



HAL
open science

Estimation des paramètres de multitrajets pour des signaux GNSS par apprentissage profond et application en environnement de simulation 3D

Thomas Gonzalez

► **To cite this version:**

Thomas Gonzalez. Estimation des paramètres de multitrajets pour des signaux GNSS par apprentissage profond et application en environnement de simulation 3D. Informatique [cs]. Université Toulouse III - Paul Sabatier, 2024. Français. NNT : . tel-04669754v1

HAL Id: tel-04669754

<https://enac.hal.science/tel-04669754v1>

Submitted on 9 Aug 2024 (v1), last revised 10 Sep 2024 (v2)

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

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

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Estimation des paramètres de multitrajets pour des signaux
GNSS par apprentissage profond et application en
environnement de simulation 3D

Thèse présentée et soutenue, le 22 avril 2024 par
Thomas GONZALEZ

École doctorale

EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité

Informatique et Télécommunications

Unité de recherche

ENAC-LAB - Laboratoire de Recherche ENAC

Thèse dirigée par

Nicolas COUELLAN et Antoine BLAIS

Composition du jury

Mme Juliette MARAIS, Présidente, Université Gustave Eiffel

Mme Audrey GIREMUS, Rapporteuse, Université de Bordeaux

Mme Ni ZHU, Examinatrice, Université Gustave Eiffel

Mme Sandrine MOUYSSET, Examinatrice, Université Toulouse III - Paul Sabatier

M. Nicolas COUELLAN, Directeur de thèse, Ecole Nationale de l'Aviation Civile

M. Antoine BLAIS, Co-directeur de thèse, Ecole Nationale de l'Aviation Civile

Membres invités

M. Christian Ruiz, Oktal-SE

Résumé

Le système de réception de signaux *Global Navigation Satellite System* (GNSS) présente une vulnérabilité face au phénomène de multitrajet. Événement fréquent en milieu urbain, il peut mener à une forte détérioration des performances de positionnement du récepteur. Malgré le développement de nombreuses techniques à des fins de détection ou de réduction de ses effets comme le *Fast Iterative Maximum-Likelihood Algorithm* (FIMLA) ou les filtres particulaires, la diminution de l'impact du multitrajet au sein d'un récepteur GNSS reste un objectif difficile à atteindre. Dans cette thèse, nous nous proposons d'appliquer un algorithme de *Machine Learning* pour éliminer la composante du multitrajet par l'estimation de quatre paramètres déterminants : son retard, sa fréquence Doppler, son amplitude et sa phase. Nous utiliserons des données sous forme d'images labellisées. Ces images sont les représentations des canaux I et Q aux sorties du corrélateur. Les signaux seront synthétiques et les multitrajets générés selon deux protocoles différents : aléatoirement ou collectés à partir d'un simulateur de propagation de signaux GNSS, capable de reproduire des situations propices aux multitrajets (ex : simulation de réception dans un canyon urbain). Les informations sur le signal et sa réplique parasite sont réparties parmi les images I et Q des sorties de corrélateur construites par le générateur, le nombre de sorties de corrélateur servant à la construction des images étant flexible afin de générer différentes résolutions d'images. Dans le but d'effectuer la tâche de régression des paramètres de multitrajets sur les images I et Q, complexe par l'aspect multicanal du problème, nous utiliserons des réseaux de neurones à convolution ou *Convolutional Neural Network* (CNN). Utilisés à maintes reprises sur des problèmes de classification, ils sont devenus une référence dans l'analyse d'images via des algorithmes de *Machine Learning*. Malgré une utilisation moins démocratisée en régression, ils montrent pourtant de bonnes capacités d'estimation comme pour l'estimation d'âge ou l'estimation de positions d'un corps dans un espace 3D. Ici nous introduisons un nouveau modèle de réseau de neurones à convolution combinant la capacité d'extraction automatique des points d'intérêts hautement informatifs par les couches de convolution et une technique de *soft labelling* améliorant la robustesse du CNN lors de son étape de généralisation. L'idée du *soft labelling* est de diminuer la confiance de l'algorithme envers les *labels* afin de rendre la tâche de régression plus aisée, étant complexe pour des données sous forme d'images. Les *soft labels* sont alors vus comme des distributions de probabilités d'espérances égales aux *labels*. Inspirés de travaux antérieurs effectués par Imani et al. (2018), nous utilisons une fonction perte spécifiquement adaptée à la mesure de dissimilarité entre les nouveaux *labels* et leur prédiction avec la divergence de Kullback-Leibler. Le nouveau modèle de CNN a été testé sur différents *datasets* d'images présentant différents niveaux de bruits et de tailles d'images afin de tester les qualités d'estimation de l'algorithme ainsi que sa robustesse. Les résultats montrent dans toutes les conditions testées une claire amélioration de la précision d'estimation par CNN avec la méthode de *soft labelling* par rapport à une régression « classique » effectuée sur une variable continue. De plus le gain de performances apporté par la méthode de *soft labelling* par distribution (*Distribution loss*) permet de travailler avec des tailles d'images en entrée réduites de 80x80 à 20x20 pixels. Le fonctionnement du CNN

Résumé

pour un nombre de sorties moins important laisse envisager une utilisation embarquée sur des technologies comme le *Vision Processing Unit* (VPU) pour l'élimination de multitrajets .

Mots-Clés : CNN, Distributional Loss, divergence de Kullback-Leibler, environnement synthétique, estimation de multitrajet, Gaussienne tronquée, GNSS, régression multicanaux, régression sur images, Soft Labelling, sorties de corrélateur, Von Mises.

