d'entrée est généralement un vaste sous-ensemble du langage C ou C++, même si des approches alternatives existent à partir de langages plus spécialisés comme Matlab. Cette approche basée sur la HLS offre de nombreux avantages, notamment une réduction significative des temps de conception et de mise sur le marché, une meilleure productivité et autonomie des ingénieurs [79, 35], et une flexibilité accrue [79]. Par ailleurs, la HLS facilite l'exploration de l'espace de conception (DSE (Design space exploration)), et permet ainsi de tester rapidement un grand nombre d'architectures différentes du système afin de trouver celle ayant le meilleur compromis performance, consommation et espace occupé [131]. Pour finir, l'utilisation de la HLS s'inscrit dans l'approche *top-down* préconisée par l'ESL [101] qui vise à augmenter la productivité de la conception de systèmes électroniques. En effet, cette approche part d'une modélisation du système complet avec un haut niveau d'abstraction, pour ensuite, par affinement successif, grâce en particulier à la HLS, aboutir à sa description matérielle.

Toutefois, jusqu'à présent, la HLS était utilisée seulement pour concevoir des sous-parties de systèmes, car, dès que la complexité ou la longueur du code devenait trop élevée, la synthèse n'était plus possible. Cette situation a évolué ces dernières années du fait de l'amélioration et du gain en performances des outils de HLS. Nous avons donc pu vérifier ici que l'utilisation de la HLS au niveau système, permet de concevoir notre récepteur AIS complet. Ces résultats sont exposés dans le présent chapitre.

5.2 Introduction à la HLS

5.2.1 Histoire

L'idée de la HLS a été émise dans les années 1970 à 1980, mais ce n'est que depuis les années 2010, grâce à l'amélioration des outils proposés, qu'il s'agit d'une solution viable pour générer des descriptions matérielles [102]. L'une des raisons de cette lente adoption est que la qualité des résultats (QoR) était initialement médiocre par rapport à l'approche RTL [102]. Cependant, l'outil s'est amélioré depuis, et alors qu'il était utilisé seulement pour la synthèse de sous-partie de système, ou d'algorithme avec un nombre de lignes limité (< 1000) [89], il semble qu'aujourd'hui, il puisse synthétiser des systèmes complexes en entier [79].

5.2.2 Flot de synthèse

Le flot de synthèse traditionnel de la HLS est représenté sur la Figure 5.2. Ce flot démarre d'une modélisation comportementale du système dans le langage C/C++, puis analyse et transforme cette modélisation en CFG (Control flow graph). Le CFG représente les différentes instructions du code et leurs relations logiques. Ensuite, le DFG (Data flow graph) est créé pour représenter la dépendance entre les données et les opérations. Il permet de faire apparaître le parallélisme d'instruction. Ces trois premières étapes sont représentées sur la Figure 5.3 et le bloc CDFG (Control data flow graph) de la Figure 5.2 correspond aux graphes CFG et DFG. Ensuite, une étape d'allocation, une étape d'ordonnancement et une étape d'assignation sont appliquées successivement. Durant l'allocation, les ressources matérielles nécessaires pour implémenter le DFG sont allouées, par exemple en utilisant des blocs préfabriqués de la bibliothèque de conception. Durant l'ordonnancement, l'ordre d'exécution des différentes instructions du code, représentées sur le DFG, est fixé. Et durant l'assignation, l'outil HLS optimise l'utilisation des ressources matérielles disponibles, tout en respectant les contraintes de temps et les exigences de conception. Une fois ces trois étapes terminées, le code matériel est généré. Ce code peut être un code VHDL, Verilog ou un autre langage de description matérielle. Par ailleurs, des directives peuvent être appliquées pour optimiser l'architecture matérielle du système conçu et améliorer le compromis entre performances, surface matérielle occupée et consommation d'énergie.

5.3 Modélisation du récepteur AIS intelligent

Pour la conception du récepteur AIS intelligent, nous utilisons Vitis HLS 2020.2 [157] qui est développé par Xilinx. Cet outil génère automatiquement la description matérielle du récepteur, exprimée dans le langage VHDL, à partir d'un code écrit en C++. Dans cette partie, nous présentons et expliquons la méthode appliquée pour modéliser et programmer le comportement du récepteur dans le langage C++. .

5.3.1 Choix de la représentation des variables

Des variables à virgule fixe (fixed point) ou des variables à virgule flottante (floating point) peuvent être utilisées pour la programmation du récepteur. Les variables à virgule fixe sont représentées avec une virgule qui ne change pas de position, contrairement aux variables à virgule flottante, représentées avec une mantisse et un exposant, ce qui leur permet d'avoir une précision variable en fonction de l'étalement des valeurs.

Les variables à virgule fixe ont l'avantage d'être plus simples à implémenter et nécessitent moins de ressources pour les calculs. En revanche, elles ont une plage de valeurs limitée et une précision fixe qui peut ne pas suffire dans certaines applications. Les va-

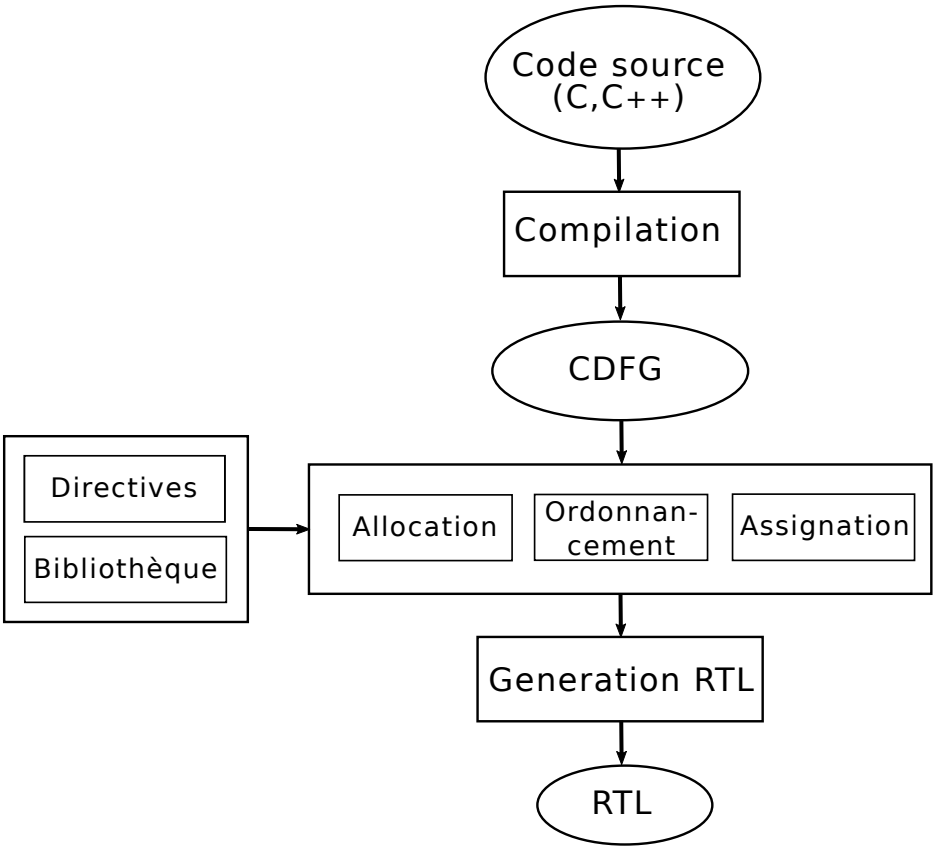

FIGURE 5.2 – Flot de synthèse de la HLS

riables à virgule flottante ont l'avantage de couvrir une plage de valeurs plus grande et offrent une précision relativement constante sur cette plage. Elles permettent également de travailler avec des variables de très grande ou très petite amplitude. En revanche, elles sont plus complexes à implémenter et nécessitent plus de ressources pour les calculs.

Considérant ces différents aspects et le fait que l'utilisation de variable à virgule flottante par la HLS détériore souvent les performances du système [134, 37], en plus de nécessiter l'utilisation d'IP (Intellectual property) de Xilinx. Nous choisissons donc d'utiliser des variables à virgule fixe pour la modélisation. Pour simplifier l'utilisation de ce type de variable, Vitis HLS met à disposition de l'utilisateur une bibliothèque dédiée. Cette bibliothèque est utilisable avec les types `ap_fixed` et `ap_ufixed`, respectivement pour les variables signées ou non signées. Avec ces types, l'utilisateur peut fixer le nombre de bits utilisés pour représenter les parties décimales et entières des variables. N'importe quel opérateur utilisé en C/C++ peut être utilisé par ce type de variable. Si la partie décimale des deux arguments d'un opérateur n'est pas la même, Vitis HLS se charge de gérer l'alignement automatiquement. Par ailleurs, Vitis HLS offre différentes options, présentées sur le site de Xilinx [26], pour gérer le dépassement de capacité (lorsque le résultat comporte plus de MSB (Most significant bit) que le type assigné ne le permet) et l'arrondit (lorsque le résultat comporte moins de LSB (Less significant bit) que le type assigné ne le permet).

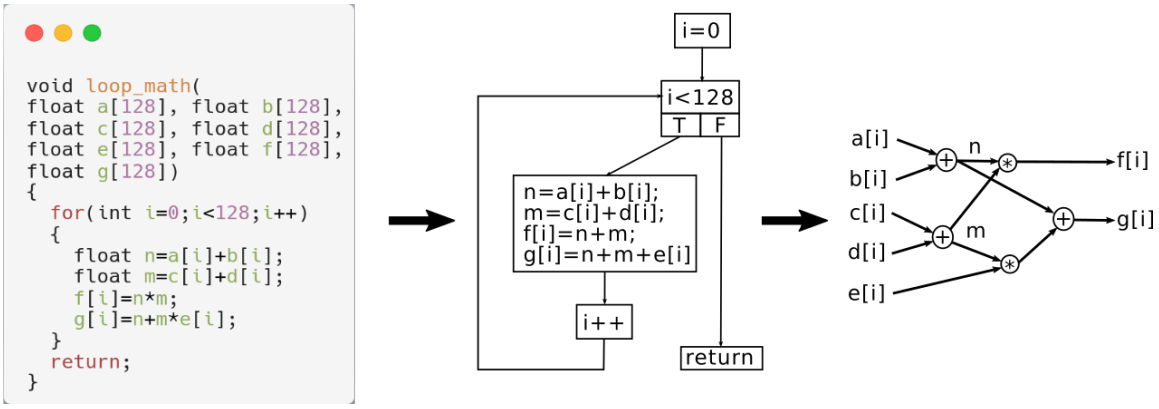

FIGURE 5.3 – Génération du CFG et du DFG à partir du code C++.

5.3.2 Programmation du décodeur de messages

Architecture matérielle

Le bloc décodant les messages reçoit en entrée des signaux en bande de base à ±25 kHz en quadrature (I/Q). En sortie, ce bloc renvoie les informations des messages contenues dans ces signaux et le CFO. Pour cela, plusieurs blocs s'exécutent en série, comme présenté sur la Figure 5.4. Les fonctions exécutées par tous ces blocs seront expliquées plus en détails dans les prochaines parties. En entrée de cet ensemble de blocs se trouve un bloc *Interface* constitué de buffers, d'un micro-contrôleur, de bus et de FIFO (First in first out). L'*Interface* gère la communication et l'envoi d'échantillons au premier bloc. Nous représentons, en plus, sur la partie gauche de la Figure 5.4, la partie de la stratégie globale, introduite dans le chapitre (4), qui correspond à cet ensemble de blocs.

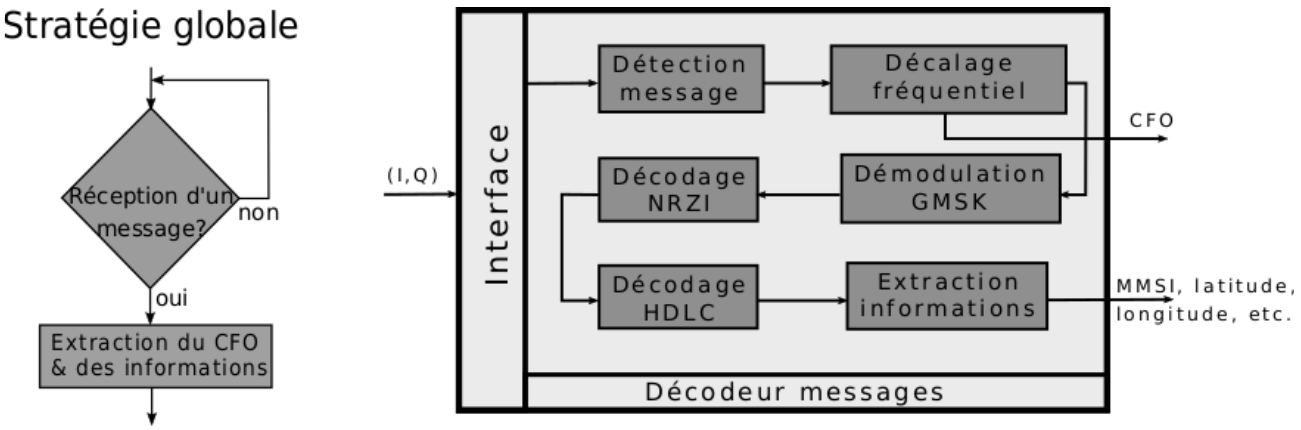

FIGURE 5.4 – Représentation de la partie de la stratégie globale correspondant au décodeur de message (gauche). Architecture du bloc permettant de démoduler les signaux pour en extraire le CFO et les données AIS (droite).

Détection de l'arrivée d'un message

Ce bloc permet de séparer les signaux de bruit des signaux utiles codant un message. Pour cela, 32 échantillons sont lus sur le bloc interface et l'énergie de l'ensemble de ces échantillons est calculée. Si cette énergie est supérieure à un seuil T alors un message est en train d'être reçu ; 5120 échantillons (nombre d'échantillons d'un message) sont alors enregistrés et transmis au bloc suivant. La lecture est bloquante, c'est-à-dire qu'il faut attendre d'avoir reçu ces 32 échantillons pour démarrer l'exécution du bloc. Nous présentons ci-dessous l'Algorithme permettant de détecter l'arrivée d'un message.

Algorithm 3 Détection arrivée message

Data: I_{in}, Q_{in}
Result: I_{out}, Q_{out}
for $i = 1$ ***to*** 32 **do**
  $I[i] = I_{in}.read()$
  $Q[i] = Q_{in}.read()$
end
E=energy(I,Q)
if $E>T$ **then**
  for $i = 1$ ***to*** 5120 **do**
    $I_{out}[i] = I_{in}.read()$
    $Q_{out}[i] = Q_{in}.read()$
  end
end

Décalage fréquentiel

Le message peut avoir été transmis soit sur le canal A ou, soit sur le canal B. Pour un message transmis sur le canal A, la fréquence centrale est de -25 kHz, et pour un message transmis sur le canal B, la fréquence centrale est de 25 kHz. Ce bloc permet de retirer la composante fréquentielle à ± 25 kHz des signaux. Parfois, dans les 5120 échantillons, il peut y avoir deux messages transmis, l'un sur le canal A et l'autre sur le canal B. Dans ce cas, deux messages sont extraits de ce bloc. L'algorithme qui permet de transposer en fréquence le signal est présenté ci-dessous (Algorithm 4). T_e est la période d'échantillonnage et f_s est la composante fréquentielle retirée au signal. En fonction du canal sur lequel est transmis le message, f_s vaut soit 25 kHz ou -25 kHz. Afin de ne pas dépendre de l'IP de Xilinx pour le calcul du cosinus et du sinus, un Tableau contenant 2048 valeurs (LUT (Look-up table)) est utilisé par chacune de ces deux fonctions. Avec 2048 valeurs, le cosinus et le sinus sont calculés avec une précision qui, dans le pire des cas, vaut $\frac{360}{2048\times 4} = 0{,}0439°$. Une façon de programmer le cosinus et le sinus avec une LUT est présentée dans la thèse [150].

Algorithm 4 Transposition en fréquence

Data: I_{in}, Q_{in}, f_s

Result: I_{out}, Q_{out}

for $i = 1$ ***to*** 5120 **do**

$I_{out}[i] = I_{in}[i]cos(2\pi f_s T_e i) - Q_{in}[i]sin(2\pi f_s T_e i)$

$Q_{out}[i] = Q_{in}[i]cos(2\pi f_s T_e i) + I_{in}[i]sin(2\pi f_s T_e i)$

end

Démodulation GMSK

La chaîne suivante, reproduite sur la Figure 5.5 et tirée de [49], permet d'appliquer une démodulation GMSK aux signaux, et d'en extraire les bits codés. Un premier filtre passe-bas est d'abord appliqué aux signaux reçus pour extraire le bruit. Ensuite, la phase du signal (I,Q) est calculée en appliquant la fonction `atan`. Comme pour les fonctions sinus et cosinus, cette fonction utilise un Tableau de valeurs. Le code de cette fonction a été pris à la référence [34]. Puis, un dérivateur est appliqué pour obtenir la fréquence instantanée du signal. Ensuite, un bloc *synchronisation* permet d'identifier le premier échantillon de la séquence de conditionnement en appliquant une corrélation entre le signal reçu et la séquence de conditionnement. Enfin, à partir de la fréquence instantanée, le bloc *prise de décision* décode les bits en utilisant un seuil. Ce seuil est égal au CFO ; les valeurs au-dessus correspondent au bit '1' et les valeurs en dessous au bit '0'. Le CFO est l'offset en fréquence sur la porteuse causé par les imperfections matérielles du transpondeur de l'émetteur et du récepteur ; cette donnée a été introduite dans le chapitre 3.

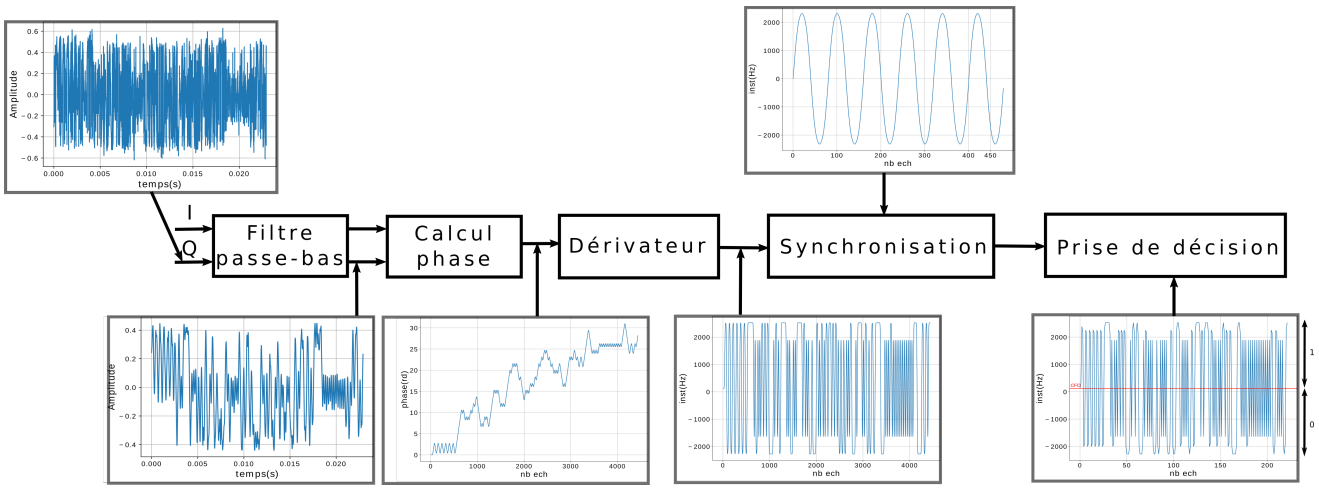

FIGURE 5.5 – Chaîne de démodulation GMSK

Décodage NRZI, HDLC et extraction information

Une fois que les bits ont été décodés, un décodage NRZI (Non return to zero inverted) est appliqué. Ensuite, un décodage HDLC retire le bourrage de zéro, vérifie le CRC et

inverse les octets. Enfin, les informations sont extraites ; ces informations sont présentées en Annexe (D) ainsi que le nombre de bits sur lesquels elles sont codées. Une description détaillée du type de CRC utilisé et du fonctionnement du protocole HDLC se trouve dans la norme AIS [133]. Le code C permettant de calculer et de vérifier le CRC est disponible grâce au lien [36].