Remerciements

Si je le pouvais, le chapitre remerciements serait la section contenant le plus de pages tant j'ai de choses à dire. Cette Thèse a été une aventure remplie de rebondissements, m'ayant fait à la fois vivre le meilleur comme le pire. J'ai traversé des déserts sans fins comme j'ai vécu des moments inoubliables.

En premier lieu, j'aimerais exprimer ma gratitude envers les membres du jury : les rapporteuses Juliette Marais et Audrey Giremus et les examinatrices Ni Zhu et Sandrine Mouysset. Je les remercie pour leur travail d'analyse de ma thèse, leur intérêt envers mes travaux et le partage de leur expertise, ce qui est un honneur connaissant leur notoriété dans leur domaine. Leur remarques sur le manuscrit ainsi que leur commentaires me permettront de m'améliorer dans de futurs travaux.

Dans un deuxième temps, je souhaiterais remercier l'entreprise Oktal-SE, avec tous mes collègues. Merci à Thierry et Jean de m'avoir fait confiance depuis le début et de ne m'avoir jamais laissé tomber même sous un confinement national. Sachez que je suis très heureux d'avoir rejoint la team Oktal-SE. L'avenir nous dira de quoi il est fait mais j'espère continuer longtemps avec vous, et pour ce début d'aventure dans le monde professionnel depuis décembre 2019 je vous remercie. Évidemment mes collègues rockeurs Arnold, Flo, Marc, César et Aude qui m'ont vu m'arracher de plus en plus les cheveux, à Manuela pour les atémis dans le bureau, évidemment Pierre pour ton soutien logistique avec mes récentes responsabilités. Merci à Anthony, Antoine, Corinne, Sandrine et Carole pour vos conseils sur la rédaction de thèse et le soutien moral, évidemment j'ai une pensée pour Lalaïna, Bertrand, Rachel, Yann, Alba, Anaël pour leur bonne humeur. Merci à tous.

J'aimerais aussi remercier l'ENAC. Je ne suis pas le dernier à cracher sur l'école parce que le self, parce que le service logement, parce que les partiels, parce que je suis surtout un râleur professionnel. Quand je regarde avec du recul tout ce que j'ai vécu depuis mon arrivée en 2016, je n'ai pas envie de partir. Je ne veux pas que cette période si incroyable de ma vie se termine. Depuis que je suis ici, certes le CHU de Ranguel me connaît un peu trop bien, mais l'ENAC m'a surtout permis de vivre une course de karting d'anthologie par équipe, une aventure inoubliable en Australie (oui Xavier que de merveilleux souvenirs qu'on ne cesse de se remémorer), des opportunités de concert au Capitole et dans des festivals, avec un souvenir mémorable au RVVGEA édition 2017. L'ENAC m'a aussi permis lors de la thèse de voyager (pardon de me déplacer en conférence) à Cologne et à Los Angeles où j'ai rêvé les yeux ouverts pendant une dizaine de jours. Pour tous ces souvenirs gravés dans ma mémoire et que je ne peux tous citer sous peine de dépasser les 200 pages de manuscrit, je vous remercie de m'avoir accepté et formé pendant toutes ces années. Eh oui René, je ne vais plus pouvoir t'enquiquiner au niveau des grillades.

C'est un remerciement assez cocasse mais je tiens à crier haut et fort mon amour pour AC/DC et Edward Lodewijk Van Halen. Avec les périodes que j'ai traversées, vous qui me lisez ne vous

rendez pas compte à quel point leur musique m'a permis de tenir le coup, jour après jour, me surmotiver dans les moments euphoriques, et m'encourager quand le moral était au plus bas. La musique a ce pouvoir, ne le sous-estimez pas. Une petite dédicace à Michel avec qui on a redécouvert et déchiffré les solos stratosphériques de nos mentors guitaristes.

Évidemment cette section de remerciements serait incomplète sans la mention de ma guilde Externels et de mes "petits enfants" au labo qui ont pris soin de Mémé Gonzalez surtout quand elle faisait des siennes. Merci à tous de m'avoir supporté pendant ces années. Sachez qu'elles ont été les plus belles de ma vie, et vous en êtes une raison. J'ai évidemment une grande pensée pour Denis et Artashes, avec qui nous avons partagé beaucoup de moments dans le bureau. Je ne pourrai pas citer tout le monde mais le coeur y est. Adrien, Alexis, Bastien, Céline, Clara, Éliot, Emmanuel, Geoffrey, Jean-Claude, Juju, Johan... Je vous fais un immense câlin parce que j'aimerais être aussi présent que vous l'avez été, à mes concerts, dans les cellules du prison Island, mais surtout dans les moments difficiles. J'aimerais également remercier mes amis, présents dans les meilleurs moments comme les passages à vide. Pour des raclettes au fromage de « pays », des randonnées avec des bières ou des spams de l'attraction Spider-man web, il y a de moments que je garderai jalousement.

Mourad, Momo, Djabrail, Jason, Simon, Thierry et Thierry (oui ils sont deux), Fabien, Joaquin, Maxime et j'en passe. Un immense merci au club de Jiu-Jitsu qui m'a permis de me forger un moral d'acier, et de me faire oublier, l'espace de quelques étranglements sanguins, les problèmes et frustrations que j'ai éprouvés lors de la thèse. Merci de m'avoir intégré dans la famille des combattants.