Conclusion

Dans cette sous-partie, tous les blocs constituant la chaîne de traitement des signaux AIS en bande de base du bloc *décodeur de messages* du récepteur AIS ont été expliqués et programmés en C++. Le récepteur conçu permet de récupérer le CFO et les informations contenues dans les messages à partir de ces signaux. Il reste maintenant à exploiter et vérifier ces données en appliquant les stratégies développées dans les quatre premiers chapitres pour détecter des falsifications d'information pouvant être contenues dans les messages. Ces stratégies sont programmées dans la prochaine sous-partie.

Remarques

L'intérêt de partir d'une modélisation du système avec un code en C++ est, comme nous l'avons montré, que bien souvent le code est déjà disponible sur internet, quand bien même des variables à virgule fixe seraient utilisées. Par contre, les stratégies que nous avons développées utilisent des algorithmes de traitement du signal complexes et peu utilisés : le code programmant ces algorithmes n'est pas disponible en libre accès sur internet. C'est pourquoi, dans la prochaine sous-partie, la stratégie 1 et en particulier le filtre IMM qu'elle contient, seront programmés en partant de zéro. Par ailleurs, des LUT ont été utilisés pour programmer les fonctions trigonométriques (cos, sin, atan) pour éviter que le code VHDL synthètisé par Vitis HLS utilise des IP. Ainsi, notre code VHDL sera utilisable pas n'importe quel simulateur et implémentable sur n'importe quel FPGA.

5.3.3 Programmation des stratégies détectant les falsifications de messages

Architecture du bloc Stratégies

Après avoir extrait les informations contenues dans les signaux AIS, nous programmons en C++, dans cette sous-partie, la stratégie globale, présentée dans le chapitre 4, et dont l'architecture est rappelée sur la Figure 4.3. Sur cette Figure, le bloc du *décodeur de messages*, de la Figure 5.4, correspond au haut de l'architecture et le bloc *Stratégies* au bas de l'architecture. Normalement, dans la partie correspondant au bloc *Stratégies* du schéma, un test global, expliqué dans le chapitre (4) doit être appliqué. Ce test global est constitué des stratégies suivantes développées dans les précédents chapitres :

— la stratégie 1 détectant les falsifications de position et de vitesse par pistage de position (1) ;
— la stratégie 2 détectant les bateaux fantômes en contrôlant le respect du mode d'accès TDMA par les bateaux (2) ;
— la stratégie 3 détectant les falsifications ou usurpations d'identité par pistage du CFO (3).

De ces trois stratégies, seulement une seule a été programmée par manque de temps : la stratégie 1. Cette stratégie permet de détecter les falsifications de position et de vitesse. Ici, nous n'avons programmé que la détection des falsifications de position. Il faut noter que les bateaux fantômes et les falsifications d'identité ne sont pas détectés. C'est pourquoi, l'architecture initiale de la stratégie globale présentée sur la Figure 4.3 ne contient plus les blocs *Vérification* et *Identification* car ils servaient à détecter les falsifications d'identité.

Programmation en C++ de l'Algorithme 1

Pour détecter les falsifications de position, la stratégie 1 applique l'Agorithme 1, présenté dans le chapitre 1. Cet algorithme est programmé en C++ avec des variables à virgules fixe. La difficulté de cette programmation est causée par la complexité du filtre IMM dont les équations ont été présentées dans la partie (1.5.2) : il contient une fonction exponentielle et racine carrée et certaines de ses variables ont des valeurs avec un ordre de grandeur à 10^{-12}.

Une des composantes de la matrice de covariance de l'erreur sur le modèle (matrice Q) peut avoir des valeurs de l'ordre de grandeur de 10^{-12}. Cet ordre de grandeur n'est pas représentable sur 32 bits avec des variables à virgules fixe utilisant les bibliothèques fournies par Vitis HLS. Pour résoudre cette difficulté, toutes les matrices de covariance, intervenant dans le filtre IMM (Q, $\widehat{P}$ et R), sont exprimées en mètre et non en degré. Pour passer des degrés aux mètres, la position du récepteur (position École Navale) et les rayons équatorial et polaire sont pris en considération. Une autre solution aurait pu être d'utiliser des variables ayant une longueur supérieure à 32 bits pour stocker les valeurs. En effet, Vitis HLS autorise d'utiliser jusqu'à 1024 bits pour représenter les variables. Aussi, des options peuvent être ajoutées pour régler le mode de quantification (lorsque la précision demandée est supérieure à celle qui peut être définie par le plus petit bit fractionnaire de la variable utilisée pour stocker le résultat) et de débordement (lorsque le résultat d'une opération dépasse la valeur maximale (ou minimale dans le cas de nombres négatifs) pouvant être stockée dans la variable utilisée pour stocker ce résultat).

Pour ailleurs, comme pour les fonctions sinus et cosinus, un Tableau de valeurs (LUT) est utilisé pour coder la fonction exponentielle. Les valeurs de la fonction exponentielle varient entre 0 et 10 avec un pas de 0,01. Pour coder la fonction racine carrée, nous avons récupéré le code suivant [106] en libre accès sur internet.

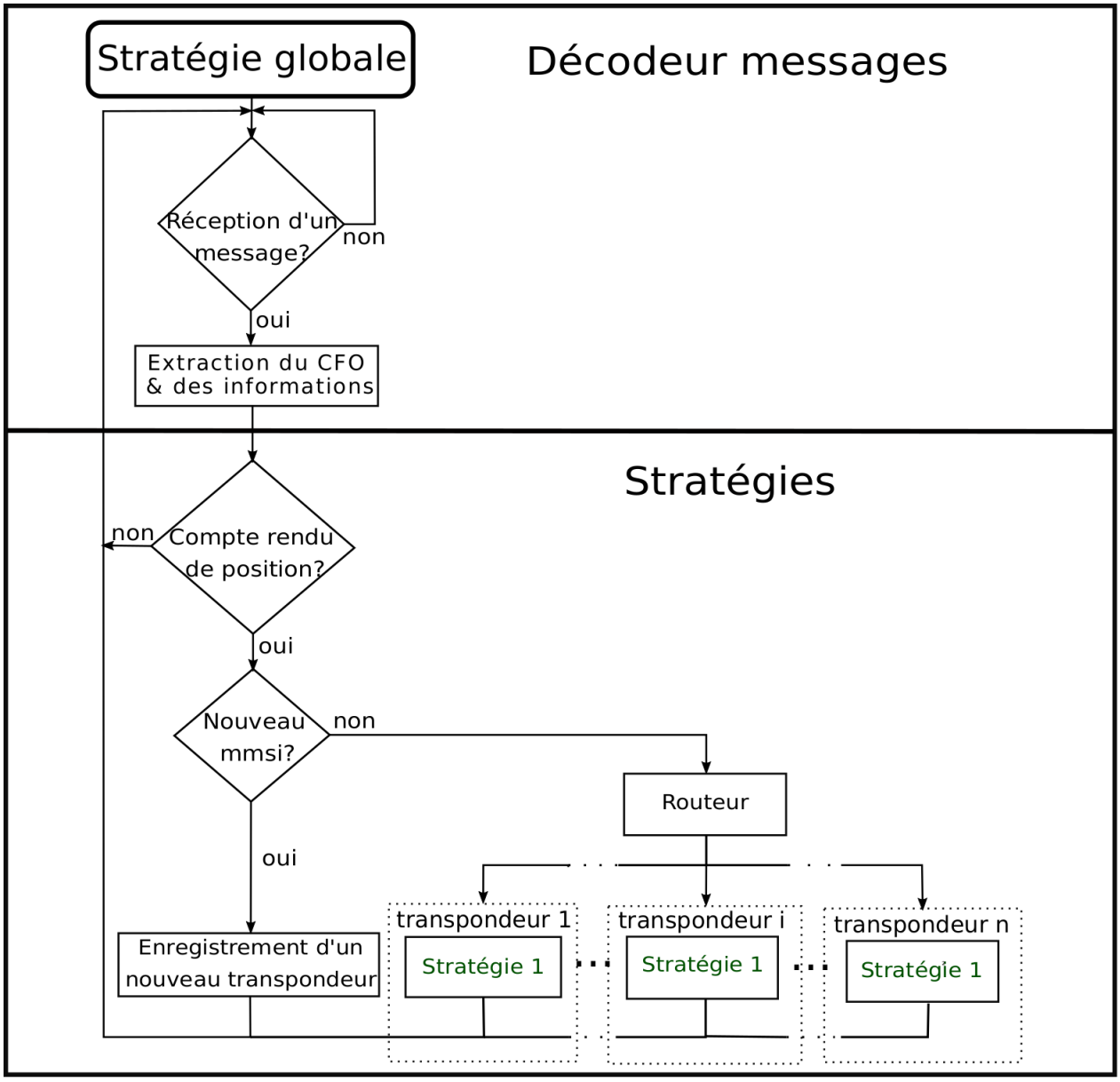

FIGURE 5.6 – Architecture de la stratégie globale implémentée sur FPGA

Évaluation des performances de la stratégie

L'efficacité de la stratégie détectant les falsifications de position dépend de la précision du filtre IMM. Cette précision a déjà été évaluée, dans le chapitre 1, par une simulation de Monte Carlo. Cependant, le filtre avait été programmé sous Matlab à l'aide de variables à virgule flottante, alors que pour utiliser l'outil Vitis HLS, nous venons de programmer de nouveau ce filtre à l'aide de variables à virgule fixe. La précision du filtre est donc de nouveau évaluée par la même simulation de Monte Carlo pour 20 exécutions et pour une route COG $= 45°$. Comme pour le chapitre 1, chaque mesure de latitude et de longitude est affectée d'un bruit blanc gaussien d'écart-type $\sigma_v \propto 5$ m. Nous traçons ci-dessous (Figures 5.7 et 5.8) l'erreur quadratique moyenne (RMSE) sur la position et l'évolution du seuil de validation et de l'innovation sur la longitude pour la simulation faite sous Matlab et

sous Vitis HLS. Nous observons que la perte en précision est très légère entre l'utilisation de variables à virgule fixe et de variables à virgule flottante. À titre d'exemple, pour le RMSE sur la position, la moyenne et l'écart type de la différence entre les variables à virgule fixe (Vitis HLS) et les variables à virgule flottante valent respectivement 0,3199 m et 0,3042 m. Ainsi, l'Algorithme 1 qui sera implémenté sur un FPGA sera aussi efficace, pour détecter les falsifications de position, que s'il tournait sous Matlab.

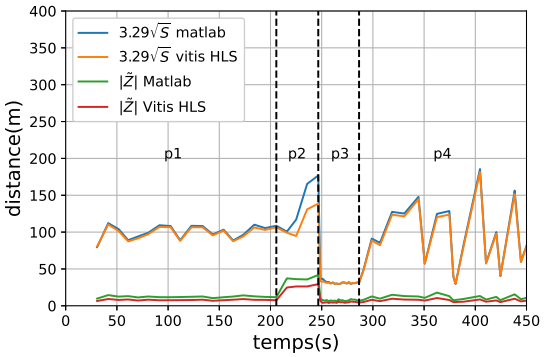

FIGURE 5.7 – Evolution du seuil de validation et de l'innovation sur la longitude sous Vitis HLS et Matlab pour COG = 45°

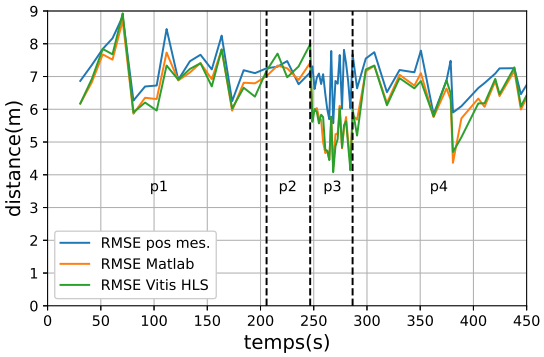

FIGURE 5.8 – Comparaison du RMSE estimée sur la position calculée sous Vitis HLS et Matlab pour COG = 45°

Conclusion

Dans cette sous-partie, nous avons programmé en C++ la stratégie globale pour détecter les falsifications de messages. Par manque de temps, seulement la stratégie 1 permettant de détecter les falsifications de position a été programmée, ce qui a entraîné une modification de l'architecture de la stratégie globale par rapport à celle définie dans le chapitre 4. La stratégie 1 applique un filtre IMM qui a été programmé en utilisant des variables à virgule fixe. L'utilisation de ce type de variables a demandé d'apporter quelques modifications au code pour gérer la plage de valeurs étendue des variables du filtre IMM. Néanmoins, la comparaison des performances de cette stratégie, programmée avec ce type de variable, avec celle ayant été programmée sous Matlab avec des variables à virgule flottante dans le chapitre 1, a montré des performances semblables.

5.4 Conception du récepteur AIS intelligent

Dans la partie précédente, le code C++ du récepteur AIS intelligent a été présenté. Dans cette partie, nous appliquons Vitis HLS à ce code pour générer la description matérielle en VHDL du récepteur qui sera implémentée sur la FPGA. Les caractéristiques matérielles du récepteur sont évaluées et son fonctionnement vérifié.

5.4.1 Synthèse matérielle

La synthèse matérielle est appliquée sur tout le code du récepteur AIS intelligent qui fait 2500 lignes de C++. La période de l'horloge est fixée à 10 ns. Toutes les fonctions sont programmées naïvement sans utiliser de directives d'optimisation pour montrer que déjà les performances du système implémenté sont très bonnes, et que n'importe quel ingénieur de traitement du signal peut utiliser cet outil sans véritable connaissance en conception matérielle. La quantité de ressources utilisées par le système est présentée sur le Tableau 5.1. La plateforme recevant la description matérielle est le Virtex UltraScale+ HBM VCU128 FPGA [156] (environ 10^6 cellules logiques du système, 3K tranches DSP, etc.). Cette plateforme possède une énorme quantité de ressources, car lors de la vérification, présentée dans le prochain chapitre, le simulateur sera aussi implémenté sur cette plateforme. Ainsi, une fois la conception et la vérification terminées, l'utilisateur pourra implémenter son système sur un plus petit FPGA moins onéreux.

La synthèse permet aussi d'obtenir une estimation du temps d'exécution du système complet et de l'ensemble des blocs qui le composent. Les temps d'exécution apparaissent sur le Tableau 5.1. Le Récepteur AIS met 2,68 ms à s'exécuter. Ce résultat est largement satisfaisant et permet de valider la contrainte d'exécution en temps réel, puisqu'au maximum deux messages (un sur chaque canal) sont reçus toutes les 26,7 ms. Il reste encore beaucoup de temps disponible pour exécuter les autres stratégies développées qui n'ont toujours pas été implémentées ou celles qui seront amenées à être développées par de nouvelles recherches. L'implémentation de la stratégie 3 devrait occuper une grande partie de ce temps disponible, car cette stratégie applique une FFT (Fast Fourier transform). Par ailleurs, le processus de développement mis en place s'inscrit dans la philosophie prônée par la méthode *Agile* [127] : le récepteur conçu sert de structure de base sur laquelle sont intégrées, au fur à mesure des mises à jour, de nouvelles stratégies augmentant l'efficacité de détection des falsifications du système. Il est donc tout à fait naturel que le FPGA soit, pour le moment, sous-utilisé.

Tableau 5.1 – Utilisation des ressources du FPGAs

	BRAM	**DSP**	**FF**	**LUT**	**Latence (ms)**
Récepteur AIS	**31** **(0 %)**	**277** **(∼ 3 %)**	**125 411** **(4 %)**	**194 689** **(14 %)**	**2,68** -
Transposition fréquence	5	1	11 816	9824	0,15
Démodulation GMSK	22	147	43 279	31 978	2,62
Décodage NRZI	0	0	527	116	$2{,}58 \times 10^{-3}$
Décodage HDLC	0	0	3286	8845	$5{,}27 \times 10^{-3}$
Extraction d'information	0	0	827	3285	~ 0
Algorithme 1	2	121	42 677	41 324	$1{,}25 \times 10^{-2}$

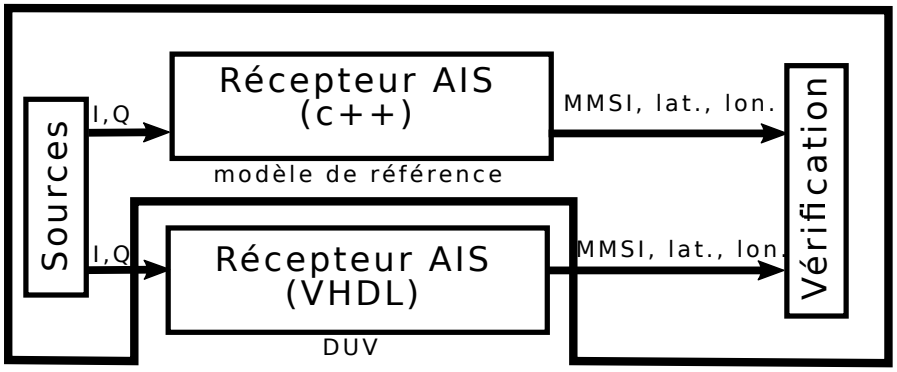

FIGURE 5.9 – Banc de tests utilisé pour vérifier le comportement du récepteur AIS

5.4.2 Simulation sur banc de test

La description matérielle générée peut être vérifiée sous Vitis HLS par un banc de test programmé en C++. Le schéma du banc de test est présenté sur la Figure 5.9. Le banc de test utilise comme sources des signaux réels enregistrés dans la rade de Brest et vérifie que la latitude, la longitude et le MMSI sont correctement décodés par le récepteur. Ces signaux sont en bande de base à ± 25 kHz, contiennent deux messages séparés par du bruit et durent 1,1906 s. Le test est validé, ce qui montre que la synthèse matérielle générée par Vitis HLS est correcte.

Par ailleurs, nous construisons aussi un banc de test en VHDL pour permettre aux simulateurs NVC et GHDL de vérifier le code VHDL de notre récepteur. La vérification utilise aussi les signaux en bande de base enregistrés dans la rade de Brest et permet de valider la portabilité du code VHDL généré par Vitis HLS. Les traces des signaux échangés sont présentées sur la Figure 5.10. Le MMSI correspond à l'information `data_out` et sa valeur correspond à celle attendue pour valider le test. Sur la Figure, il apparaît que dès qu'un message est détecté, plus aucun échantillon n'est lu en entrée, car le récepteur traite les signaux du message. Les nouveaux échantillons arrivant, à la fréquence de 192 kHz, sont donc enregistrés en mémoire, en attendant que le traitement soit terminé. Pour assurer l'enregistrement de tous ces échantillons, la mémoire doit avoir une taille minimale de 2 ko. En effet, le temps d'exécution du récepteur pour traiter les signaux d'un message est de 2,68 ms et chaque échantillon est codé sur 32 bits. Il est tout à fait envisageable de construire un système avec cette quantité de mémoire.

5.4.3 Limites du banc de test

Néanmoins, ce banc de test reste très simple et ne permet pas de vérifier de manière exhaustive le comportement du récepteur. En effet, le récepteur est en interaction forte avec son environnement dont les conditions peuvent être très variables (changement de météo, présence d'île, de reliefs, arrivé, départ d'un bateau communicant, variation des CFO des transpondeurs). Pour toutes ces conditions, il convient de vérifier que le récepteur

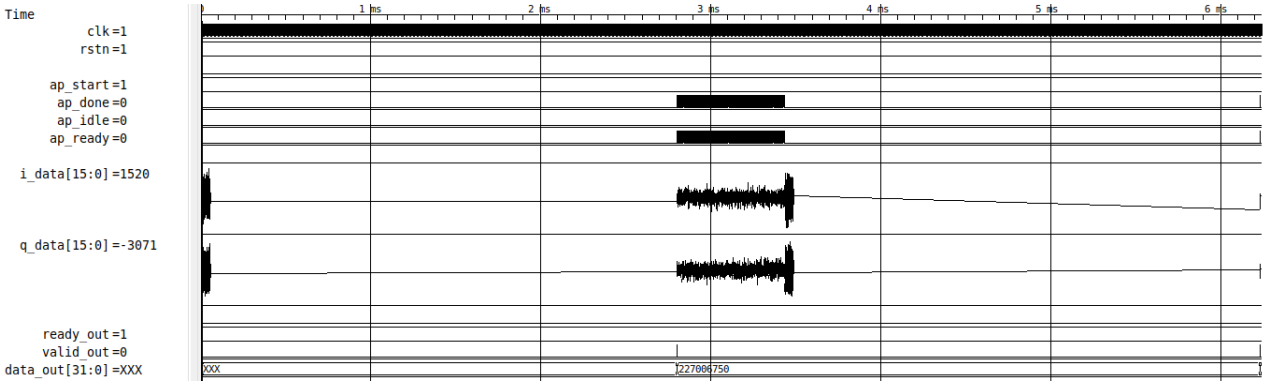

FIGURE 5.10 – Évolution des traces des signaux au cours du temps.

fonctionne correctement. Avec le banc de test que nous venons d'utiliser, le comportement du récepteur a été vérifié pour un seul niveau bruit, un seul CFO, une seule cartographie et un seul émetteur. Par ailleurs, il faut aussi vérifier que le récepteur est capable de détecter des falsifications de position. Pour cela, des scénarios de falsification de position doivent pour être générés pour vérifier la capacité à détecter les falsifications de position de notre récepteur. Ainsi, pour que l'étape de vérification soit rendue plus exhaustive, un environnement virtuel, capable d'être modifié pour tester une grande variété de scénarios et de conditions environnementales d'utilisation, doit être créé. C'est justement ce que nous allons présenter dans le prochain chapitre.

5.5 Conclusion

Dans ce chapitre, nous avons montré que la HLS est aujourd'hui une technologie mature pour générer automatiquement la description matérielle de systèmes complets et complexes. Ainsi, grâce à cet outil, la barrière qui séparait autrefois les activités de traitement du signal de celles de conception matérielle est en passe d'être effacée : un ingénieur en traitement du signal, avec des connaissances élémentaires en conception matérielle, peut maintenant maîtriser presque entièrement tout le processus de conception d'un système embarqué. Cet outil a été utilisé pour concevoir le récepteur AIS intelligent sur FPGA contenant la stratégie 1 que nous avions présentée dans le chapitre 1. Le récepteur ainsi conçu est capable d'extraire les informations contenues dans les signaux AIS en bande de base, et de détecter si les informations concernant la position ont été falsifiées. Ce récepteur peut encore être amélioré en intégrant de nouvelles stratégies par application de la HLS, notamment celles que nous avons développées dans les chapitres 3 et 2, et qui n'ont toujours pas été implémentées. L'intérêt d'utiliser un FPGA est qu'il offre une puissance de calcul élevé, beaucoup de ressources matérielles et est facilement reconfigurable, ce qui se prête bien à l'amélioration itérative et incrémentale de notre récepteur selon la philosophie de la méthode Agile. À la fin du chapitre, quelques critiques ont été émises

concernant le banc de test utilisé pour vérifier le comportement du récepteur. En effet, ce banc de test, utilise des signaux enregistrés dans la rade de Brest et ne permet pas d'effectuer une vérification exhaustive. C'est pourquoi, dans le prochain chapitre, nous allons proposer une nouvelle méthode de vérification du comportement du récepteur.

Chapitre 6

Vérification accélérée sur FPGA du récepteur AIS plongé dans son environnement : *laboratoire virtuel*

6.1 Introduction

Dans le chapitre précédent, nous avons conçu un récepteur AIS intelligent qui, à partir de signaux en bande de base, détecte les falsifications de position. Le comportement du récepteur a été vérifié avec succès à l'aide de signaux réels enregistrés dans la rade Brest. Cependant, cette vérification reste incomplète et insuffisante : nous ne savons pas, par exemple, à partir de quel niveau de bruit ou de quel offset de la fréquence porteuse les signaux ne sont plus décodés. C'est pourquoi, dans ce chapitre, nous proposons une nouvelle méthode de vérification de notre récepteur qui soit plus exhaustive. Nous cherchons à établir une méthode générale permettant de disposer d'un véritable *laboratoire virtuel* permettant de mettre au point de nouveaux algorithmes embarqués implémentant des stratégies de détection de falsification.

Cette méthode consiste à considérer notre récepteur AIS comme un élément d'un système plus vaste et à *co-modéliser* cet ensemble. Ce système plus vaste est constitué d'un environnement virtuel du récepteur, que l'on souhaite le plus réaliste possible et du récepteur lui-même. L'environnement virtuel contient des éléments informatico-électroniques, comme des émetteurs AIS, mais également des éléments plus proches de l'environnement physique et caractérisant les propagations des signaux électromagnétiques dans un environnement marin. Le système complet ainsi modélisé est un véritable *système cyber-physique* selon la définition donnée par E. Lee [82]. À partir de ce modèle, on souhaite exciter l'ensemble des composants de manière simultanée, en simulation, afin de vérifier le comportement du récepteur en son sein pour différentes conditions d'utilisation.

La simulation de l'ensemble est effectuée sur un FPGA, et l'implémentation du FPGA est générée automatiquement par la HLS à partir de la modélisation du CPS dans un langage haut-niveau. Dans ce cas d'utilisation, la HLS sert de *synthétiseur de bancs de test*. Il s'agit d'une nouveauté dans le domaine de la vérification de système sur FPGA, car jusqu'à présent la plupart des bancs de test sont écrits à la main directement au niveau

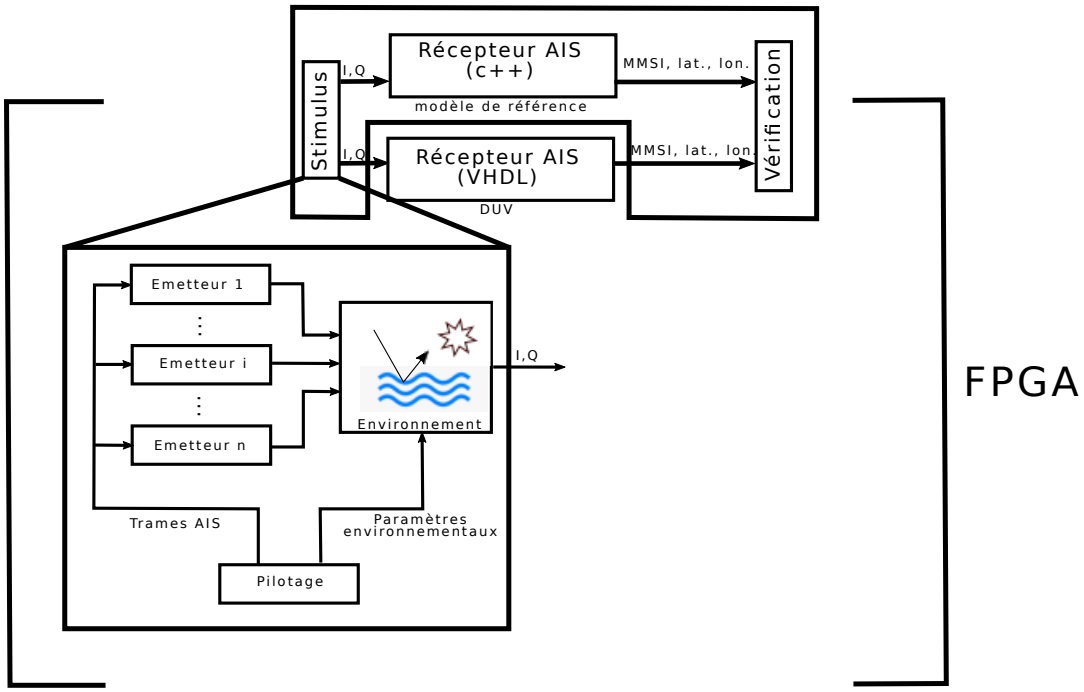

FIGURE 6.1 – Architecture du laboratoire virtuel de simulation du CPS sur FPGA.

VHDL. Un schéma de ce simulateur de CPS sur FPGA est présenté sur la Figure 6.1. Sur ce schéma, il apparaît que, contrairement à la vérification effectuée dans le chapitre 5, les stimulus de vérification ne sont pas déjà pré-enregistrés, mais générés grâce au modèle de l'environnement physique du récepteur.

6.2 Laboratoire virtuel de simulation de CPS sur FPGA

6.2.1 Présentation

Dans cette étude, nous suivons la définition de CPS donnée par E. Lee [82] qui définit un CPS comme un système contenant un ou des systèmes embarqués en plus de l'environnement physique dans lequel ils intéragissent. Cette définition se différencie de celle rappelée dans [67] qui définit les CPS comme des systèmes embarqués en interaction étroite avec leur environnement. Modéliser et simuler un CPS est, encore aujourd'hui, très difficile à effectuer et sujet à de nombreuses erreurs. L'hétérogénéité intrinsèque des composants constitutifs du système [40] (des unités de calcul, un réseau de communication, un environnement physique, etc) requiert, a priori, des modèles de calcul et des plateformes de simulation différents, qu'il est difficile de composer pour modéliser et simuler le système complet. Partant de ce constat, nous proposons, comme défendu dans l'état de l'art présenté dans l'Introduction, une modélisation de l'ensemble du CPS sous forme d'un réseau d'acteurs interagissant entre eux selon un modèle de calcul unique (Kahn process network) [83], et une simulation de ce modèle sur une seule plateforme, le FPGA. Notons

que dans notre acception, le terme *acteur* se réfère à des entités de calculs opérant en parallèle, sans précision quant à leur mode de synchronisation. Ces acteurs n'agissent pas selon un MOC purement *dataflow*, comme cela est programmé par des langages comme CAL [38].

Sachant qu'un CPS est composé de plusieurs composants interagissant entre eux de manière indépendante et simultanée, il est tout à fait naturel d'appliquer une modélisation à base d'acteurs. Par ailleurs, la simulation de ce type de modélisation, intrinsèquement parallèle, n'est absolument pas naturelle à l'aide d'ordinateurs standards. C'est pourquoi, nous avons décidé d'utiliser une compilation sur des FPGA comme solution, car ces circuits présentent une forme de "parallélisme ultime" qui s'adapte parfaitement à l'architecture du modèle à base d'acteurs. Par ailleurs, le FPGA offre une puissance de calcul élevée, beaucoup de ressources matérielles et est remarquablement scalable : ceci rend envisageable la simulation de CPS de très grande taille. Ces caractéristiques de simulation ne sont pas partagées par les autres plateformes de simulation, telles que les plateformes multicœurs et GPU comme montré dans [111, 149].

Ainsi, l'idée défendue dans ce manuscrit est de simuler l'ensemble du CPS sur un même FPGA pour vérifier le comportement de notre récepteur AIS. Cette idée équivaut à la conception d'un véritable laboratoire virtuel de simulation de CPS. Un exemple de ce laboratoire de simulation est présenté sur la Figure 6.2. Sur cette Figure apparaissent deux *Emetteurs* de signaux AIS, un *Environnement* maritime qui, en fonction de l'environnement simulé, modifie ces signaux, et le *Récepteur* AIS que nous concevons. En plus, les composants *Pilotage* et *Observateur* permettent respectivement de modifier les scénarios testés durant la vérification et d'observer les signaux échangés. L'architecture matérielle du *Récepteur* est reproduite sur le FPGA et forme un véritable prototype, d'où l'idée de laboratoire sur FPGA. Le FPGA joue ainsi à la fois le rôle de simulateur de l'environnement et d'émulateur matériel du dispositif embarqué final. On définit, en se basant sur [90], l'émulation comme la reproduction du comportement d'un modèle dont toutes les variables sont connues et la simulation comme la reproduction du comportement d'un modèle mais en devant extrapoler une partie des variables qui lui sont inconnues.

6.2.2 État de l'art de la simulation de CPS

Les premiers chercheurs à avoir proposé d'utiliser les FPGA comme émulateur d'environnement physique sont Miller et Givargis en 2010. Le FPGA était utilisé pour émuler le fonctionnement d'un poumon artificiel [108]. Le rôle du FPGA était alors de permettre le test et la mise au point de ventilateurs externes : le FPGA jouait le rôle d'un organe artificiel, modélisé par un système d'équations différentielles établi avec des spécialistes de la respiration. Il devenait facile de placer ce poumon artificiel dans des situations complexes ou difficiles à gérer pour le ventilateur. Malgré ces travaux remarquables, la

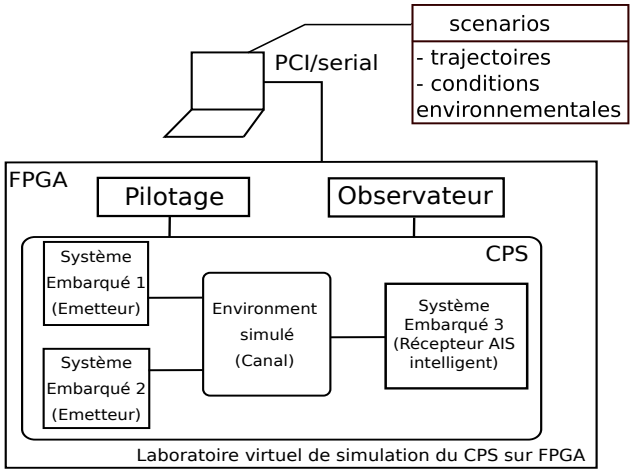

FIGURE 6.2 – Schéma du laboratoire virtuel de simulation de CPS sur FPGA.

plupart des émulations d'environnement se font encore aujourd'hui sur GPU (Graphics processing unit) ou CPU (Central processing unit). Néanmoins, quelques rares travaux utilisant le FPGA comme plateforme d'émulation d'environnement ont été présentés. Ils concernent principalement l'émulation de machines ou de réseaux électriques [32, 111] ; il existe aussi d'autres cas d'application, comme l'émulation de réactions biochimiques [167] ou de communications radio [109, 107]. Ces émulations d'environnement sont utilisées presque exclusivement dans le cadre d'une simulation du type HIL ou co-simulation, c'est-à-dire que l'ensemble du CPS est simulé, au minimum, sur deux plateformes différentes. Dans l'article [56] c'est seulement le système embarqué qui est émulé sur un FPGA. Néanmoins, ces travaux se différencient de la méthode que nous défendons dans ce manuscrit, et qui vise à simuler le CPS en entier sur un même FPGA. En effet, comme nous l'avons expliqué dans l'introduction (partie), composer plusieurs plate-formes pour émuler un CPS est une tâche difficile, sujette à des erreurs de synchronisation qui peuvent provoquer des artefacts temporels et détériorer la précision de l'émulation. Un des rares cas, à notre connaissance, simulant un CPS en entier sur un FPGA a été proposé dans [141]. Cependant, la méthode utilisée exécute des outils qui ne sont pas en libre accès, comme ceux que nous utilisons, ce qui limite la reproductibilité de la méthode.

6.2.3 Méthode de conception

La méthode appliquée pour concevoir le laboratoire de simulation de CPS est présentée sur la Figure 6.3. L'ensemble du CPS est modélisé au niveau système et divisé en composants : l'un d'entre eux simule l'environnement (partie physique) et les autres émulent les systèmes embarqués (partie cyber), comme présenté sur la Figure 6.3. La description des composants se fait en C++ avec des variables à virgule fixe, et utilise les directives proposées par Vitis HLS [157] pour gérer leurs interactions. Ces directives permettent d'exprimer la concurrence et les communications entre composants. Par exemple,

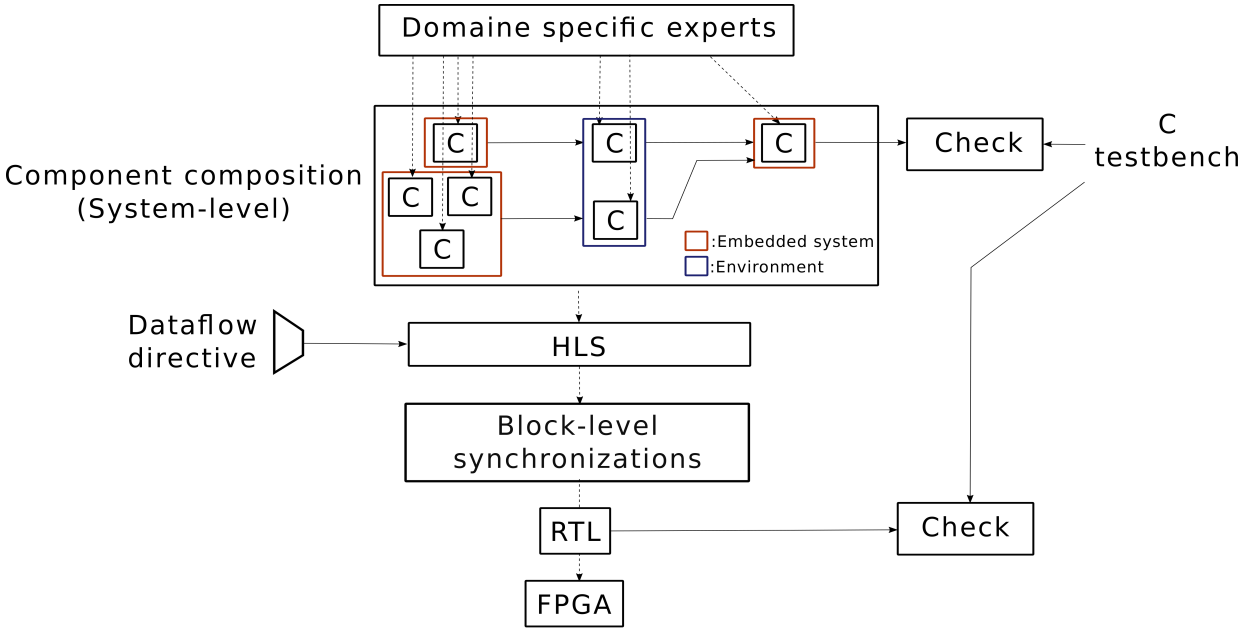

FIGURE 6.3 – Schéma représentant le flot de conception permettant de concevoir un laboratoire sur FPGA.