Un immense merci à mon groupe de hardrock Evil Phœnix (un peu de pub ne fait de mal à personne) avec qui j'ai vécu des choses incroyables. On a débuté avec des concerts inter-écoles et maintenant nos chansons sont disponibles sur les plateformes de streaming. Merci pour votre soutien et votre dévotion. Charles, Magali, Victor, vous êtes extraordinaires...

Vincent et Mathieu, après s'être perdu de vue l'espace de quelques années vous êtes revenus dans ma vie comme si vous aviez entendu un appel à l'aide. Une amitié de plus de 13 ans, à se bousculer dans des moshpits et à vivre d'autres moments inoubliables. Merci pour toutes ces années et les prochaines à venir. L'écurie Formule Aéro a été, est et restera invincible.

Yanis, toi qui a toujours été le batteur m'assurant sur tous mes concerts, tous mes faux pas et tout le temps. Toi avec qui j'ai partagé, je partage et je continuerai de partager des aventures FOLLES telles les musiciens de Mötley Crüe. Ton soutien a toujours été inébranlable, tu m'as toujours tiré vers le haut, les adjectifs mélioratifs me manquent quand je réfléchis à notre amitié. On jouera ensemble à 130 décibels jusqu'aux portes des Enfers, to Shout at The Devil!

Andréas, on a commencé la thèse quasiment ensemble et on va la finir aussi ensemble. Merci pour tous ces moments où on parlait de tout de rien, ces dingeries avec le Palmashow, le voyage à Marrakech, les démontadas du Real. Oh t'y as le kimono... Bref le S, merci d'avoir toujours été là, surtout quand ça n'allait pas.

J'aimerais aussi remercier avec toute ma gratitude mes directeurs de thèse Nicolas et Antoine ainsi que mon tuteur Christian. Je sais que j'étais loin d'être le plus performant des doctorants. Mais vous avez vu et su comment je marchais. Vous m'avez donné votre confiance, vous m'avez encouragé, vous avez aussi été très patient (pour mes écritures d'article en anglais notamment). Vous avez surtout été compréhensif, me donner plein de conseils qui m'ont quasiment toujours servis et surtout m'épauler dans les moments difficiles. Vous m'avez fait évoluer au cours de ces dernières années et c'est grâce à vous que je suis ce jeune homme aujourd'hui. C'est donc un

remerciement immense que je vous fais pour m'avoir accompagnés pendant ces années de thèse. Merci infiniment.

Malheureusement des personnes chères à mon cœur sont partis pour un monde meilleur pendant ces années charnières de ma vie. Elles ont ô combien participé à me rendre heureux et c'est donc naturellement que je leur dédie cette Thèse. Mi abuela, Angel, José y sobre todo mi compañero de Tute, mi Tio Amador. Ojala pudierais leer esas palabras para saber a que punto os hecho de menos. Os quiero un montón.

Pour finir cette section remerciements (il serait temps), je tiens évidemment à remercier mes Bros, Jérôme et Nico, que je supporte (ou qui me supportent) depuis tant d'années. Je remercie ma famille pour tout le soutien qu'elle m'a apportée. Evidemment, unos besos inimaginables para Mamy, mira como ha crecido el nieto! Un immense merci à ma sœur, qui a déjà publié son premier article (eh oui plus efficace que son frère et j'en suis très fier). Claro, esas ultimas palabras son para mis padres. El ser humano que soy, con esos valores, con esa valentia, con el amor para las otras personas, toda mi vida la debo a vosotros dos. Si daria a mis hijos el 1% de lo que me habeis dado durante toda mi vida seria una locura. Mis queridos padres, nunca os agradecere lo suficiente, por eso os lo digo : os quiero como nadie y lo hare hasta mi último latido de corazón. Y por supuesto, os dedico mi Tesis.

A mon Papy, mis abuelos y mi Tio Amador.

Table des matières

Introduction	1
Contexte et motivations	1
Démarche générale de la thèse	3
Contribution de la thèse	4
Organisation du mémoire	4
Productions scientifiques et logiciels	5
1 Réception d'un signal GNSS	7
1.1 Principe de positionnement GNSS	7
1.1.1 Pseudo-distance	9
1.1.2 Constellations et bande utilisée	10
1.1.3 Format d'un signal GPS L1 C/A	10
1.2 Réception et synchronisation du signal	12
1.2.1 Réception d'un signal	12
1.2.2 Corrélateur	12
1.2.3 Synchronisation de la réplique avec le signal reçu : acquisition et poursuite du signal	14
1.3 Phénomène de multitrajet	16
1.3.1 Définition d'un multitrajet	16
1.3.2 Modélisation d'un multitrajet	18
1.3.3 Impacts d'un multitrajet au niveau du signal en réception	19
1.3.4 Méthodes actuelles déployées contre le multitrajet	19
1.4 Conclusion	25
2 Génération de données synthétiques sous forme d'images	27
2.1 Présentation du logiciel	27
2.2 Modélisation du signal dans le générateur	29
2.3 Génération du multitrajet	30
2.4 Ajout du bruit et mise à l'échelle des images au C/N_0	32
2.5 Génération de sorties de corrélateur	33
2.6 Construction des différentes images des canaux I & Q	36
2.7 Ensemble de données des expériences	39
2.7.1 Dataset régulé par l'amplitude du multitrajet	39
2.7.2 Dataset régulé par le niveau de C/N_0	40
2.8 Conclusion	40
3 Estimation de paramètres par CNN	41
3.1 Introduction au Deep Learning	42
3.1.1 Machine Learning	42
3.1.2 Deep Learning	42
3.2 Réseaux de neurones	43
3.2.1 Architecture	43
3.2.2 Couches cachées	44
3.2.3 Couche de sortie	45
3.3 Apprentissage	45
3.3.1 Ensembles de données d'apprentissage	48
3.4 Réseaux de neurones à convolution	49
3.4.1 Bloc de convolution	50
3.4.2 Structure globale	50
3.5 Régression par CNN	54