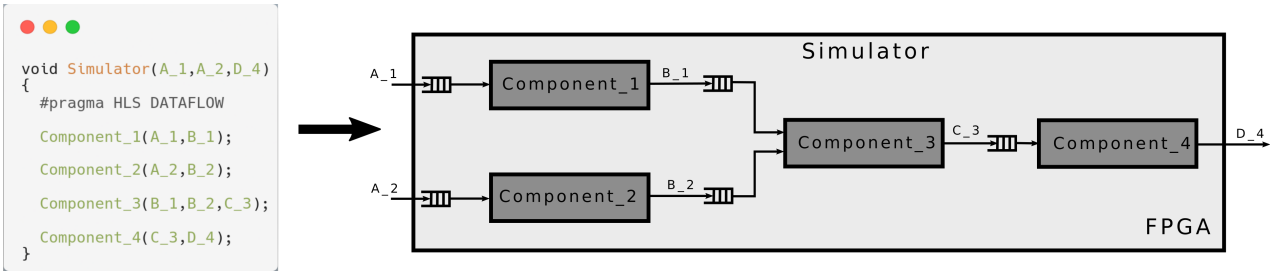

FIGURE 6.4 – Exemple de synthèse d'un modèle en C++ utilisant la directive HLS dataflow

la directive *dataflow* permet l'exécution parallèle des composants. Les directives de communication permettent de gérer les interfaces entre composants et de mettre en place différents protocoles de communication. L'utilisateur peut appliquer une lecture ou une écriture bloquante et utiliser des mémoires tampons FIFO ou PIPO (Ping-pong), ou des mémoires vives entre les composants. Par l'utilisation de ces directives, l'utilisateur a la possibilité d'appliquer le MOC KPN. Un exemple de synthèse du code d'une modélisation à base d'acteurs utilisant la directive *dataflow* est présenté sur la Figure 6.4.

Le MOC KPN utilisé pour la modélisation a la propriété intéressante d'avoir un comportement déterministe qui ne dépend pas des caractéristiques temporelles de ses processus et de ses canaux de communication. Ainsi, après la synthèse, il est garanti que les fonctionnalités du modèle restent les mêmes que celles de la description originale. Avec ce MOC, les opérations ont été ordonnées, mais non planifiées avec une référence temporelle. Par ordonné, nous entendons que nous savons lorsqu'une opération commence et se termine par rapport aux moments où les autres opérations commencent et se terminent. C'est l'opération de synthèse de haut niveau qui va ensuite affiner cela en ajoutant une

notion explicite de temps et nous amener jusqu'à un modèle précis en termes de cycles d'horloge.

La vérification du comportement du CPS se fait à plusieurs niveaux d'abstraction : au niveau d'abstraction système, à partir de la modélisation en C++, ou au niveau d'abstraction RTL produit par la synthèse HLS. L'intérêt de commencer la vérification avec un haut niveau d'abstraction est de détecter les erreurs de fonctionnement du système tôt dans le flot de conception pour les corriger rapidement : les corrections sont alors peu coûteuses. Durant les premières phases de vérifications, les variables utilisées peuvent être des variables à virgule flottante et le temps n'est pas défini explicitement puisque le MOC KPN est appliqué. Ensuite, le niveau d'abstraction peut être diminué pour se rapprocher de la description matérielle du système. Les variables sont transformées en variables à virgule fixe avec un nombre de bits fixé par l'utilisation des bibliothèques `ap_fixed` et `ap_ufixed`, et les protocoles de communication sont définis, avec la possibilité d'appliquer des lectures ou écritures bloquantes et d'utiliser des FIFO ou PIPO. Enfin, la synthèse du simulateur par la HLS est effectuée, ce qui permet d'obtenir une description du système au niveau RTL avec la considération du temps au cycle d'horloge près. Une autre vérification, écrite en C++, peut ensuite être effectuée pour vérifier cette description matérielle. Cette dernière vérification nécessite une puissance de calcul importante, car le niveau de détail sur l'architecture du système est plus élevé, en particulier à cause de la considération du temps. La vérification s'effectue donc sur FPGA pour assurer une puissance de calcul suffisante. Cette méthode de vérification *top-down* est celle défendue par la conception ESL [101] et fait partie de la conception *modèle based design*, présentée dans l'état de l'art de ce manuscrit et opposée à la conception en cascade. Ce processus de vérification est présenté sur la Figure 6.3.

6.3 Application de la méthode au récepteur AIS intelligent

6.3.1 Architecture du CPS

L'architecture de l'ensemble du CPS qui est simulé sur un FPGA est présentée sur la Figure 6.5. Sur cette Figure, plusieurs composants apparaissent : un *Récepteur* dont le modèle a été présenté dans le chapitre précédent, un *Environnement*, un *Emetteur*, un *Pilotage* et un *Observateur*.

Le composant *Emetteur* contient huit blocs. En entrée, ce composant reçoit une trame contenant 168 bits, et en sortie, il émet des signaux en quadrature (I,Q). Le premier bloc lit la trame NMEA. Le bloc suivant transforme cette trame conformément au protocole HDLC : chaque octet est inversé, le CRC est calculé et ajouté à la fin de la trame, un bourrage de bits à 0 est appliqué et un drapeau de début et de fin sont ajoutés. Ensuite,

une séquence de conditionnement est ajoutée au début de la trame. Puis, un codage sans retour à zéro inversé (NRZI) et une modulation GMSK sont appliqués. Ensuite, le signal est transposé en fréquence à la fréquence 25 kHz ou −25 kHz. Enfin, le signal est amplifié pour être émis avec une puissance de 12,5 W, conformément à ce qu'impose la norme AIS. Le signal n'est pas transposé à la fréquence des porteuses (161,975 MHz ou 162,025 MHz), car considérer le signal à cette fréquence aurait demandé une mémoire trop grande. C'est justement la raison pour laquelle le signal est considéré en bande de base à la fréquence ±25 kHz.

Le composant *Environnement* contient quatre blocs et permet d'appliquer des effets environnementaux variés sur les signaux envoyés entre l'émetteur et le récepteur. Un premier bloc ajoute un offset en fréquence au signal pour caractériser les erreurs sur les fréquences porteuses de l'émetteur et du récepteur causées par des imperfections matérielles. Ces erreurs ont été appelées CFO dans le chapitre 3. Un autre bloc reproduit l'affaiblissement de parcours de la puissance du signal dû à la distance entre l'émetteur et le récepteur. Un troisième bloc ajoute un bruit blanc gaussien au signal pour caractériser le bruit de propagation. L'amplitude du bruit est fixée en fonction du rapport signal/bruit (SNR) que l'on veut imposer. Enfin, un dernier bloc additionne les signaux de chaque émetteur.

Pour calculer l'atténuation du signal, le modèle de Friis est appliqué. Il stipule que le rapport entre la puissance du signal reçu et la puissance du signal émis est donné par :

$$\frac{P_r}{P_t} = G_r G_t \left(\frac{\lambda}{4\pi R}\right)^2 \tag{6.1}$$

où P_r et P_t sont respectivement les puissances reçues et émises. G_t et G_r sont respectivement les gains d'antenne en émission et en réception. λ la longueur d'onde de la fréquence de travail et R la distance séparant l'émetteur du récepteur. Nous fixons le gain des antennes à $G_t = G_r = 2$ dBi, comme suggéré par [105], et la puissance de l'émetteur est $P_r = 12{,}5$ W comme défini par la norme AIS. La référence [105] montre que ce modèle est fiable et reproduit correctement l'évolution de l'énergie du signal jusqu'à trente kilomètres. Ce modèle suppose que les signaux se déplacent en ligne directe de l'émetteur vers le récepteur, ce qui est tout à fait valable pour un milieu marin. Un modèle plus complexe pourrait être substitué ici pour mieux tenir compte des possibles réflexions sur l'eau ou l'atmosphère et de la forme de l'atténuation du signal transmis au-delà de 30 km.

Sur la Figure 6.5 apparaît aussi deux composants spécifiques à la simulation, un composant *Pilotage* et un composant *Observateur*. Le composant *Pilotage* pilote l'environnement maritime simulé (SNR, CFO) et gère, à partir d'un réglage initial, effectué à l'initialisation du simulateur, l'évolution des données dynamiques de chaque navire au cours de la simulation. Avec ces données et les informations statiques, ce composant crée les trames NMEA transmises à chaque *Émetteur*. L'évolution des données dynamiques

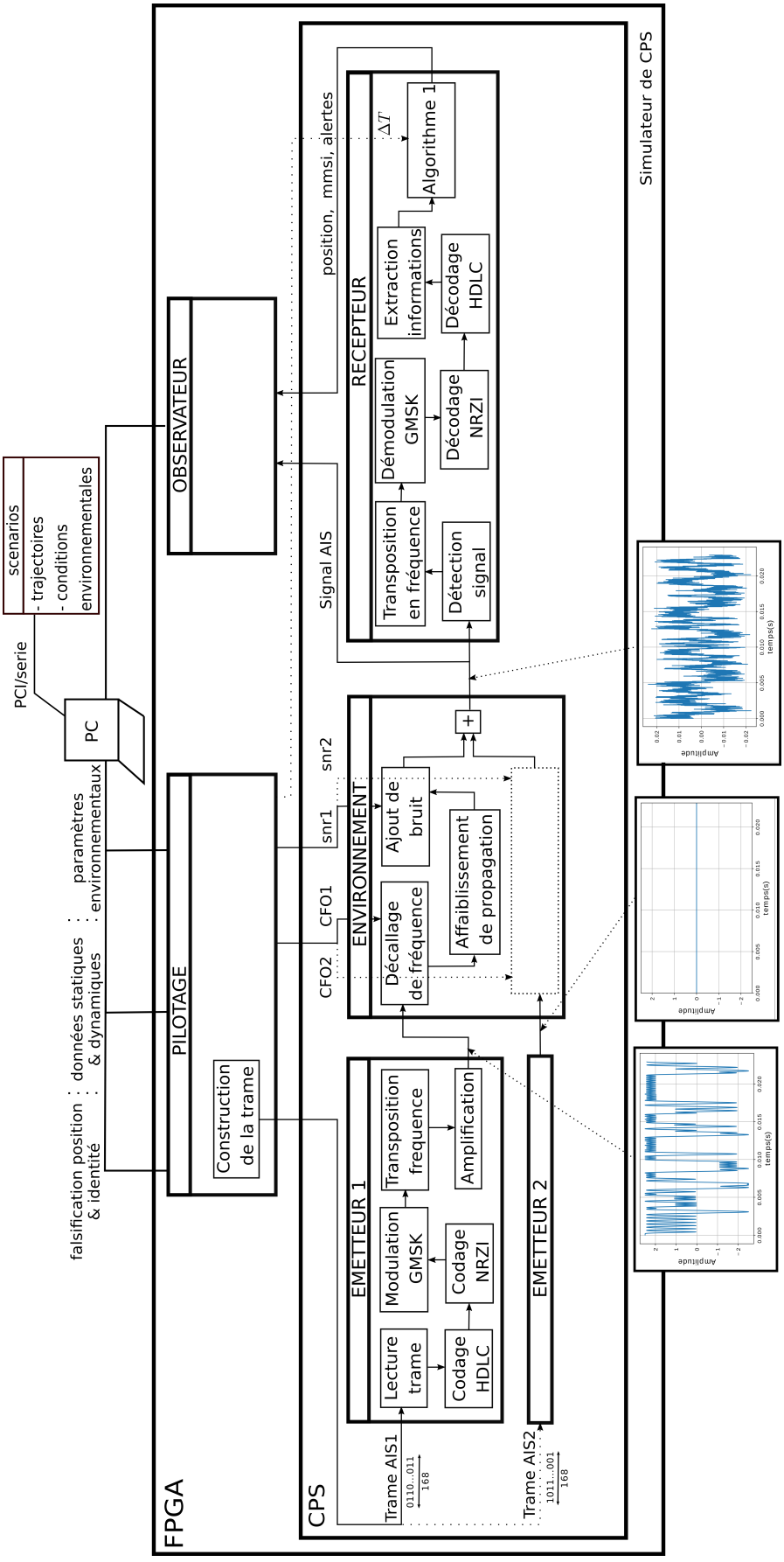

FIGURE 6.5 – Architecture détaillée du simulateur du CPS

nécessite la considération du temps. Pour cela, la période d'émission des messages qui dépend de la vitesse du bateau, comme présenté sur le Tableau 2.1 du chapitre 2, est considérée. Cette période d'émission (ΔT) est précise à ± 20 %, comme spécifié par la norme AIS. C'est pourquoi un Tableau contenant des valeurs tirées aléatoirement dans l'intervalle $[\Delta T - \frac{20\times\Delta T}{100} ; \Delta T + \frac{20\times\Delta T}{100}]$ est construit. Pour chaque nouveau message créé, un intervalle de temps est pioché dans ce Tableau et nous permet de déduire, en considérant la position, la vitesse et la route du bateau, sa nouvelle position. Par ailleurs, cet intervalle de temps (ΔT) est transmis au récepteur, car le filtre IMM de l'Agorithme 1 a besoin de cette donnée pour s'exécuter. Un bruit d'écart-type 5 m est ajouté à chaque position calculée pour reproduire l'erreur de mesure des GPS. Pour finir, ce composant offre la possibilité de créer des falsifications en fonction du scénario que l'on souhaite tester. Ces falsifications peuvent être l'émission d'une fausse position ou d'une fausse identité. Le composant *Observateur* permet quant à lui d'observer les signaux échangés entre l'*Emetteur*, l'*Environnement* et le *Récepteur*.

Sur la Figure de l'architecture détaillée, nous affichons le signal en phase (I) sortant de chaque *Emetteur*. Pour l'*Emetteur 2* ce signal I est nul car un seul émetteur émet à la fois, grâce au mode d'accès TDMA, pour empêcher qu'il y ait des interférences. Par ailleurs, pour améliorer l'illustration et faciliter la compréhension du rôle des blocs du composant *Environnement*, le signal est présenté en bande de base avec une fréquence centrale de 0 Hz. Cela veut dire que pour l'illustration le bloc *"Transposition fréquence"* du composant *Emetteur* n'a pas été appliqué. Le signal sortant du composant *Environnement* apparaît effectivement atténué, bruité et décalé en fréquence. Pour l'illustration, la distance entre l'émetteur 1 et le récepteur était de 4000 m, le SNR était fixé à 10 dB et le CFO à 500 Hz. Par ailleurs, il faut noter que l'architecture est modélisée par un graphe orienté acyclique : le composant *Pilotage* est situé au début du graphe et le composant *Observateur* à la fin. Cela permet grâce à la directive *dataflow* de Vitis HLS de générer la description matérielle avec une exécution parallèle des composants.

6.3.2 Synthèse matérielle

Nous avons expérimenté notre méthode de conception présentée sur la Figure 6.3. L'ensemble du simulateur du CPS représente environ 5000 lignes de code C++. Notez que certaines parties du système sont dédiées à la simulation pure, tandis que certains éléments (comme notre *Recepteur*) peuvent être considérés comme un prototype viable d'un futur dispositif embarqué. Les résultats de la synthèse du simulateur et de ses composants sont présentés sur le Tableau 6.1. Le pourcentage d'occupation d'un simulateur avec les trois composants du CPS plus le composant *Pilotage* est faible (moins de 17 % pour les éléments BRAM (Bloc RAM), DSP (Digital signal processing), FF (Flip-flop), et LUT).

Cependant, le FPGA n'est pas pour autant sur-dimensionné. En effet, les bancs de

test doivent simuler des environnements synthétiques avec plusieurs *Emitters* se déplaçant physiquement et échangeant des messages AIS en même temps. Déjà, un scénario, présenté dans la section 6.3.3, avec dix *Emitters* a été testé, et le taux d'occupation a atteint presque 37 %. En outre, l'environnement simulé était trop simple pour reproduire tous les scénarios d'usurpation et de falsification. L'environnement doit avoir la possibilité de représenter différents environnements marins, comme la présence d'îles ou de côtes, car ces éléments sont utilisés par certains bateaux pour falsifier leur identité sans être repérés. La simulation de tels environnements nécessite des capacités de calcul élevées. Pour finir, les stratégies 2 et 3 doivent encore être ajoutées au *Récepteur*. Ces stratégies augmenteront de quelques pourcents le taux d'occupation du FPGA, d'autant plus que la stratégie 2 applique une FFT. Notez que les ressources utilisées par le composant *Observateur* ne sont pas affichées sur le Tableau 6.1 ; elles sont très faibles et peuvent donc être négligées.

Tableau 6.1 – Ressources utilisées par chaque composant et latence associée

	BRAM	**DSP**	**FF**	**LUT**	**Latence (ms)**
Simulateur	**259** (6 %)	**302** (3 %)	**139 730** (5 %)	**225 196** (17 %)	**3,27** -
Émetteur	97	1	8371	16 143	$5{,}97 \times 10^{-1}$
Environnement	98	13	2824	7150	$2{,}07 \times 10^{-1}$
Récepteur	32	278	125 003	194 122	2,68
Pilotage	0	10	2569	6967	$1{,}32 \times 10^{-3}$

6.3.3 Exploitation du Simulateur

Grâce à notre *laboratoire virtuel*, les performances de notre récepteur ont été évaluées sur des conditions environnementales variées. Par exemple, pour plusieurs valeurs de SNR et de CFO, le taux d'erreur binaire de la démodulation GMSK de notre récepteur est calculé. Les résultats obtenus sont présentés sur les courbes de la Figure 6.6. Les courbes sont tracées pour un CFO de 0 Hz, 200 Hz et 400 Hz. Si nous avions pris des valeurs négatives pour le CFO, nous aurions obtenu les mêmes courbes. Les performances obtenues sont similaires à celles d'autres démodulateurs GMSK rencontrés dans la littérature [6], et permettent de définir les conditions d'utilisation limites de notre *Récepteur*.

Par ailleurs, pour illustrer la capacité de notre simulateur à générer des trajectoires de navires pouvant contenir des messages falsifiés ou usurpés (mais aussi pour prouver le fonctionnement de notre modèle de canal basé sur le modèle de Friis), cinq scénarios ont été créés et observés, et sont présentés sur la Figure 6.7. Parmi ces scénarios, trois d'entre eux reproduisent une trajectoire de navire dont les messages reçus ne contiennent aucune information falsifiée, et pour les deux autres la position a été falsifiée. Le navire 4 reproduit le type d'usurpation signalé dans [10] et le navire 5 reproduit le type de falsification signalé dans [72]. Dans cette dernière référence, un bateau de pêche falsifie sa

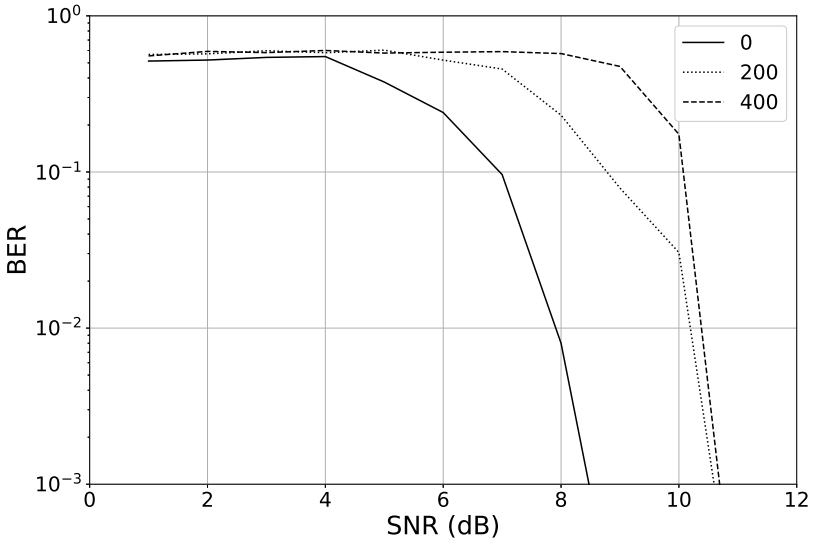

FIGURE 6.6 – Évolution du BER en fonction du SNR et du CFO.

position pour pêcher dans une zone restreinte. Pour toutes ces trajectoires, le composant *Environnement* a ajouté un bruit blanc gaussien au signal pour atteindre un SNR égal à 10 dB. Pour simuler ces cinq scénarios en même temps, cinq composants *Emetteur* sont émulés sur le FPGA, en plus d'un composant *Environnement*, *Récepteur*, *Pilotage* et *Observateur*. Il apparaît aussi, sur la Figure, qu'à partir d'une certaine distance du récepteur, les signaux émis par les bateaux ne sont plus reçus.

6.3.4 Performances de simulation

Nous comparons, dans le Tableau 6.2, le temps de simulation logicielle au temps de simulation sur FPGA lorsque 1, 2, 5 et 10 bateaux sont simulés en même temps. La simulation avec 5 émetteurs correspond au scénario présenté sur la Figure 6.7. La simulation logicielle se fait sur un processeur Intel Core I5 standard à 1,7 GHz avec 16 Gb de RAM. La simulation matérielle se fait sur le FPGA Virtex UltraScale+ HBM VCU128 présenté ci-dessus. Au cours de chaque simulation, 1000 messages sont reçus par le *Recepteur* depuis les *Emetteurs* qui émettent successivement. Pour exécuter ces simulations, des composants *Emitter* ont été ajoutés ou retirés de la plateforme FPGA, c'est pourquoi le taux d'utilisation du FPGA change. En considérant les résultats de ces trois simulations, nous prédisons que le nombre maximum d'*Emitters* pouvant être émulés, en même temps, sur le FPGA, est de 40. Dans ce cas, le pourcentage d'occupation de la BRAM atteint 97 %.

Le gain de performance est déjà de $\times 587$ par rapport à la simulation logicielle pour un seul *Emitter*. Le gain a été obtenu sans appliquer de directives d'optimisation au code telles que le déroulage de boucle, de pipeline, ou la partition de tableau, ce qui laisse

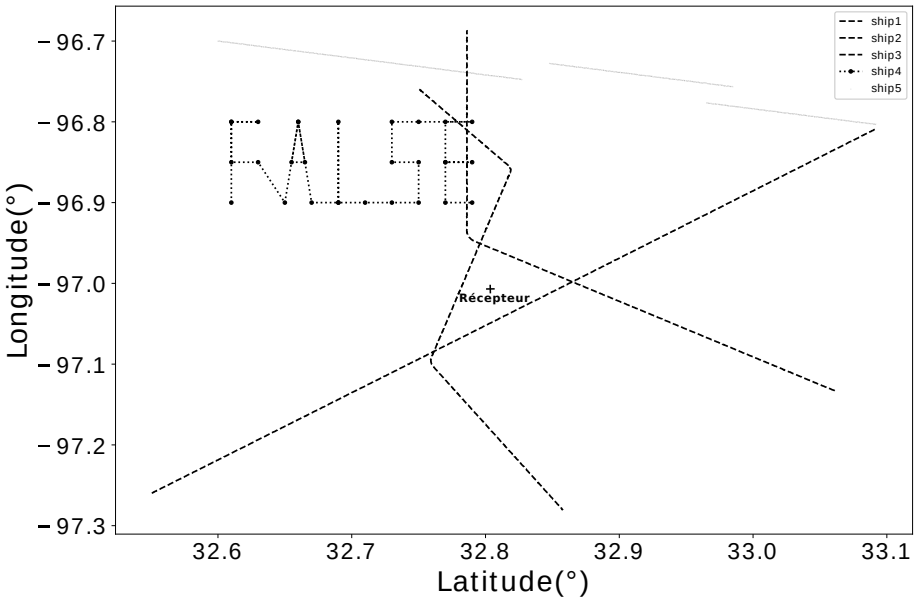

FIGURE 6.7 – Trajectoires simulées sur le FPGA

TABLEAU 6.2 – Temps de simulation matérielle et logicielle

Nb. Emitters	**Soft.**	**Hard.**	**Gain**	**FPGA usage**
1	1920 s	3,27 s	587	17 %
2	2010 s	3,27 s	614	19 %
5	2280 s	3,27 s	697	27 %
10	2730 s	3,27 s	834	37 %
40 (pred.)	5430 s	3,27 s	1660	97 %

encore une marge d'accélération pour la simulation sur FPGA. Seule la directive dataflow est appliquée au code, comme indiqué dans la partie méthodologie 6.2.3, pour permettre l'exécution parallèle des composants. L'accélération obtenue grâce à la simulation sur FPGA réduit le temps de simulation de plusieurs dizaines de minutes à seulement quelques secondes. De plus, bien que le nombre de composants *Emitter* ait augmenté, le temps de simulation matérielle est resté le même : les composants *Emitter* ont été exécutés en parallèle. C'est pourquoi pour 40 *Emitter* le gain de performance est prédit à $\times 1660$. Ce fait montre l'intérêt de l'aspect scalable des FPGAs. Néanmoins, puisque la majorité du temps d'exécution est occupé par le *Récepteur* le gain augmente doucement à mesure que le nombre d'*Emetteur* augmente.

L'utilisation de variables à virgule fixe introduit des erreurs qui influencent la précision des positions générées. Une analyse statistique sur 1000 valeurs montre que ces erreurs suivent une distribution gaussienne centrée avec un écart-type de $\sigma_{lat} = 0{,}32$ m pour la latitude et de $\sigma_{lon} = 0{,}37$ m pour la longitude. Ces écart-types sont négligeables par rapport à l'écart-type d'un GPS (5 m), qui est justement le capteur utilisé par l'AIS pour

connaître la position des bateaux. Cela valide la capacité de notre simulateur à simuler précisément des trajectoires falsifiées ou usurpées de bateaux.

6.4 Discussion et Conclusion

6.4.1 Discussion

Généralement, la connaissance précise, durant la simulation, des instants d'exécution des évènements du système embarqué a une importance cruciale. En effet, un système embarqué interagit en permanence avec des processus physiques qui évoluent de manière continue. Pour considérer le temps dans les modulations, des MOC temporisés ont été proposés [83]. Parmi ce type de MOC, on compte le MOC à évènement discret (DE (Discret event)) [168], qui est un réseau d'acteurs où chaque acteur réagit aux événements d'entrée et produit des événements de sortie dans l'ordre de l'horodatage. On compte aussi le MOC à temps continu (C.T. (Continuous time)) qui, pour être exécuté sur un ordinateur numérique, utilise un solveur qui approxime son exécution. Un solveur bien connu est le solveur d'Euler. Un MOC C.T. peut être considéré comme un modèle SR (Synchrone-reactif) avec un pas de temps entre les réactions globales déterminées par le solveur. Dans notre cas d'application, pour la simulation de l'environnement maritime du récepteur, nous nous sommes contentés d'utiliser le MOC KPN car, sachant que l'environnement maritime à simuler ne demandait pas de résoudre d'équations différentielles temporelles, ce type de MOC suffisait. Il a tout de même fallu considérer le temps pour la génération des trajectoires des *Emetteurs* et l'exécution du filtre IMM. Pour cela, le temps était transmis comme une donnée entre composants. Il est fort probable que pour les prochaines mises à jour de notre simulateur, qui permettront d'appliquer des modèles plus riches et complets pour modéliser l'environnement, il soit nécessaire de considérer un MOC temporisé.

6.4.2 Conclusion

Dans ce chapitre, nous avons proposé une méthode concevant un laboratoire virtuel de simulation de CPS sur FPGA. Ce laboratoire a été utilisé pour vérifier le comportement du récepteur AIS, et permet de simuler son environnement maritime et de générer des signaux AIS synthétiques transmis au récepteur. Par la modification de cet environnement maritime, une grande variété de scénarios de test ont été générés et ont permis de connaître les conditions limites d'utilisation du récepteur. En effet, avec ce laboratoire, l'utilisateur peut fixer le nombre de navires transmettant des signaux au récepteur, le niveau de bruit et le CFO des signaux transmis. Par ailleurs, il peut aussi générer des scénarios de falsification de position, vitesse et identité. La conception d'un tel simulateur a été rendue possible grâce à la HLS qui est utilisée pour synthétiser automatiquement ces bancs de

test sur FPGA. Par rapport à une simulation purement logicielle, nous montrons des gains d'accélération remarquables (de $\times 587$ à $\times 1660$).

Conclusion générale et perspectives

L'AIS est un système de communication pour bateaux très répandu qui permet l'échange automatique d'informations de navigation. Ce système, développé dans les années 90, est peu sécurisé, car à cette époque, les cyberattaques étaient rares et peu sophistiquées. Par conséquent, des utilisateurs malveillants peuvent facilement le manipuler pour émettre de fausses informations, brouiller les communications ou même créer des bateaux fantômes. Ces pratiques, de plus en plus courantes, peuvent cacher des activités frauduleuses, perturber le trafic ou justifier des prétendues "*violations de territoire*". Sachant l'utilisation grandissante de ce système (500 000 navires l'utilisaient en 2021), il est plus que nécessaire de le sécuriser.

Objectifs et problématiques

Partant de ce constat, l'objectif des travaux de thèse a été de concevoir un récepteur AIS intelligent, capable de détecter, en temps réel, les falsifications de position et d'identité, et l'émission de faux messages. Pour détecter ces falsifications, des stratégies ont été développées et intégrées au récepteur et avaient comme contrainte d'être d'une complexité limitée pour pouvoir être exécutées en temps réel. Par ailleurs, la conception du récepteur sur un système embarqué de type FPGA a fait surgir d'autres difficultés, car la conception devait être rapide (3 ans de thèse), bon marché et accessible pour un doctorant sortant d'une formation en traitement du signal. Pour finir, du fait de l'interaction étroite du récepteur avec son environnement, ces interactions devaient être considérées avec attention durant l'étape de vérification pour assurer un comportement fiable du système dans son environnement réel. Cela nous a amené à développer une nouvelle méthode de conception permettant de modéliser et simuler un système cyber-physique.

Contributions

Les contributions de ce manuscrit peuvent se séparer en deux parties. La première partie concerne le développement de stratégies détectant les manipulations frauduleuses de l'AIS et la deuxième partie concerne la conception sur FPGA du récepteur AIS intelligent.

Développement des stratégies détectant les manipulations frauduleuses de l'AIS

Pour détecter les manipulations frauduleuses de l'AIS, trois stratégies ont été développées.

La première stratégie, présentée dans le chapitre 1, détecte les falsifications de position et de vitesse en pistant la position de chaque bateau avec un filtre IMM. Ce filtre applique deux modèles qui caractérisent la dynamique d'un navire utilisant un AIS de classe A (jauge brute supérieure à 300 tonneaux). Avec ces modèles, le filtre prédit, pour la position et la vitesse, une zone de validité, avec un niveau de confiance fixé, dans laquelle la prochaine mesure doit se trouver. Si la mesure de position ou de vitesse se trouve en dehors de cette zone, c'est qu'elle a été falsifiée. Nous avons évalué la sensibilité de cette stratégie par simulations de Monte Carlo. Les simulations ont montré que la zone de validité variait de 35 m à 250 m pour la position et de 4 kn à 8,5 kn pour la vitesse. Aussi, la stratégie a été appliquée à 100 000 messages enregistrés dans la rade de Brest et les résultats obtenus ont montré son utilité et son efficacité pour détecter ce type de falsifications.

La deuxième stratégie, présentée dans le chapitre 2, vérifie que le mode d'accès TDMA est respecté par les bateaux pour communiquer. Cette stratégie détecte les faux messages et les bateaux fantômes. Le mode d'accès TDMA est complexe et très souvent non respecté lorsque de faux messages sont émis. C'est justement ce que nous avons observé pour un bateau fantôme détecté par l'OTAN en mer Noire. Ainsi, pour chaque bateau contrôlé, le pourcentage de messages transmis ne respectant pas ce mode d'accès est calculé ; dès qu'il dépasse un certain seuil, fixé par exploitation de résultats obtenus suite à l'application de la stratégie sur 100 000 messages (80 %), une alarme est envoyée.

La troisième stratégie, présentée dans le chapitre 3, calcule, pour chaque bateau, une signature radiométrique, à partir de leurs signaux transmis. Cela permet d'identifier les bateaux par la signature radiométrique de leur transpondeur et non pas par l'identité (numéro MMSI) qu'ils transmettent dans leurs messages. Cette stratégie permet de détecter les falsifications d'identité et utilise, comme signature radiométrique, l'offset en fréquence (CFO). L'offset en fréquence peut, en fonction des transpondeurs, dériver au cours du temps et être plus ou moins bruité. C'est pourquoi, un filtre de Kalman adaptatif, pistant cette signature pour chaque transpondeur, est utilisé pour s'adapter automatiquement à leur dérives et leur bruits. Ce filtre adaptatif crée une zone de validité des mesures d'offset, comme pour le filtre IMM avec la position et la vitesse. Si l'offset en fréquence mesuré est en dehors de cette zone, une alarme est renvoyée signifiant que l'identité du bateau a été falsifiée.

Pour finir, ces trois stratégies sont regroupées dans une stratégie globale pour être exécutées conjointement. La synergie de ces stratégies permet d'améliorer l'efficacité de

la stratégie détectant les falsifications d'identité. Une expérimentation appliquée à des signaux réels enregistrés dans la rade de Brest a montré une division par trois des taux d'erreur de première et de deuxième espèce sur le test détectant les falsifications d'identité. Cette contribution est présentée dans le chapitre 4.

Conception du récepteur AIS intelligent sur FPGA

Par ailleurs, l'autre partie des contributions a consisté à concevoir le récepteur AIS intelligent, contenant la stratégie 1 que nous avions développée, et à développer aussi un laboratoire virtuel de simulation pour vérifier son comportement.

La conception du récepteur AIS est présentée dans le chapitre 5. Le récepteur a été implémenté avec la stratégie détectant les falsifications de position sur un FPGA. Pour simplifier et accélérer l'implémentation, nous avons utilisé un outil de synthèse HLS (Vitis HLS) pour générer automatiquement le code VHDL, implémentant le FPGA, à partir d'une modélisation niveau système en C++. Cet outil a été appliqué à la modélisation de l'ensemble du récepteur et a fourni, automatiquement, son architecture matérielle. Le récepteur conçu reçoit en entrée des signaux en bande de base, transmis par des AIS de classe A, extrait les informations contenues dans les messages et détecte les falsifications de position. Le comportement du récepteur a été vérifié sur un banc de test avec des signaux enregistrés dans la rade de Brest. La vérification a montré une extraction réussie des informations contenues dans ces signaux, en plus d'une exécution en temps réel.

Pour finir, pour vérifier de manière plus exhaustive le comportement du récepteur, un laboratoire virtuel pouvant synthétiser des bancs de test sur FPGA a été développé. Le développement de ce laboratoire est présenté dans le chapitre 6. Ce laboratoire simule, en même temps, le récepteur et son environnement maritime, ce qui représente un système cyber-physique. L'environnement simulé peut être modifié : le nombre d'émetteurs, leur trajectoire, l'offset en fréquence de leur transpondeur et le niveau de bruit des signaux transmis peuvent être réglés. En plus, des scénarios de falsification de position, de vitesse ou d'identité peuvent être générés par ce laboratoire de simulation. Pour concevoir ce laboratoire, une méthode de conception *top-down* a été appliquée. Elle part d'une modélisation en flot de données du CPS complet en considérant chaque composant (Emetteur, Récepteur ou Environnement) comme un acteur s'exécutant de manière indépendante et parallèle aux autres. La description du modèle se fait avec un langage haut niveau (C++) et l'utilisation de directives proposées par Vitis HLS. Le modèle est ensuite synthétisé automatiquement par Vitis HLS et conserve l'architecture en flot de données avec l'exécution parallèle des composants. L'utilisation de la HLS est ainsi étendue à la synthèse de systèmes cyber-physiques pour servir de bancs de tests. Le temps d'exécution du laboratoire ainsi conçu a été évalué par rapport à celui d'une exécution sur CPU et a montré une accélération allant de $\times 587$ à $\times 1660$.

Perspectives

Toutefois, plusieurs perspectives d'évolution sont envisageables. La stratégie 2, détectant les bateaux fantômes par la vérification du respect du mode d'accès TDMA, a comme inconvénient de renvoyer un nombre important de fausses alarmes. Ces fausses alarmes sont dues à la non-réception de messages provoqués par des déficiences techniques des transpondeurs ou par des conditions environnementales non favorables (obstacles naturels, tempête, etc.). Une voie d'amélioration pourrait être de développer une méthode de tri, basée sur de l'intelligence artificielle, pour séparer plus finement les fausses alarmes des vraies alarmes. La stratégie 3 qui détecte les falsifications d'identité, en utilisant l'offset en fréquence comme signature radiométrique des transpondeurs, peut, quant à elle, être améliorée et rendue plus efficace en considérant des signatures radiométriques supplémentaires (bruit de phase, déséquilibre I/Q...). Par ailleurs, une nouvelle stratégie pourrait être développée pour détecter l'arrêt volontaire d'un transpondeur AIS. La stratégie pourrait suivre précisément l'évolution de l'énergie des signaux transmis par chaque bateau pour décider si, lorsqu'un bateau arrête d'émettre des messages, cet arrêt est légitime ou non. Pour finir, nos stratégies ne s'appliquent, pour l'instant, qu'aux AIS de classe A. Il est largement envisageable d'étendre leur application aux AIS de classe B. Dans ce cas, de légères modifications devraient être apportées au récepteur, en particulier pour la stratégie 2, car une nouvelle configuration (CSTDMA) du mode d'accès TDMA doit être considérée. De la même manière, les trois stratégies peuvent être adaptées pour être utilisées par d'autres systèmes de communication sans fil. C'est en particulier le cas pour l'ADS-B, qui est le système équivalent à l'AIS, mais appliqué à la circulation aérienne. Pour ce système de communication, les stratégies 1 et 3 sont directement applicables, tout comme l'idée de la stratégie 2, qui doit cependant être modifiée pour s'adapter au protocole de communication de l'ADS-B.