3.6	Utilisation du <i>Machine Learning</i> dans le domaine GNSS	56
3.6.1	Données	56
3.6.2	<i>Features</i> ou caractéristiques	57
3.6.3	Algorithmes de <i>Machine Learning</i>	58
3.7	Conclusion	60
4	Estimation de paramètres de multitrajets sur des images de sorties de corrélateur	61
4.1	Architecture en VGG	62
4.2	Puissance de calcul mise en oeuvre	64
4.3	Protocole expérimental	64
4.4	Calcul MAE moyennée	65
4.4.1	Résultats selon les différentes amplitudes	65
4.4.2	Résultats selon les différents niveaux de C/N_0	69
4.4.3	Discussion	73
4.5	Conclusion	73
5	Méthode du <i>Distributional Loss</i> appliquée aux réseaux de neurones convolutionnels	75
5.1	Lien entre performances et <i>soft labelling</i>	76
5.1.1	Robustesse et généralisation	76
5.1.2	<i>Soft-Labelling</i>	77
5.2	<i>Distributional Loss</i> pour la régression	78
5.2.1	Rappel général régression	78
5.2.2	Principe du <i>Distributional Loss</i>	79
5.2.3	Divergence de Kullback-Leibler	79
5.2.4	Histogramme	79
5.2.5	Fonction perte : <i>Histogram Loss</i> (HL)	80
5.2.6	Choix de la distribution à priori	81
5.2.7	Loi de Von Mises	82
5.3	Application du <i>Distributional Loss</i> aux CNN	84
5.3.1	Calcul MAE moyennée et adaptée à l'architecture en <i>Distributional Loss</i>	84
5.3.2	Expériences	85
5.3.3	Analyse des performances de régression avec <i>Distributional Loss</i>	94
5.4	Conclusion	96
6	Estimation des paramètres de multitrajets générés par simulation	97
6.1	Logiciel de propagation de signaux GNSS : SE-Nav	97
6.2	Environnements simulés	98
6.2.1	Toulouse centre - place du Capitole	100
6.2.2	Quartier de Manhattan, New York City, États-Unis d'Amérique	101
6.2.3	Aéroport Roissy Charles-De-Gaulle, Paris Gonesse, France	103
6.3	Régression par CNN-HL	106
6.3.1	Production des images de sorties de corrélateur des multitrajets simulés	106
6.3.2	Résultats	108
6.3.3	Discussion	111
6.4	Conclusion	111
7	Perspectives	113
7.1	Élimination du multitrajet	113
7.2	Estimation de paramètres d'un seul multitrajet dans un environnement multi-interféré	114
7.3	Estimation des paramètres de plusieurs multitrajets en environnement multi-interféré	119
7.3.1	Régression de multi multitrajets : approche itérative	119
7.3.2	Régression de multi multitrajets : approche en simultané	119
7.4	Améliorations	121
7.5	Perspectives à plus long terme	122
	Conclusion	123
	Glossaire	127
	Bibliographie	129
A	Trajectoire d'un récepteur sur SE-Nav	139
B	Puissance du signal reçu	140

C	Influence de l'hyper-paramètre contrôlant la largeur de la distribution de probabilité dans le processus de <i>soft labelling</i>	141
D	Notes explicatives sur les réseaux de neurones	144
E	Robustesse envers un apprentissage avec <i>labels</i> ambigus	147
F	Comparaison de performances entre les distributions Gaussienne tronquées et selon une loi de Von Mises	148

Table des figures

1.1	Illustration du positionnement par satellites.	8
1.2	Schéma simplifié d'un récepteur GPS.	13
1.3	Répliques locales lors d'une estimation incorrecte de la phase de code du signal.	15
1.4	Répliques locales lors d'une estimation correcte de la phase de code du signal.	15
1.5	Matrice d'acquisition sur le code PRN 21.	17
1.6	Schéma d'une DLL sur la composante I d'un signal au sein d'un récepteur GPS.	17
1.7	Phénomène de multitrajet.	18
1.8	Effets d'un multitrajet sur un corrélateur.	20
2.1	Première chaîne de génération des données.	28
2.2	Schématisation de la chaîne de réception GPS.	30
2.3	Multitrajet n'impactant pas les sorties E-P-L.	32
2.4	Grille provisoire de construction des valeurs d'autocorrélation.	33
2.5	Grille provisoire de construction de la porteuse avec décalage Doppler et de phase.	34
2.6	Images I et Q issues de la multiplication entre les deux grilles provisoires.	35
2.7	Exemple de construction d'une image I de dimension 10x10.	36
2.8	Exemple d'une image I de dimension 10x10.	37
2.9	Différentes étapes de construction d'une image I.	37
2.10	Différentes étapes de construction d'une image Q.	38
2.11	Les différents <i>datasets</i> de multitrajets aléatoirement générés.	39
3.1	Représentation du paradigme d'un algorithme de <i>Machine Learning</i>	43
3.2	Image du <i>dataset</i> MNIST.	44
3.3	Couche de <i>softmax</i>	46
3.4	Apprentissage : étape de <i>feedforward</i>	47
3.5	Apprentissage : étape de <i>backforward</i>	48
3.6	Spécificité des basses couches d'un CNN.	50
3.7	Architecture d'un CNN.	51

3.8	Calcul d'une <i>feature map</i>	53
3.9	Couche de <i>max-pooling</i>	54
4.1	Division en blocs de convolution au sein d'un VGG.	62
4.2	Architecture du CNN pour des images 80x80.	63
4.3	Architecture du CNN pour des images 20x20.	63
5.1	Différentes représentations des <i>soft labels</i>	80
5.2	Image de compas de <i>label</i> ambiguë.	82
5.3	Distributions Gaussiennes tronquées pour deux caps proches modulo 2π	83
5.4	Distributions de Von Mises pour deux caps proches modulo 2π	84
5.5	Architecture du CNN-HL pour des images 80x80.	85
5.6	Deux cas de prédictions de <i>label</i> par CNN-HL.	89
5.7	$avg_{MAE}(\Delta\tau_{MP})$ en fonction du C/N_0 pour différentes résolutions d'images.	92
5.8	$avg_{MAE}(\Delta f_D)$ en fonction du C/N_0 pour différentes résolutions d'images.	93
5.9	$avg_{MAE}(\Delta\alpha_{MP})$ en fonction du C/N_0 pour différentes résolutions d'images.	93
5.10	$avg_{MAE}(\Delta\phi_{MP})$ en fonction du C/N_0 pour différentes résolutions d'images.	94
5.11	Distributions Gaussiennes tronquées pour différents écart-types.	95
5.12	Distributions de Von Mises pour différents écart-types.	95
6.1	Chaîne de génération des données.	99
6.2	Trajectoires récepteur dans le scénario aéroport Charles-De-Gaulle.	99
6.3	Scénario urbain Toulouse centre - place du Capitole, France.	100
6.4	Distribution des paramètres de multitrajets (Toulouse centre).	101
6.5	Nuages de points des paramètres de multitrajets (Toulouse centre).	102
6.6	Scénario urbain Manhattan à NYC, États-Unis d'Amérique.	102
6.7	Distribution des paramètres de multitrajets (quartier de Manhattan).	103
6.8	Nuages de points des paramètres de multitrajets (quartier de Manhattan).	104
6.9	Scénario rural avec mobiles centré sur l'aéroport Roissy-CDG, Gonesse, France.	104
6.10	Différentes trajectoires empruntées.	105
6.11	Distribution des paramètres de multitrajets (aéroport Charles-De-Gaulle).	106
6.12	Nuages de points des paramètres de multitrajets (aéroport Charles-De-Gaulle).	107
7.1	Phénomène de multitrajets dans le scénario Toulouse centre.	115
7.2	$avg_{MAE}(\Delta\tau_{MP})$ en environnement multi-interféré.	116
7.3	$avg_{MAE}(\Delta f_D)$ en environnement multi-interféré.	116
7.4	$avg_{MAE}(\Delta\alpha_{MP})$ en environnement multi-interféré.	117

7.5	$avg_{MAE}(\Delta\phi_{MP})$ en environnement multi-interféré.	118
7.6	Problème de parallélisation pour l'estimation de plusieurs multitrajets.	120
A.1	Interface du logiciel SE-Nav.	139
E.1	Illustration d'une ambiguïté de <i>label</i> sur test captcha.	147

Liste des tableaux

4.1	$avg_{MAE}(\Delta\tau_{MP})$ par le CNN-Reg pour différentes amplitudes de multitrajets.	67
4.2	$avg_{MAE}(\Delta f_D)$ par le CNN-Reg pour différentes amplitudes de multitrajets.	68
4.3	$avg_{MAE}(\Delta\alpha_{MP})$ par le CNN-Reg pour différentes amplitudes de multitrajets.	68
4.4	$avg_{MAE}(\Delta\phi_{MP})$ par le CNN-Reg pour différentes amplitudes de multitrajets.	69
4.5	$avg_{MAE}(\Delta\tau_{MP})$ par le CNN-Reg pour différents niveaux de C/N_0	70
4.6	$avg_{MAE}(\Delta f_D)$ par le CNN-Reg pour différents niveaux de C/N_0	71
4.7	$avg_{MAE}(\Delta\alpha_{MP})$ par le CNN-Reg pour différents niveaux de C/N_0	72
4.8	$avg_{MAE}(\Delta\phi_{MP})$ par le CNN-Reg pour différents niveaux de C/N_0	72
5.1	$avg_{MAE}(\Delta\tau_{MP})$ par le CNN-Reg et le CNN-HL pour différentes amplitudes de multitrajets.	87
5.2	$avg_{MAE}(\Delta f_D)$ par le CNN-Reg et le CNN-HL pour différentes amplitudes de multitrajets.	88
5.3	$avg_{MAE}(\Delta\alpha_{MP})$ par le CNN-Reg et le CNN-HL pour différentes amplitudes de multitrajets.	90
5.4	$avg_{MAE}(\Delta\phi_{MP})$ par le CNN-Reg, le CNN-HL- GT et le CNN-HL-VM pour différentes amplitudes de multitrajets.	91
6.1	$avg_{MAE}(\Delta\tau_{MP})$ sur les différentes ensembles de données.	108
6.2	$avg_{MAE}(\Delta f_D)$ sur les différentes ensembles de données.	109
6.3	$avg_{MAE}(\Delta\alpha_{MP})$ sur les différentes ensembles de données.	110
6.4	$avg_{MAE}(\Delta\phi_{MP})$ sur les différentes ensembles de données.	110
C.1	Influence de l'écart-type σ sur $avg_{MAE}(\Delta\tau_{MP})$	141
C.2	Influence de l'écart-type σ sur $avg_{MAE}(\Delta f_D)$	142
C.3	Influence de l'écart-type σ sur $avg_{MAE}(\Delta\alpha_{MP})$	142
C.4	Influence de l'écart-type κ sur $avg_{MAE}(\Delta\tau_{MP})$	143
D.1	Exemples de fonction d'activation utilisée en couche cachée.	144