Concernant la conception sur FPGA du récepteur, une amélioration pourrait être obtenue en implémentant d'autres stratégies (celles développées et celles envisagées) en suivant la méthode appliquée pour implémenter la stratégie 1. Par ailleurs, l'environnement modélisé peut être rendu plus complet et étoffé. Par exemple, un modèle plus proche de la réalité que le modèle de Friis peut être appliqué pour modéliser le canal de communication entre émetteur et récepteur. Ce modèle pourrait prendre, par exemple, en considération les réflexions des signaux sur l'eau ou l'atmosphère qui peuvent survenir lors des transmissions. Aussi, le mode d'accès TDMA pourrait être implémenté pour organiser les communications entre les émetteurs et notre récepteur, comme cela se fait dans la réalité. Enfin, la considération de cartes maritimes plus complexes, modélisant la présence d'îlots et de côtes terrestres, pourrait être ajoutée à la modélisation de l'environnement. Cette amélioration de la modélisation de l'environnement demande la prise en considération du temps physique et donc l'utilisation de modèles de calcul plus complexes.

Annexe A

Calcul des pseudo-accélérations sur la latitude et la longitude

Prenons comme exemple un point M repéré à la surface de la Terre dans le système de coordonnées WGS84 comme présenté sur la figure A.1. Soit G le centre de la Terre, M est repéré par ses 3 coordonnées qui sont : la distance (r), la latitude (λ) et la longitude (ϕ). H est le projeté de M dans le plan (G,$\vec{X}$,$\vec{Y}$). Nous définissons une base locale associée à ce système de coordonnées sphérique [160] notée ($\vec{e_r}$;$\vec{e_\lambda}$;$\vec{e_\phi}$) tel que :

— $\vec{e_r}$ est parallèle à $\overrightarrow{GM}$;
— $\vec{e_\lambda}$ est dans le plan ($\vec{Z}$,$\overrightarrow{GM}$) et orthogonal à $\vec{e_r}$ pointé dans le sens des λ croissant ;
— $\vec{e_\phi}$ est dans le plan ($\vec{X}$,$\vec{Y}$) tel que $\vec{e_\phi} = -\vec{e_r} \wedge \vec{e_\lambda}$

Les vecteurs de cette base locale ont pour coordonnées dans le repère (G,$\vec{e_X}$,$\vec{e_Y}$,$\vec{e_Z}$) orthonormé :

$$\vec{e_r} = \begin{pmatrix} \cos(\lambda)\cos(\phi) \\ \cos(\lambda)\sin(\phi) \\ \sin(\lambda) \end{pmatrix} ; \vec{e_\lambda} = \begin{pmatrix} -\sin(\lambda)\cos(\phi) \\ -\sin(\lambda)\sin(\phi) \\ \cos(\lambda) \end{pmatrix} ; \vec{e_\phi} = \begin{pmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{pmatrix}$$

Sachant que le repère (G,$\vec{e_X}$,$\vec{e_Y}$,$\vec{e_Z}$) est fixe (les vecteurs $\vec{e_X}$, $\vec{e_Y}$ et $\vec{e_Z}$ conservent la même direction, le même sens et la même norme au cours du temps), on peut déterminer les différentielles des vecteurs $\vec{e_r}$, $\vec{e_\lambda}$ et $\vec{e_\phi}$:

$$\begin{aligned} d\vec{e_r} &= d\lambda\vec{e_\lambda} + \cos(\lambda)d\phi\vec{e_\phi} \\ d\vec{e_\lambda} &= -d\lambda\vec{e_r} - \sin(\lambda)d\phi\vec{e_\phi} \\ d\vec{e_\phi} &= -\cos(\lambda)d\phi\vec{e_r} + \sin(\lambda)d\phi\vec{e_\lambda} \end{aligned}$$

Avec ces différentielles on peut déterminer, pour tout point M, sont vecteur position, vitesse et accélération dans le système de coordonnées WGS84. Le vecteur position s'écrit :

$$\overrightarrow{GM} = r\vec{e_r} \tag{A.1}$$

Le vecteur vitesse du point M s'écrit :

$$\frac{\overrightarrow{dGM}}{dt} = \dot{r}\vec{e_r} + r\dot{\lambda}\vec{e_\lambda} + r\cos(\lambda)\dot{\phi}\vec{e_\phi} \tag{A.2}$$

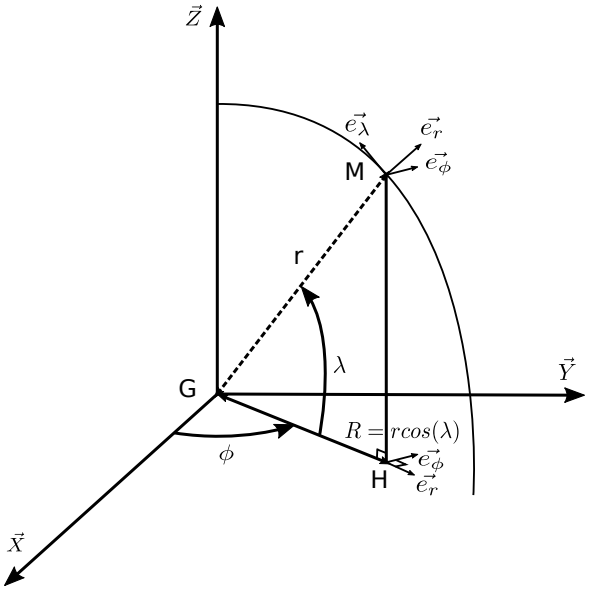

FIGURE A.1 – Représentation des coordonnées sphériques

Le vecteur accélération du point M s'écrit :

$$\overrightarrow{\frac{d^2GM}{dt^2}} = \left(\ddot{r} - r\dot{\lambda}^2 - rcos^2(\lambda)\dot{\phi}^2\right)\vec{e_r} + \left(2\dot{r}\dot{\lambda} + r\ddot{\lambda} + r\sin(\lambda)\cos(\lambda)\dot{\phi}^2\right)\vec{e_\lambda} + \left(-2r\sin(\lambda)\dot{\lambda}\dot{\phi} + 2\dot{r}\cos(\lambda)\dot{\phi} + r\cos(\lambda)\ddot{\phi}\right)\vec{e_\phi} \quad (A.3)$$

Dans un soucis de simplification r représente ici le rayon moyen de la terre. Le mouvement d'un navire s'effectue à la surface de la mer ce qui implique que le rayon r est contant ($\dot{r} = 0$ et $\ddot{r} = 0$). Ce qui donne :

$$\overrightarrow{\frac{d^2GM}{dt^2}} = \left(-r\dot{\lambda}^2 - r\cos^2(\lambda)\dot{\phi}^2\right)\vec{e_r} + \left(r\ddot{\lambda} + r\sin(\lambda)\cos(\lambda)\dot{\phi}^2\right)\vec{e_\lambda} + \left(-2r\sin(\lambda)\dot{\lambda}\dot{\phi} + r\cos(\lambda)\ddot{\phi}\right)\vec{e_\phi} \quad (A.4)$$

Maintenant avec cette équation, on peut déterminer l'accélération angulaire suivant $\vec{e_\lambda}$ et $\vec{e_\phi}$:

$$a_\lambda = \frac{(r\ddot{\lambda} + r\sin(\lambda)\cos(\lambda)\dot{\phi}^2)}{r} = \ddot{\lambda} + \sin(\lambda)\cos(\lambda)\dot{\phi}^2 \quad (A.5)$$

$$a_\phi = \frac{(-2r\sin(\lambda)\dot{\lambda}\dot{\phi} + r\cos(\lambda)\ddot{\phi})}{r\cos(\lambda)} = \ddot{\phi} - 2\tan(\lambda)\dot{\lambda}\dot{\phi} \quad (A.6)$$

Ces deux dernières équations (A.5) et (A.6) font apparaître des pseudo-accélérations angulaires qui sont les termes $\sin(\lambda)\cos(\lambda)\dot{\phi}^2$ et $-2\tan(\lambda)\dot{\lambda}\dot{\phi}$.

Annexe B

Démonstration de l'équation d'état à temps discret pour le modèle CA

Supposons que la dynamique d'une cible soit caractérisée par une accélération nulle sur la latitude et la longitude que nous caractérisons par la coordonnée générique ξ.

ξ suit donc l'équation différentielle suivante :

$$\ddot{\xi}(t) = 0 \tag{B.1}$$

En pratique la vitesse de la cible n'est jamais parfaitement constante et l'accélération a donc pour expression :

$$\ddot{\xi}(t) = \tilde{v} \tag{B.2}$$

avec $\tilde{v}(t)$ un bruit blanc continu tel que :

$$\mathbb{E}[\tilde{v}(t)] = 0 \tag{B.3}$$

$$\mathbb{E}[\tilde{v}(t)\tilde{v}(\tau)] = \tilde{q}\delta(t-\tau) \tag{B.4}$$

En posant le vecteur d'état $X(t) = \begin{pmatrix} \xi(t) \\ \dot{\xi}(t) \end{pmatrix}$ relatif à la composante $\xi(t)$, l'évolution de l'état de la cible peut être caractérisée par l'équation différentielle suivante :

$$\dot{X}(t) = AX(t) + D\tilde{v}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} X(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tilde{v}(t) \tag{B.5}$$

Cette équation d'état pilotée par un bruit blanc décrit un processus de Markov.

Cette équation a pour solution :

$$X(t) = F(t, t_0)X(t_0) + \int_{t_0}^{t} F(t,\tau)D\tilde{v}(\tau)\mathrm{d}\tau \tag{B.6}$$

Avec F la matrice de transition qui a pour expression (A invariant dans le temps) :

$$F(t, t_0) = e^{A(t-t_0)} = \begin{pmatrix} 1 & t-t_0 \\ 0 & 1 \end{pmatrix} \tag{B.7}$$

La représentation de l'état en temps discret devient :

$$X(t_{n+1}) = F(t_{n+1}, t_n)X(t_n) + V(t_n) \tag{B.8}$$

avec :

$$F(t_{n+1}, t_n) = F(t_{n+1} - t_n) = e^{A(t_{n+1}-t_n)} \equiv F_n \tag{B.9}$$

et :

$$V(t_n) = \int_{t_n}^{t_{n+1}} e^{(t_{n+1}-\tau)A} D\tilde{v}(\tau)\mathrm{d}\tau \equiv V_n \tag{B.10}$$

$\tilde{v}(t)$ est un bruit blanc ce qui implique :

$$\mathbb{E}[V_n] = 0 \tag{B.11}$$

$$\mathbb{E}[V_n V_n] = Q_n = \int_{t_n}^{t_{n+1}} \begin{pmatrix} t_{n+1} - \tau \\ 1 \end{pmatrix} \begin{pmatrix} t_{n+1} - \tau & 1 \end{pmatrix} \tilde{q}\mathrm{d}\tau = \begin{pmatrix} \frac{\Delta T_n^3}{3} & \frac{\Delta T_n^2}{2} \\ \frac{\Delta T_n^2}{2} & \Delta T_n \end{pmatrix} \tilde{q} \tag{B.12}$$

avec $\mathbb{E}[.]$ l'espérance mathématique, Q_n la matrice de covariance du bruit de modèle et $\Delta T_n = t_{n+1} - t_n$.

Annexe C

Test de conformité

Un test de conformité est destiné à vérifier si un échantillon peut être considéré comme représentatif d'une population de données, vis-à-vis d'un paramètre comme la moyenne ou la variance. Par exemple, dans notre application, nous savons que l'innovation $\tilde{Z}_n$ doit suivre une loi normale centrée de variance S_n. Chaque nouvelle mesure reçue dont l'innovation associée vérifiera la distribution normale de variance S_n sera acceptée. On dit que l'hypothèse nulle (H_0) est vérifiée. L'hypothèse alternative notée H_1 est l'hypothèse complémentaire à l'hypothèse nulle. H_1 est choisie uniquement par défaut si H_0 n'est pas considérée comme crédible. Pour savoir si H_1 ne peut pas être considérée comme crédible, nous fixons une région de rejet dépendant du risque de première espèce (α), appelé aussi probabilité de fausse alarme, que nous acceptons. Ce risque correspond à la probabilité que l'on accepte H_1 si la vérité est H_0 et s'écrit :

$$\alpha = P_{H_0}(H_1) \tag{C.1}$$

Le risque de deuxième espèce β, appelé aussi probabilité de non détection, est la probabilité de ne pas rejeter H_0 alors que la vérité est H_1. Il s'agit d'un risque qui n'est pas fixé a priori par le test, et est souvent difficile à estimer.

$$\beta = P_{H_1}(H_0) \tag{C.2}$$

La quantité $1-\alpha$ est la confiance du test. La quantité $1-\beta$ est la puissance du test. Ces différents risques sont représentés généralement sous forme d'un tableau du même type que le tableau (C.1). On définit généralement le risque α de façon arbitraire et la valeur du risque β s'ajuste automatiquement. Ce choix détermine alors une valeur seuil (notée S sur le schéma) qui représente la valeur de bascule pour la statistique du test entre les deux décisions (rejet ou non-rejet de H_0). Le graphique de la figure (C.1) tente de représenter visuellement ces risques, les courbes bleue et orange représentent respectivement la densité de probabilité sous l'hypothèse H_0 et la densité de probabilité sous l'hypothèse H_1. Les densités suivent une distribution normale dans notre exemple.

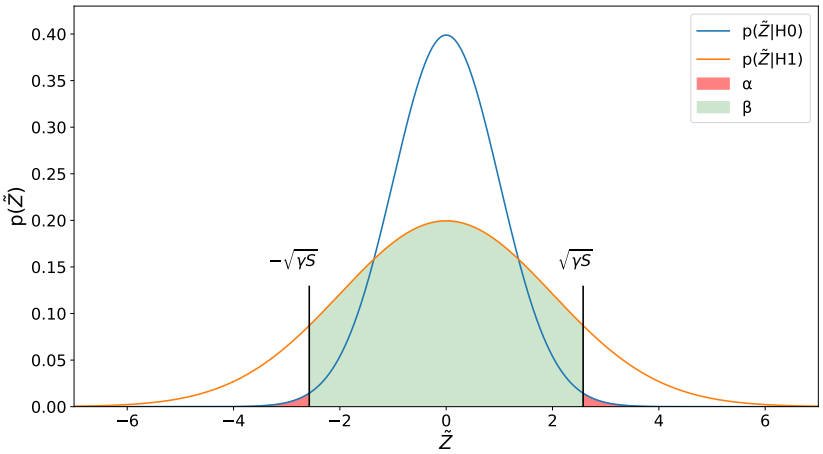

FIGURE C.1 – Evolution des densités de probabilité $p(\tilde{Z}|H_0)$ et $p(\tilde{Z}|H_1)$ de l'erreur de première α et de deuxième espèce β.

TABLEAU C.1 – Tableau associé au test statistique.

	Hypothèse H_0 vraie	**Hypothèse H_1 vraie**
Hypothèse H_0 acceptée	Bonne décision $(1-\alpha)$	Risque (β)
Hypothèse H_1 acceptée	Risque (α)	Bonne décision $(1-\beta)$

Annexe D

Données associées au mode d'accès TDMA

Les tableaux suivant présentent toutes les informations contenues dans un message de position et les informations utilisées pour mettre en place le mode d'accès TDMA. Ces tableaux sont extraits de la norme AIS [133].

TABLEAU D.1 – Information des messages de position 1, 2 et 3.

Parameter	**Nb of bits**	**Description**
Message ID	6	Identifier for this Message 1, 2 or 3
Repeat indicator	2	Used by the repeater to indicate how many times a message has been repeated. $0-3$; 0 = default ; 3 = do not repeat any more.
User ID	30	Unique identifier such as MMSI number
Navigational status	4	0 = under way using engine, 1 = at anchor, 2 = not under command, 3 = restricted maneuverability, 4 = constrained by her draught, 5 = moored, 6 = aground, 7 = engaged in fishing, 8 = under way sailing, 9 = reserved for future amendment of navigational status for ships carrying DG, HS, or MP, or IMO hazard or pollutant category C, high speed craft (HSC), 10 = reserved for future amendment of navigational status for ships carrying dangerous goods (DG), harmful substances (HS) or marine pollutants (MP), or IMO hazard or pollutant category A, wing in ground (WIG) ; 11 =power-driven vessel towing astern (regional use), 12 = power-driven vessel pushing ahead or towing alongside (regional use) ; 13 = reserved for future use, 14 = AIS-SART (active), MOB-AIS, EPIRB-AIS 15 = undefined = default (also used by AIS-SART, MOB-AIS and EPIRB-AIS under test).
Rate of turn ROT_{AIS}	8	0 à +126 = turning right at up to 708° per min or higher 0 à −126 = turning left at up to 708° per min or higher Values between 0° and 708° per min coded by $ROT_{AIS} = 4,733$, SQRT(ROT_{sensor}) degrees per min where ROT_{sensor} is the Rate of Turn as input by an external Rate of Turn Indicator (TI). ROT_{AIS} is rounded to the nearest integer value. +127 = turning right at more than 5° per 30 s (No TI available) −127 = turning left at more than 5° per 30 s (No TI available) −128 (80 hex) indicates no turn information available (default). ROT data should not be derived from COG information.
SOG	10	Speed over ground in 1/10 kn steps (0-102.2 kn) 1023 = not available, 1022 = 102,2 kn or higher
Position accuracy	1	1 = high ($\leq$ 10 m) 0 = low ($>$ 10 m) 0 = default
Longitude	28	Longitude in 1/10000 min (±180°, East = positive (as per 2′s complement) West = negative (as per 2's complement). 181 = (6791AC0h) = not available = default)
Latitude	27	Latitude in 1/10000 min (±90°, North = positive (as per 2's complement), South=negative (as per 2's complement). 91° = (3412140_h) = not available = default)
COG	12	Course over ground in 1/10 = $(0-3599).3600$ ($E10_h$) = not available = default. $3601-4095$ should not be used
True heading	9	Degrees $(0-359)$ (511 indicates not available = default)
Time stamp	6	UTC second when the report was generated by the electronic position system (EPFS) $(0-59$, or 60 if time stamp is not available, which should also be the default value, or 61 if positioning system is in manual input mode, or 62 if electronic position fixing system operates in estimated (dead reckoning) mode, or 63 if the positioning system is inoperative)
Special manœuvre indicator	2	0 = not available = default 1 = not engaged in special manœuvre 2 = engaged in special manœuvre (i.e. regional passing arrangement on Inland Waterway)
Spare	3	Not used. Should be set to zero. Reserved for future use.
RAIM-flag	1	Receiver autonomous integrity monitoring (RAIM) flag of electronic position fixing device ; 0 = RAIM not in use = default ; 1 = RAIM in use.
Communication state	19	voir Tableau D.2 pour SOTDMA et Tableau D.4 pour ITDMA
Number of bits	168	

Tableau D.2 – Données de l'état de communication du SOTDMA

Parameter	**Nb of bits**	**Description**
Sync state	2	0 UTC direct 1 UTC indirect 2 Station is synchronized to a base station (base direct) 3 Station is synchronized to another station based on the highest number of received stations or to another mobile station, which is directly synchronized to a base station
Slot time-out (STO (Slot time-out))	3	Specifies frames remaining until a new slot is selected 0 means that this was the last transmission in this slot $1-7$ means that 1 to 7 frames respectively are left until slot change
Sub message	14	The sub message depends on the current value in slot time-out Tableau D.3

Tableau D.3 – Explication du sous message de l'état de communication du SOTDMA

Slot time-out	**Sub message**	**Description**
3, 5, 7	Received stations	Number of other stations (not own station) which the station currently is receiving (between 0 and 16383).
2, 4, 6	Slot number	Slot number used for this transmission (between 0 and 2249).
1	UTC hour and minutes	If the station has access to UTC, the hour and minute should be indicated in this sub message. Hour $(0-23)$ should be coded in bits 13 to 9 of the sub message (bit 13 is MSB). Minute $(0-59)$ should be. coded in bit 8 to 2 (bit 8 is MSB). Bit 1 and bit 0 are not used.
0	Slot offset (Offset)	If the slot time-out value is 0 (zero) then the slot offset should indicate the offset to the slot in which transmission will occur during the next frame. If the slot offset is zero, the slot should be de-allocated after transmission.