Liste des tableaux

E1 Erreur d'estimation du cap en degrés par le CNN-Reg et deux distributions distinctes : Gaussienne tronquée et loi de Von Mises. 148

Introduction

Sommaire

Contexte et motivations	1
Démarche générale de la thèse	3
Contribution de la thèse	4
Organisation du mémoire	4
Productions scientifiques et logiciels	5

Contexte et motivations

Le positionnement par satellite est un procédé quotidiennement utilisé par de nombreux utilisateurs pour déterminer une position précise sur la surface terrestre. Les cadres d'utilisation civile pour cette technologie sont multiples : l'aviation, le maritime, les services de localisation sur smartphone, l'agriculture... Cette technologie, très répandue aujourd'hui, continue de croître : avec 3.6 milliards de récepteurs GNSS en 2015 et plus de 6 milliards en 2019, le rapport de l'Agence de l'Union européenne pour le programme spatial (EUSPA) établit des prévisions montant à 10 milliards pour 2029 [1]. Le positionnement par satellite s'appuie sur la mesure du temps de propagation entre des satellites et le récepteur, fournissant un calcul plus ou moins précis de la position du récepteur. Ce système est néanmoins sensible à plusieurs phénomènes d'interférence dont le multitrajets.

La problématique des multitrajets au sein d'un récepteur GNSS représente un défi majeur dans la réception des signaux provenant des satellites de navigation. Ce type d'interférence se produit lorsque les signaux émis par les satellites GNSS atteignent le récepteur après avoir été réfléchis par des obstacles tels que des bâtiments, des véhicules ou le relief du terrain. Ces signaux réfléchis sont des répliques du signal direct avec des variations de leurs paramètres comme leur retard ou leur phase. Leur présence en entrée du récepteur peut entraîner des erreurs de mesure de distance et par conséquent des erreurs de mesure de position, pouvant être significatives en milieu urbain dense. Les erreurs, non détectées par le système d'intégrité du GNSS, peuvent compromettre la sécurité des différentes applications s'appuyant sur sa fonctionnalité, pouvant avoir des conséquences lourdes en fonction de la criticité des applications comme en aviation civile. Enfin la multiplicité des multitrajets, c'est-à-dire la présence de plusieurs multitrajets différents à l'entrée du récepteur, rendent la correction de leurs erreurs d'autant plus complexe. Le multitrajets peut mener à une forte détérioration de la précision de positionnement [2, 3]. Il est prépondérant dans l'erreur de positionnement avec des signaux GNSS et représente l'une des causes d'erreur les plus difficiles à maîtriser notamment en raison de sa ressemblance avec le signal attendu. Afin d'atténuer les effets des multitrajets, les concepteurs de récepteurs GNSS ont développé plusieurs techniques s'appliquant à différents

endroits du récepteur. Parmi les méthodes déployées, certaines s'appliquent directement sur l'antenne, pour rejeter les multitrajets en provenance d'angles de réception indésirables par exemple. D'autres techniques avancées s'appliquent au sein du récepteur avec un traitement du signal identifiant et corrigeant les mesures issues des multitrajets avec l'analyse des sorties de corrélateur comme les techniques du *narrow correlator* [4], la *Multipath Insensitive Delay-Locked Loop* (MIDLL) [5] et l'utilisation de filtres à particules [6]. Malgré le développement de nombreuses techniques, les environnements urbains denses et les zones avec beaucoup d'obstacles continuent de poser des défis importants en matière de réception GNSS précise. De nombreuses recherches ont été menées afin de détecter, diminuer ou supprimer l'impact du multitrajet au sein d'un récepteur.

Avec la montée en puissance des algorithmes de *Machine Learning* et notamment des avancées depuis les années 2000, la communauté du GNSS s'intéresse de plus en plus à l'utilisation de ces algorithmes à des fins d'amélioration des performances du système de positionnement. Des algorithmes de *Machine Learning* comme les *Support Vector Regression* (SVR) [7] et les *Support Vector Machine* (SVM) [8] ont été testés pour diminuer l'impact des multitrajets. Des algorithmes de *Deep Learning* ont également été utilisés afin de répondre au problème des multitrajets [9]. Parmi les travaux effectués, on peut citer l'utilisation d'un perceptron multicouches pour la diminution de l'impact des multitrajets sur des signaux GNSS pour des applications *Low Earth Orbit* (LEO) [2]. Dans [10] l'étude menée par Suzuki et al. a montré qu'un réseau de neurones permettait d'obtenir de meilleures performances pour la détection des multitrajets qu'un SVM. Parmi les réseaux de neurones utilisés, les réseaux de neurones à convolution attirent particulièrement l'attention. Les CNN se différencient des réseaux de neurones « conventionnels » de type perceptron multicouches par la présence de couches de convolution à l'entrée du réseau. Leur structure, automatisant la construction de points d'intérêts informatifs (également appelés *features*) leur confère une grande efficacité quant à l'analyse de données sous la forme d'images. Principalement connus pour leurs résultats dans des tâches de classification avec l'exemple d'application sur la base de données de chiffres manuscrits *Mixed National Institute of Standards and Technology* (MNIST) dans [11], les CNN affichent de bonnes performances sur des tâches de régression, c'est-à-dire dans le cas d'une estimation de variables continues à partir d'images en entrée. Les CNN ont été utilisés à des fins de régression dans plusieurs domaines comme l'estimation d'âge de personnes dans [12], l'estimation de position d'une tête ou d'un corps humain dans un espace 3D [13, 14] ainsi que sur des images générées par Imagerie par Résonance Magnétique (IRM) dans [15] et des tenseurs d'images en rayons X dans [16]. Dans le domaine du GNSS, le CNN a aussi été testé, notamment concernant la détection de multitrajets dans [3, 17, 18, 19, 9].

Dans le cadre de cette thèse, nous allons appliquer des réseaux *Visual Geometry Group* (VGG) (variante d'une architecture CNN présentée dans [20]) sur des images de sorties de corrélateur afin de supprimer la composante de multitrajet. Dans le cadre d'estimation de multitrajet dans [21], les auteurs introduisent une nouvelle représentation des signaux GNSS en sortie de corrélateur sous la forme d'une paire d'images 2D. Ces images sont calculées à partir des sorties de corrélateur avec des sorties supplémentaires ayant pour but de donner à l'algorithme le maximum d'informations concernant les signaux présents. Le niveau de détection de multitrajet atteint est meilleur que de précédentes recherches [22, 10] par la simple utilisation d'images en sortie de corrélateur. Naturellement nous décidons d'utiliser ces données pour entraîner nos algorithmes de *Deep Learning* pour cette tâche de régression.

La tâche ambitieuse de suppression de la composante du multitrajet passe par l'estimation de celui-ci. L'estimation d'un multitrajet peut être résumée à l'estimation de quatre paramètres : son retard, sa fréquence Doppler, son amplitude et sa phase. Le signal total en sortie de cor-

rélateur étant, par linéarité de la chaîne de réception par rapport aux composantes du signal, l'addition entre la composante du signal direct, du multitrajet et de bruit, l'estimation précise du multitrajet permettra la soustraction de ses effets. Dans le cadre de la thèse, on supposera le récepteur en phase de poursuite, c'est-à-dire ayant une estimation précise des paramètres du signal direct. Jusqu'au chapitre 6, on considérera la présence d'un seul multitrajet avec le signal direct pour des raisons de preuve de concept. L'application d'un CNN sur des images de sortie de corrélateur classe notre approche dans la catégorie des méthodes *post*-corrélacion pour l'estimation de multitrajet.

Le CNN étant un réseau de neurones, son apprentissage nécessite une quantité conséquente de données. Dans le cadre d'un apprentissage supervisé, nous avons besoin d'une grande base de données réelles contenant des multitrajets labellisés. La labellisation des multitrajets étant aussi compliquée voire plus que la détection de ces derniers, une alternative, choisie dans [21] est de générer un *dataset* à partir de signaux synthétiques. Faisant l'objet du chapitre 2, le générateur de données est capable de calculer les sorties de corrélateur correspondant à un signal *Line-Of-Sight* (LOS) en poursuite en présence d'un multitrajet dont les paramètres sont aléatoirement générés. Les images seront générées à différents niveaux de bruit pour différentes résolutions. Dans un objectif de données labellisées plus réalistes, le générateur d'images sera couplé à un logiciel de propagation de signaux GNSS en environnement 3D. Présenté dans le chapitre 6, ce logiciel permet de recréer des conditions d'apparition du phénomène de multitrajet et d'en récupérer ses paramètres. Une fois collectés, ces paramètres sont ensuite utilisés comme base pour la génération des multitrajets dans le générateur.

La tâche d'estimation des paramètres de multitrajet par régression est complexe : en effet les images en sortie du générateur sont les représentations des sorties de corrélateur des canaux I et Q. Contrairement aux images *Red-Green-Blue* (RGB) habituelles qui sont des répliques d'une même image sous un canal de couleur différente, les images I et Q représentent deux composantes distinctes et corrélées du signal. L'information concernant le multitrajet peut alors être répartie de manière plus ou moins inégale parmi les deux images, transformant la tâche de régression en une tâche de régression multicanaux. Pour améliorer les performances de généralisation du modèle VGG, nous appliquerons la méthode du *Distributional Loss*, méthode de *soft labelling* introduite par Imani et al. dans [23]. Cette technique vise à améliorer les performances de régression en substituant les *labels* réels par des distributions de probabilité. La tâche de régression sur variables continues devient alors une tâche de régression sur distribution. La comparaison des distributions et la rétropropagation des erreurs dans le réseau s'effectuera à partir d'un calcul de dissimilarité s'appuyant sur la divergence de Kullback-Leibler. Nous introduirons une nouvelle distribution adaptée à la labellisation de variables angulaires, améliorant ainsi la précision de leur estimation par le réseau de neurones.