Tableau D.4 – Explication des données contenues dans l'état de communication de l'ITDMA

Parameter	**Nb of bits**	**Description**
Sync state synchronisation	2	0 UTC direct 1 UTC indirect 2 Station is synchronized to a base station (base direct) 3 Station is synchronized to another station based on the highest number of received stations or to another mobile station, which is directly synchronized to a base station
Slot increment (Slot Incr.)	13	Offset to next slot to be used, or zero (0) if no more transmissions
Number of slots	3	Number of consecutive slots to allocate. $0 = 1$ slot, $1 = 2$ slots, $2 = 3$ slots, $3 = 4$ slots, $4 = 5$ slots, $5 = 1$ slot ; offset= slot increment +8192, $6 = 2$ slots ; offset = slot increment +8192, $7 = 3$ slots ; offset = slot increment +8192. Use of 5 to 7 removes the need for RATDMA broadcast for scheduled transmissions up to 6 min intervals.
Keep flag	1	Set to TRUE=1 if the slot remains allocated for one additional frame

Tableau D.5 – Mode d'accès et état de communication des messages

Message ID	**Name**	**Description**	**Priority**	**Access scheme**	**Communication state**	**M/B**
1	Position report	Scheduled position report ; (Class A shipborne mobile equipment)	1	SOTDMA, RATDMA, ITDMA	SOTDMA	M
2	Position report	Assigned scheduled position report ; (Class A shipborne mobile equipment)	1	SOTDMA	SOTDMA	M
3	Position report	Special position report, response to interrogation ; (Class A shipborne mobile equipment)	1	RATDMA	ITDMA	M

BIBLIOGRAPHIE

[1] 14 :00-17 :00, *ISO/IEC 13239 :2002*, URL : https://www.iso.org/cms/render/live/fr/sites/isoorg/contents/data/standard/03/70/37010.html.

[2] G ACKERSON et K FU, « On state estimation in switching environments », in : *IEEE transactions on automatic control* 15.*1* (1970), p. 10-17.

[3] AIRFLEET-VEDRAN, *AIS Spoofing : New Technologies for New Threats*, déc. 2022, URL : https://windward.ai/blog/ais-spoofing-new-technologies-for-new-threats/.

[4] *AISAtoN.Pdf*, URL : http://www.rokem.com/AISAtoN.pdf.

[5] Shahrokh AKHLAGHI, Ning ZHOU et Zhenyu HUANG, « Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation », in : *IEEE power & energy society general meeting*, 2017, p. 1-5.

[6] Rajoua ANANE, Kosai RAOOF et Ridha BOUALLEGUE, « On the evaluation of GMSK scheme with ECC techniques in wireless sensor network », in : *arXiv preprint arXiv :1505.05755* (2015).

[7] Andrej ANDROJNA et al., « AIS data vulnerability indicated by a spoofing case-study », in : *Appl. Sc.* 11.*11* (2021), p. 5015.

[8] François AUGER et al., « Industrial applications of the Kalman filter : A review », in : *IEEE Transactions on Industrial Electronics* 60.*12* (2013), p. 5458-5471.

[9] Ahmed AZIZ et al., « SecureAIS-securing pairwise vessels communications », in : *2020 IEEE Conference on Communications and Network Security (CNS)*, IEEE, 2020, p. 1-9.

[10] Marco BALDUZZI, Alessandro PASTA et Kyle WILHOIT, « A security evaluation of AIS automated identification system », in : *Proc. Annual Comput. Sec. Appl. Conf.* 2014, p. 436-445.

[11] Yaakov BAR-SHALOM et Kailash BIRMIWAL, « Variable dimension filter for maneuvering target tracking », in : *IEEE transactions on Aerospace and Electronic Systems 5* (1982), p. 621-629.

[12] Yaakov BAR-SHALOM, Thomas E FORTMANN et Peter G CABLE, *Tracking and data association*, 1990.

[13] Yaakov Bar-Shalom, X Rong Li et Thiagalingam Kirubarajan, *Estimation with application to tracking and navigation : theory algorithms and software*, John Wiley & Sons, 2004.

[14] Yaakov Bar-Shalom, X Rong Li et Thiagalingam Kirubarajan, *Estimation with application to tracking and navigation : theory algorithms and software*, John Wiley & Sons, 2004, p. 56.

[15] Yaakov Bar-Shalom et Xiao-Rong Li, *Multitarget-multisensor tracking : principles and techniques*, t. 19, YBs Storrs, CT, 1995.

[16] Samuel Blackman et Robert Popoli, « Design and analysis of modern tracking systems(Book) », in : *Norwood, MA : Artech House, 1999.* (1999).

[17] WD Blair, GA Watson et TR Rice, « Interacting multiple model filter for tracking maneuvering targets in spherical coordinates », in : *IEEE Proceedings of the SOUTHEASTCON'91*, IEEE, 1991, p. 1055-1059.

[18] HAP BLOM, « Bayesian estimation for decision-directed stochastic control(Ph. D. Thesis-Technische Hogeschool) », in : (1990).

[19] Henk AP Blom et Yaakov Bar-Shalom, « The interacting multiple model algorithm for systems with Markovian switching coefficients », in : *IEEE transactions on Automatic Control* 33.*8* (1988), p. 780-783.

[20] Henricus Albertus Petrus Blom, « An efficient filter for abruptly changing systems », in : *The 23rd IEEE Conference on Decision and Control*, IEEE, 1984, p. 656-658.

[21] Ruud M Bolle et al., *Guide to biometrics*, Springer Science & Business Media, 2013.

[22] Ettus Research Brand a National Instruments, *USRP E310 Embedded Software Defined Radio (SDR)*, URL : `https://www.ettus.com/all-products/e310/`.

[23] Vladimir Brik et al., « Wireless device identification with radiometric signatures », in : *Proceedings of the 14th ACM international conference on Mobile computing and networking*, 2008, p. 116-127.

[24] Eli Brookner, *Tracking and Kalman filtering made easy*, Wiley New York, 1998.

[25] Joseph T Buck et al., « Ptolemy : A framework for simulating and prototyping heterogeneous systems », in : (1994).

[26] *C++ Arbitrary Precision Fixed-Point Types • Vitis High-Level Synthesis User Guide (UG1399) • Reader • AMD Adaptive Computing Documentation Portal*, URL : `https://docs.xilinx.com/r/en-US/ug1399-vitis-hls/C-Arbitrary-Precision-Fixed-Point-Types?tocId=ZxETdjcRoXYHvIvpszYp9Q`.

[27] Jessica NA Campbell, Anthony W Isenor et Martha Dais Ferreira, « Detection of invalid AIS messages using machine learning techniques », in : *Procedia Computer Science* 205 (2022), p. 229-238.

[28] Maurantonio Caprolu et al., « Vessels cybersecurity : Issues, challenges, and the road ahead », in : *IEEE Comm. Mag.* 58.*6* (2020), p. 90-96.

[29] Henrik Carlsson et al., « Methods for reliable simulation-based PLC code verification », in : *IEEE Transactions on Industrial Informatics* 8.*2* (2012), p. 267-278.

[30] YT Chan, AGC Hu et JB Plant, « A Kalman filter based tracking scheme with input estimation », in : *IEEE transactions on Aerospace and Electronic Systems 2* (1979), p. 237-244.

[31] Wen Chen et al., « Challenges and trends in modern SoC design verification », in : *IEEE Design & Test* 34.*5* (2017), p. 7-22.

[32] Yuan Chen et Venkata Dinavahi, « Hardware emulation building blocks for real-time simulation of large-scale power grids », in : *IEEE Transactions on Industrial Informatics* 10.*1* (2013), p. 373-381.

[33] Charles K Chui, Guanrong Chen et al., *Kalman filtering*, Springer, 2017.

[34] *Compute Atan2 with LUT*, url : `https://forum.microchip.com/s/topic/a5C3l000000MQqdEAG/t333263?comment=P-2610916`.

[35] Jason Cong et al., « High-level synthesis for FPGAs : From prototyping to deployment », in : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30.*4* (2011), p. 473-491.

[36] *CRC-ITU Calculation in c#*, url : `https://social.msdn.microsoft.com/Forums/vstudio/en-US/e1ef3191-b4a8-4b15-882d-99ec594221cf/crcitu-calculation-in-c?forum=csharpgeneral`.

[37] Serena Curzel et al., « De-specializing an HLS library for Deep Neural Networks : improvements upon hls4ml », in : *arXiv preprint arXiv :2103.13060* (2021).

[38] Johan Eker et J Janneck, *CAL language report : Specification of the CAL actor language*, December, 2003.

[39] *Explication du nouveau standard AIS Classe B SOTDMA - Digital Yacht*, nov. 2018, url : `https://digitalyacht.fr/blog/2018/11/classeb-sotdma/`.

[40] Hafiyyan Sayyid Fadhlillah et al., « Towards heterogeneous multi-dimensional variability modeling in cyber-physical production systems », in : *Proceedings of the 25th ACM International Systems and Software Product Line Conference-Volume B*, 2021, p. 123-129.

[41] Sindre Fossen et Thor I Fossen, « Extended Kalman Filter Design and Motion Prediction of Ships using Live Automatic Identification System (AIS) Data », in : *European Conf. Elec. Eng. Comput. Sc.* 2018, p. 464-470.

[42] Harry D Foster, « Trends in functional verification : A 2014 industry study », in : *Proceedings of the 52nd Annual Design Automation Conference*, 2015, p. 1-6.

[43] Antoine Fremont, « Le transport maritime depuis 1945 : facteur clé de la mondialisation », in : *Entreprises et histoire 1* (2019), p. 16-29.

[44] Antoine Frémont, « Maritime transport : The threat of de-globalization ? », in : *Futuribles* 445.*6* (2021), p. 63-86.

[45] Franco Fummi et al., « Moving from co-simulation to simulation for effective smart systems design », in : *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2014, p. 1-4.

[46] Cláudio Gomes et al., « Co-simulation : a survey », in : *ACM Computing Surveys (CSUR)* 51.*3* (2018), p. 1-33.

[47] Cláudio Gomes et al., « The FMI 3.0 Standard Interface for Clocked and Scheduled Simulations », in : *Modelica Conferences*, 2021, p. 27-36.

[48] Athanassios Goudossis et Sokratis K Katsikas, « Towards a secure automatic identification system (AIS) », in : *Journal of Marine Science and Technology* 24.*2* (2019), p. 410-423.

[49] Pradeep Kumar Govindaiah, « Design and Development of Gaussian Minimum Shift Keying (GMSK) Demodulator for Satellite Communication », in : *Bonfring International Journal of Research in Communication Engineering* 2.*2* (2012), p. 06-11.

[50] Maria Daniela Graziano, Alfredo Renga et Antonio Moccia, « Integration of Automatic Identification System (AIS) data and single-channel Synthetic Aperture Radar (SAR) images by SAR-based ship velocity estimation for maritime situational awareness », in : *Remote Sensing* 11.*19* (2019), p. 2196.

[51] Mohinder S Grewal et Angus P Andrews, « Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives] », in : *IEEE Control Systems Magazine* 30.*3* (2010), p. 69-78.

[52] Marco Guerriero et al., « Analysis of AIS Intermittency and Vessel Characterization using a Hidden Markov Model. », in : *Gi jahrestagung (2)*, 2010.

[53] Marco Guerriero et al., « Radar/AIS data fusion and SAR tasking for Maritime Surveillance », in : *2008 11th International Conference on Information Fusion*, 2008, p. 1-5.

[54] Xavier Guillaud et al., « Applications of real-time simulation technologies in power and energy systems », in : *IEEE Power and Energy Technology Systems Journal* 2.*3* (2015), p. 103-115.

[55] Shanzeng Guo, « Space-based detection of spoofing AIS signals using Doppler frequency », in : *Multisensor, Multisource Information Fusion : Architectures, Algorithms, and Appl.* T. 9121, 2014, p. 1-6.

[56] Steven Herbst et al., « An open-source framework for FPGA emulation of analog/mixed-signal integrated circuit designs », in : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41.*7* (2021), p. 2223-2236.

[57] Simen Hexeberg, Andreas L Flåten, Edmund F Brekke et al., « AIS-based vessel trajectory prediction », in : *Int. Conf. Inf. Fusion*, 2017, p. 1-8.

[58] Weikun Hou et al., « Physical layer authentication for mobile systems with time-varying carrier frequency offsets », in : *IEEE TC* 62.*5* (2014), p. 1658-1667.

[59] Inseok Hwang et Chze Eng Seah, « Intent-based probabilistic conflict detection for the next generation air transportation system », in : *Proceedings of the IEEE* 96.*12* (2008), p. 2040-2059.

[60] *Indonesia spots Chinese research vessel with tracking system off - SAFETY4SEA*, url : `https://safety4sea.com/indonesia-spots-chinese-research-vessel-with-tracking-system-off/`.

[61] *International Convention for the Safety of Life at Sea (SOLAS) - DGRM*, url : `https://www.dgrm.mm.gov.pt/solas`.

[62] *Iran, Tanzania and Falsifying AIS Signals to Trade with Syria*, en, url : `https://www.maritime-executive.com/article/iran-tanzania-and-falsifying-ais-signals-to-trade-with-syria`.

[63] *Iranian Tanker Hacks AIS to Disguise Itself Off Singapore*, url : `https://gcaptain.com/iranian-tanker-hacks-disguise/`.

[64] Ali Jafarnia-Jahromi et al., « GPS vulnerability to spoofing threats and a review of antispoofing techniques », in : *International Journal of Navigation and Observation* 2012 (2012).

[65] Krzysztof Jaskólski, « Automatic Identification System (AIS) dynamic data estimation based on discrete Kalman Filter (KF) algorithm », in : *Scientific J. Polish Naval Academy* 211.*4* (2017), p. 71-87.

[66] Krzysztof Jaskólski et al., « Automatic Identification System (AIS) Dynamic Data Integrity Monitoring and Trajectory Tracking Based on the Simultaneous Localization and Mapping (SLAM) Process Model », in : *Sensors* 21.*24* (2021), p. 8430.

[67] Nasser Jazdi, « Cyber physical systems in the context of Industry 4.0 », in : *2014 IEEE international conference on automation, quality and testing, robotics*, IEEE, 2014, p. 1-4.

[68] Jeff C Jensen, Danica H Chang et Edward A Lee, « A model-based design methodology for cyber-physical systems », in : *2011 7th international wireless communications and mobile computing conference*, IEEE, 2011, p. 1666-1671.

[69] Paul R Kalata, « α-β target tracking systems : A survey », in : *1992 American Control Conference*, IEEE, 1992, p. 832-836.

[70] Rudolph E Kalman et Richard S Bucy, « New results in linear filtering and prediction theory », in : (1961).

[71] Elliott D Kaplan et Christopher Hegarty, *Understanding GPS/GNSS : principles and applications*, Artech house, 2017.

[72] Fotios Katsilieris, Paolo Braca et Stefano Coraluppi, « Detection of malicious AIS position spoofing by exploiting radar information », in : *Int. Conf. Inf. Fusion*, 2013, p. 1196-1203.

[73] John G Kemeny, J Laurie Snell et Anthony W Knapp, *Denumerable Markov chains : with a chapter of Markov random fields by David Griffeath*, t. 40, Springer Science & Business Media, 2012.

[74] GC Kessler, « Protected AIS : a demonstration of capability scheme to provide authentication and message integrity », in : *TransNav : International Journal on Marine Navigation and Safety of Sea Transportation* 14.*2* (2020).

[75] Thiagalingam Kirubarajan et Yaakov Bar-Shalom, « Kalman filter versus IMM estimator : when do we need the latter ? », in : *IEEE Transactions on Aerospace and Electronic Systems* 39.*4* (2003), p. 1452-1457.

[76] Ioannis Kontopoulos et al., « Countering Real-Time Stream Poisoning : An architecture for detecting vessel spoofing in streams of AIS data », in : *IEEE Conf. Dependable, Autonomic and Secure Computing,* 2018, p. 981-986.

[77] Harry H Ku et al., « Notes on the use of propagation of error formulas », in : *J. Res. Nat. Bur. Stand.* 70.*4* (1966), p. 263-273.

[78] John A Kusters et John R Vig, « Thermal hysteresis in quartz resonators-A review (frequency standards) », in : *44th Annual Symposium on Frequency Control*, IEEE, 1990, p. 165-175.

[79] Sakari Lahti et al., « Are we there yet ? A study on the state of high-level synthesis », in : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38.*5* (2018), p. 898-911.

[80] Philipp Last, Martin Hering-Bertram et Lars Linsen, « How automatic identification system (AIS) antenna setup affects AIS signal quality », in : *Ocean Engineering* 100 (2015), p. 83-89.

[81] Philipp Last et al., « Comprehensive analysis of automatic identification system (AIS) data in regard to vessel movement prediction », in : *J. Navig.* 67.*5* (2014), p. 791-809.

[82] Edward A Lee, « Cyber physical systems : Design challenges », in : *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*, IEEE, 2008, p. 363-369.

[83] Edward Ashford Lee et Sanjit Arunkumar Seshia, *Introduction to embedded systems : A cyber-physical systems approach*, Mit Press, 2016.

[84] Mauro Leonardi, Luca Di Gregorio et Davide Di Fausto, « Air traffic security : Aircraft classification using ADS-B message's phase-pattern », in : *Aerospace* 4.*4* (2017), p. 51.

[85] X Rong Li et Vesselin P Jilkov, « Survey of maneuvering target tracking : III. Measurement models », in : *Signal and Data Processing of Small Targets 2001*, t. 4473, SPIE, 2001, p. 423-446.

[86] X Rong Li et Vesselin P Jilkov, « Survey of maneuvering target tracking. Part I. Dynamic models », in : *IEEE Transactions on aerospace and electronic systems* 39.*4* (2003), p. 1333-1364.

[87] X Rong Li et Youmin Zhang, « Multiple-model estimation with variable structure. V. Likely-model set algorithm », in : *IEEE Transactions on Aerospace and Electronic Systems* 36.*2* (2000), p. 448-466.

[88] Fiona Jiazi Liu, Xianbin Wang et Helen Tang, « Robust physical layer authentication using inherent properties of channel impulse response », in : *2011-MILCOM 2011 Military Communications Conference*, IEEE, 2011, p. 538-542.

[89] Xinheng Liu et al., « High level synthesis of complex applications : An H. 264 video decoder », in : *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2016, p. 224-233.

[90] *Livre Blanc - Émulation et Simulation*, URL : `http://yannickprie.net/archives/VEILLE-2009-2012/2010/simu3d-tpsreel/lb\_simul.html\#:\textasciitilde:text=En\%20effet\%2C\%20la\%20simulation\%20consiste,une\%20machine\%20par\%20un\%20logiciel.`.

[91] Maelic Louart, *Stratégie détantant les falsifications d'identité*, URL : `https://github.com/maelic-louart/identity_spoofing_detection`.

[92] Maelic Louart, *Stratégie globale*, URL : https://github.com/maelic-louart/global_strategy.

[93] Maelic Louart, *TDMA method checking*, URL : https://github.com/maelic-louart/TDMA-method-checking.

[94] Maelic Louart et al., « Detection of AIS messages falsifications and spoofing by checking messages compliance with TDMA protocol », in : *Digital Signal Processing* (2023), p. 103983, ISSN : 1051-2004.

[95] Maelic Louart et al., « Émulation de Systèmes Cyber-Physiques sur FPGA », in : (2022).

[96] Maelic Louart et al., « HLS-based Accelerated Simulation of Large Scale Cyber-Physical Systems on FPGAs », in : *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)*, IEEE, 2022, p. 332-336.

[97] Maelic Louart et al., « Stratégie de détection des Falsifications des Positions des Messages AIS Basée sur l'Application du Filtre IMM », in : *Gretsi'22 XXVIIIème Colloque Francophone de Traitement du Signal et des Images*, 2022.

[98] Herman Lundkvist et Alexander Yngve, *Accelerated simulation of modelica models using an FPGA-based approach*, 2018.

[99] Shangbo Mao et al., « An automatic identification system (AIS) database for maritime trajectory prediction and data mining », in : *Proc. ELM*, 2018, p. 241-257.

[100] *MarineTraffic : Global Ship Tracking Intelligence | AIS Marine Traffic*, URL : https://www.marinetraffic.com/en/ais/home/centerx:25.7/centery:25.5/zoom:2.

[101] Grant Martin, Brian Bailey et Andrew Piziali, *ESL design and verification : a prescription for electronic system level methodology*, Elsevier, 2010.

[102] Grant Martin et Gary Smith, « High-level synthesis : Past, present, and future », in : *IEEE Design & Test of Computers* 26.*4* (2009), p. 18-25.

[103] Peter Marwedel, *Embedded system design : embedded systems foundations of cyber-physical systems, and the internet of things*, Springer Nature, 2021.

[104] Fabio Mazzarella, Virginia Fernandez Arguedas et Michele Vespe, « Knowledge-based vessel position prediction using historical AIS data », in : *Sensor Data Fusion : Trends, Solutions, Appl.* 2015, p. 1-6.

[105] Fabio Mazzarella et al., « A novel anomaly detection approach to identify intentional AIS on-off switching », in : *Expert Syst. Appl.* 78 (2017), p. 110-123.

[106] « Methods of Computing Square Roots », in : *Wikipedia* (fév. 2021).

[107] Swapnil Mhaske et al., « FPGA-accelerated simulation of a hybrid-ARQ system using high level synthesis », in : *2016 IEEE 37th Sarnoff Symposium*, IEEE, 2016, p. 19-21.

[108] Bailey Miller, Frank Vahid et Tony Givargis, « Digital mockups for the testing of a medical ventilator », in : *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, 2012, p. 859-862.

[109] Shridhar Mubaraq Mishra et al., « A real time cognitive radio testbed for physical and link layer experiments », in : *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.* IEEE, 2005, p. 562-567.

[110] *Modules Amplificateurs LNA Avec Filtre SAW et Fréquence Unique*, URL : `https://www.kubii.fr/cartes-extension-cameras-raspberry-pi/3349-modules-amplificateurs-lna-avec-filtre-saw-3272496306646.html`.

[111] Sameer Mojlish et al., « Review of hardware platforms for real-time simulation of electric machines », in : *IEEE Transactions on Transportation Electrification* 3.*1* (2017), p. 130-146.

[112] Jun Min Mou, Cees Van der Tak et Han Ligteringen, « Study on collision avoidance in busy waterways by using AIS data », in : *Ocean Eng.* 37.*5-6* (2010), p. 483-490.

[113] Saad Mubeen, Elena Lisova et Aneta Vulgarakis Feljan, « Timing predictability and security in safety-critical industrial cyber-physical systems : A position paper », in : *Applied Sciences* 10.*9* (2020), p. 3125.

[114] Gabriela Nicolescu et Pieter J Mosterman, *Model-based design for embedded systems*, Crc Press, 2018.

[115] Peter Palensky, Edmund Widl et Atiyah Elsheikh, « Simulating cyber-physical energy systems : Challenges, tools and methods », in : *IEEE Transactions on Systems, Man, and Cybernetics : Systems* 44.*3* (2013), p. 318-326.

[116] Benjamin Pannetier, « Fusion de données pour la surveillance du champ de bataille », thèse de doct., Université Joseph-Fourier-Grenoble I, 2006.

[117] Francesco Papi et al., « Radiolocation and tracking of automatic identification system signals for maritime situational awareness », in : *IET Radar, Sonar & Navigation* 9.*5* (2014), p. 568-580.

[118] Neal Patwari et Sneha K Kasera, « Robust location distinction using temporal link signatures », in : *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, 2007, p. 111-122.

[119] *Positions of Two NATO Ships Were Falsified Near Russian Black Sea Naval Base - USNI News*, URL : https://news.usni.org/2021/06/21/positions-of-two-nato-ships-were-falsified-near-russian-black-sea-naval-base.

[120] Mark L PSIAKI et Todd E HUMPHREYS, « GNSS spoofing and detection », in : *Proceedings of the IEEE* 104.*6* (2016), p. 1258-1270.

[121] Akshay RAJHANS et al., « Supporting heterogeneity in cyber-physical systems architectures », in : *IEEE Transactions on Automatic Control* 59.*12* (2014), p. 3178-3193.

[122] Denise RATASICH et al., « A roadmap toward the resilient internet of things for cyber-physical systems », in : *IEEE Access* 7 (2019), p. 13260-13283.

[123] Cyril RAY, Clément IPHAR et Aldo NAPOLI, « Methodology for Real-Time Detection of AIS Falsification », in : *Maritime Knowledge Discovery and Anomaly Detection Workshop, Michele Vespe and Fabio Mazzarella, Eds., Ispra, Italy*, 2016, p. 74-77.

[124] Martin REDOUTEY et al., « Efficient vessel tracking with accuracy guarantees », in : *Int. Symposium on Web and WGIS*, Springer, 2008, p. 140-151.

[125] Branko RISTIC et al., « Statistical analysis of motion patterns in AIS data : Anomaly detection and motion prediction », in : *Int. Conf. Inf. Fusion*, 2008, p. 1-7.

[126] Maria RIVEIRO, Giuliana PALLOTTA et Michele VESPE, « Maritime anomaly detection : A review », in : *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery* 8.*5* (2018), p. 1-34.

[127] D. ROY et al., « Waterfall is Too Slow, Let's Go Agile : Multi-Domain Coupling for Synthesizing Automotive Cyber-Physical Systems », in : *Proc. of the Int. Conf. on Computer-Aided Design*, ICCAD '18, San Diego, California, ISBN : 9781450359504, DOI : 10.1145/3240765.3243500, URL : https://doi.org/10.1145/3240765.3243500.

[128] Winston W ROYCE, « Managing the development of large software systems : concepts and techniques », in : *Proceedings of the 9th international conference on Software Engineering*, 1987, p. 328-338.

[129] A. SANGIOVANNI-VINCENTELLI, « The tides of EDA », in : *IEEE Design & Test of Computers* 20.*6* (2003), p. 59-75, DOI : 10.1109/MDT.2003.1246165.

[130] Alberto SANGIOVANNI-VINCENTELLI, « The tides of EDA », in : *IEEE Design & Test of Computers* 20.*6* (2003), p. 59-75.

[131] Benjamin Carrion SCHAFER et Zi WANG, « High-level synthesis design space exploration : Past, present, and future », in : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39.*10* (2019), p. 2628-2639.

[132] Savio Sciancalepore et al., « Auth-AIS : Secure, Flexible, and Backward-Compatible Authentication of Vessels AIS Broadcasts », in : *IEEE Trans. Dependable and Secure Computing* (2021).

[133] M Series, « Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band », in : *Recommendation ITU : Geneva* (2014), p. 1371-1375.

[134] Mageda Sharafeddin et al., « On the effectiveness of accelerating MapReduce functions using the Xilinx Vivado HLS tool », in : *International Journal of High Performance Systems Architecture* 6.*1* (2016), p. 1-12.

[135] Pan Sheng et Jingbo Yin, « Extracting shipping route patterns by trajectory clustering model based on automatic identification system data », in : *Sustainability* 10.*7* (2018), p. 1-3.

[136] Yan Shi et Michael A Jensen, « Improved radiometric identification of wireless devices using MIMO transmission », in : *IEEE Trans. on Inf. Forensics and Security* 6.*4* (2011), p. 1346-1354.

[137] Abdoulaye Sidibé et Gao Shu, « Study of automatic anomalous behaviour detection techniques for maritime vessels », in : *J. Navig.* 70.*4* (2017), p. 847-858.

[138] Gregor Siegert et al., « EKF based trajectory tracking and integrity monitoring of AIS data », in : *Proceedings of IEEE/ION PLANS 2016*, 2016, p. 887-897.

[139] PAM Silveira, AP Teixeira et C Guedes Soares, « Use ofAIS data to characterize marine traffic patterns and ship collision risk off the coast of Portugal », in : *J. Navig.* 66.*6* (2013), p. 879.

[140] Robert W Sittler, « An optimal data association problem in surveillance theory », in : *IEEE transactions on military electronics* 8.*2* (1964), p. 125-139.

[141] Surinder Sood, Avinash Malik et Partha Roop, « Robust Design and Validation of Cyber-physical Systems », in : *ACM Transactions on Embedded Computing Systems (TECS)* 18.*6* (2019), p. 1-21.

[142] Svetoslav Sotirov et Chavdar Alexandrov, « Improving AIS data reliability », in : *Global perspectives in MET : Towards Sustainable, Green and Integrated Maritime Transport*, 2017, p. 237-244.

[143] Martin Strohmeier et al., « Crowdsourcing security for wireless air traffic communications », in : *Int. Conf. Cyber Conflict (CyCon)*, 2017, p. 1-18.

[144] Yang Sun et al., « Ship trajectory cleansing and prediction with historical ais data using an ensemble ann framework », in : *Int. J. Innov. Comput. Inf. Control* 17 (2021), p. 443-459.

[145] *Systematic GPS Manipulation Occuring at Chinese Oil Terminals and Government Installations*, 2019, URL : https://skytruth.org/2019/12/systematic-gps-manipulation-occuring-at-chinese-oil-terminals-and-government-installations/.

[146] Janos SZTIPANOVITS et al., « Toward a science of cyber–physical system integration », in : *Proceedings of the IEEE* 100.*1* (2011), p. 29-44.

[147] Kenneth I TALBOT, Paul R DULEY et Martin H HYATT, « Specific emitter identification and verification », in : *Technology Review* 113 (2003), p. 113-130.

[148] OH TEKBAS, Nur SERINKEN et O URETEN, « An experimental performance evaluation of a novel radio-transmitter identification system under diverse environmental conditions », in : *Canadian Journal of Electrical and Computer Engineering* 29.*3* (2004), p. 203-209.

[149] Petros TOUPAS, Andreas BROKALAKIS et Ioannis PAPAEFSTATHIOU, « Accelerating Physics Engine Components with Embedded FPGAs », in : *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, 2019, p. 88-94.

[150] Mai-Thanh TRAN, « Towards hardware synthesis of a flexible radio from a high-level language », thèse de doct., Rennes 1, 2018.

[151] Jitendra K TUGNAIT, « Wireless user authentication via comparison of power spectral densities », in : *IEEE Journal on Selected Areas in Communications* 31.*9* (2013), p. 1791-1802.

[152] Jitendra K TUGNAIT et Abraham H HADDAD, « A detection-estimation scheme for state estimation in switching environments », in : *Automatica* 15.*4* (1979), p. 477-481.

[153] Michele VESPE et al., « Maritime multi-sensor data association based on geographic and navigational knowledge », in : *IEEE Radar Conf.* 2008, p. 1-6.

[154] *Vessel Database Search - Discover more than 500000 ships*, URL : https://www.fleetmon.com/vessels/.

[155] *Viewpoint : The Next IC Design Methodology Transition Is Long Overdue*, URL : https://www.accellera.org/resources/articles/icdesigntrans.

[156] *Virtex UltraScale+ HBM VCU128 FPGA Evaluation Kit*, URL : https://www.xilinx.com/products/boards-and-kits/vcu128.html.

[157] *Vitis High-Level Synthesis 2022.2*, URL : https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0090-vitis-hls-hub.html.

[158] Ruokun Wang, Biyang Wen et Weimin Huang, « A support vector regression-based method for target direction of arrival estimation from HF radar data », in : *IEEE Geoscience and Remote Sensing Letters* 15.*5* (2018), p. 674-678.

[159] Global Fishing Watch, *Spoofing : One Identity Shared by Multiple Vessels*, juill. 2016.

[160] Eric W. Weisstein, *Spherical Coordinates*, Text, url : `https://mathworld.wolfram.com/`.

[161] *Wikiwand - AIS-SART*, url : `https://wikiwand.com/en/AIS-SART`.

[162] TH Witte et AM Wilson, « Accuracy of non-differential GPS for the determination of speed over ground », in : *J. Biomechanics* 37.*12* (2004), p. 1891-1898.

[163] Liang Xiao et al., « Fingerprints in the ether : Using the physical layer for wireless authentication », in : *2007 IEEE International conference on communications*, IEEE, 2007, p. 4646-4651.

[164] Ning Xie, Zhuoyuan Li et Haijun Tan, « A survey of physical-layer authentication in wireless communications », in : *IEEE Communications Surveys & Tutorials* 23.*1* (2020), p. 282-310.

[165] Jie Yang et al., « Detection and localization of multiple spoofing attackers in wireless networks », in : *IEEE Transactions on Parallel and Distributed systems* 24.*1* (2012), p. 44-58.

[166] Murali Yeddanapudi, Yakoov Bar-Shalom et Krishna Pattipati, « IMM estimation for multitarget-multisensor air traffic surveillance », in : *Proceedings of the IEEE* 85.*1* (1997), p. 80-96.

[167] Masato Yoshimi et al., « Stochastic simulation for biochemical reactions on FPGA », in : *International Conference on Field Programmable Logic and Applications*, Springer, 2004, p. 105-114.

[168] Bernard P Zeigler, Tag Gon Kim et Herbert Praehofer, *Theory of modeling and simulation*, Academic press, 2000.

[169] Kai Zeng, Kannan Govindan et Prasant Mohapatra, « Non-cryptographic authentication and identification in wireless networks [security and privacy in emerging wireless networks] », in : *IEEE Wireless Communications* 17.*5* (2010), p. 56-62.

[170] Tao Zhang et al., « Detection of ais closing behavior and mmsi spoofing behavior of ships based on spatiotemporal data », in : *Remote Sensing* 12.*4* (2020), p. 702.

[171] Xi Zheng et al., « Perceptions on the state of the art in verification and validation in cyber-physical systems », in : *IEEE Systems Journal* 11.*4* (2015), p. 2614-2627.

[172] Hui Zhou et al., « Frequency accuracy & stability dependencies of crystal oscillators », in : *Carleton University, Systems and Comput. Engineering, Technical Report SCE-08-12* (2008).

[173] Xin Zhou et al., « Review on testing of cyber physical systems : Methods and testbeds », in : *IEEE Access* 6 (2018), p. 52179-52194.

COLLEGES SCIENCES
BRETAGNE POUR L'INGENIEUR
LOIRE ET LE NUMERIQUE

ENSTA BRETAGNE

Title: Design of an AIS receiver detecting messages falsifications: strategies development and FPGA prototyping

Keywords: AIS, falsifications detection, FPGA-based simulation of CPS, HLS

Abstract: AIS is a widely used communication system for ships. It allows the automatic exchange of information such as position, speed, course, identity, port of departure and arrival, etc. This information assists navigation, secures maritime traffic and simplifies surveillance by the coastguard. However, having been developed in the 90s, AIS is not very secure. It is easy for a malicious user to transmit false information, to jam communications or to make ghost ships appear. The interest can be to hide fraudulent activities, disrupt traffic or justify false "territorial violations".

Based on this observation, in this manuscript, three strategies have been developed to detect these attacks. The first one tracks ships with a filter to detect position and speed falsifications. The second verifies that the boats respect the TDMA access mode when they communicate. The verification of this access mode, specified by the AIS standard, makes it possible to detect the transmission of false messages and the creation of ghost ships. Finally, the last strategy calculates a radiometric signature from the signals transmitted by each transponder in order to identify them physically and not by the identity they transmit in their messages. This makes it possible to detect identity spoofing. These three strategies are applied jointly and have been tested on real signals recorded in the Brest harbour. First and second kind error rates of 0.0035 and 0.0171 respectively on the detection of identity spoofing have been obtained. In addition, Monte Carlo simulations have shown a sensitivity to position falsifications ranging from 35m to 250m.

Some of these strategies have been implemented on a FPGA to be used by the coastguard or the navy. The entire AIS signal demodulation chain was also implemented on the same board to recover the signals and information used by these strategies. The generation of the code implementing the FPGA was done using the HLS which is a high level synthesis tool. This application case showed that this tool is now a mature technology that can efficiently and automatically synthesise large complete systems from high-level modelling, thus removing the separation that once existed between signal processing and hardware design activities. The resulting AIS receiver can detect position falsifications from baseband signals in real time.

Finally, an FPGA-based virtual laboratory that simulates the receiver and its environment at the same time has also been developed. It allows to easily generate various test scenarios to verify the behaviour of the receiver in its limit conditions of use. The joint consideration of the receiver, its environment and their interactions led us to talk about cyber-physical systems and the difficulties, still current, associated with the modelling and simulation of such systems. A new method has been proposed to solve these difficulties. This method models the CPS as a set of concurrently executing actors, uses the FPGA as a simulation platform and applies HLS to automatically generate the hardware description from the actor-based modelling. A virtual laboratory has been created for our receiver.It allows to simulate, at the same time, our receiver and its maritime environment for which the number of communicating vessels, their trajectories, the SNR of their transmitted signals and their transponder's CFO can be modified.