Démarche générale de la thèse

La thèse a eu pour ligne directrice l'exploration de techniques d'estimation des paramètres de multitrajets. Dans un cadre autorisant une utilisation d'un nombre de sorties plus important que sur un corrélateur conventionnel, nous avons exploré une méthode possible utilisant la capacité d'analyse d'algorithmes de *Deep Learning* à des fins de régression sur images. Nous avons également testé une méthode d'amélioration des performances d'estimation, transformant le problème de régression en un problème d'estimation de densités de probabilités. Enfin dans une démarche visant à s'approcher de données réelles, nous avons réalisé l'interface entre un générateur d'images de sorties de corrélateur et un simulateur de propagation de signaux

industriel afin de générer des données de multitrajets reproduits par simulation et aux paramètres physiquement réalistes.

Contribution de la thèse

Cette section résume les contributions scientifiques et logicielles de la thèse :

- Méthode d'estimation des paramètres de multitrajets à partir d'images de sorties de corrélateur par *Deep learning*,
- Technique de *soft labelling* à partir d'une fonction perte *Histogram Loss* pour l'amélioration des performances de généralisation d'un CNN,
- Introduction d'une distribution adaptée pour des paramètres périodiques,
- Tests dans des environnements physiquement réalistes.

Organisation du mémoire

Le mémoire est divisé en sept chapitres. Le chapitre 1 porte sur le positionnement par signaux GNSS. Après quelques rappels concernant le calcul de pseudo-distances, nous détaillerons le fonctionnement du corrélateur, étage du récepteur qui est au cœur de la génération des images utilisées. Le chapitre se termine sur une définition du phénomène de multitrajet avec une analyse d'une liste non exhaustive de techniques développées pour diminuer l'impact des multitrajets après l'étage du corrélateur en décrivant la méthode étudiée, ses performances et ses principaux désavantages. Le chapitre 2 détaille le générateur de sorties de corrélateur synthétique. Le processus de construction des images de sorties de corrélateur y est décrit ainsi que les modèles utilisés pour les signaux et les multitrajets. Ce chapitre présente également les différents ensembles de données qui serviront aux chapitres 4 à 7. Le troisième chapitre introduit la régression par réseau de neurones à convolution, technique d'estimation de paramètres choisie dans cette thèse. Le chapitre se termine sur un bilan des techniques de *Machine Learning*, notamment appliquées au problème de détection de multitrajets. Les chapitres suivants concentrent les expériences liées aux travaux de recherche de cette thèse. Le chapitre 4 expose les premières expériences de régression sur les paramètres de multitrajet par réseau de neurones à convolution. Le chapitre 5 présente le principe de robustesse par *Distributional Loss*, technique de *soft labelling* et son application dans notre cadre d'étude. Un exemple sur la régression de cap sur des images de compas magnétiques illustre la méthode de ce chapitre pour une distribution nouvellement introduite, la distribution selon une loi de Von Mises, adaptée pour des données de *labels* périodiques. Les différentes distributions utilisées sont présentées et les performances du CNN s'appuyant sur les distributions sont comparées avec les résultats obtenus en chapitre 4. Le chapitre 6 définit une nouvelle chaîne de génération de données où le générateur de données présenté au chapitre 2 est alimenté par un logiciel de propagation de signaux fourni par l'entreprise Oktal-SE, permettant de générer des données de paramètres de multitrajet plus réalistes pour une application des algorithmes du chapitre 5. Le chapitre présente les résultats obtenus par les expériences sur différents ensembles de données introduits dans ce même chapitre. Le chapitre 7 présente des travaux préliminaires réalisés sur l'estimation d'un multitrajet en présence d'autres multitrajets. Nous nous intéressons également aux différentes alternatives possibles pour la suppression des composantes de multitrajet et discutons de points d'amélioration des différentes techniques utilisées dans les travaux de cette

thèse. Le chapitre est suivi d'une conclusion générale résumant les travaux et résultats de la thèse.

Organisation de la bibliographie

Afin de faciliter la lecture du manuscrit, la bibliographie n'est pas concentrée en un seul chapitre mais répartie en trois sections qui sont :

- Section 1.3.4 (chapitre 1) : état de l'art sur des techniques non *Machine Learning* pour détecter/estimer les multitrajets au niveau du corrélateur,
- Section 3.6 (chapitre 3) : état de l'art sur des techniques *Machine Learning* pour détecter/estimer les multitrajets au niveau du récepteur,
- Section 5.1.1 (chapitre 5) : robustesse d'un algorithme de *Deep Learning*.

Productions scientifiques et logiciels

Cette section liste les différentes productions académiques et les logiciels développés autour des travaux de recherche effectués pendant toute la thèse.

Publications scientifiques

Articles

Gonzalez, T., Blais, A., Couëllan, N., & Ruiz, C. (2022). Distributional loss for convolutional neural network regression and application to GNSS multi-path estimation, soumis, preprint disponible à l'adresse <https://arxiv.org/abs/2206.01473>

Conférences

- (avec acte) ITM/PTTI 2023 - Los Angeles, Californie (USA) : Gonzalez, T., Blais, A., Couëllan, N., & Ruiz, C. (2023, January). Multipath parameters estimation in physically based synthetic environment using robust deep neural regression. In Proceedings of the 2023 International Technical Meeting of The Institute of Navigation (pp. 808-822),
- ION GNSS 2021+ - St Louis, Missouri (USA) : « Multipath Parameters Estimation with CNN-regression on GNSS Correlator Outputs » : Gonzalez, T., Blais, A., Couëllan, N., & Ruiz, C.,
- EASA conference 2023 - Cologne (Allemagne) : « Detection, Classification, Characterization and Reduction of satellite navigation signals degradations by Machine Learning and application in synthetic environment » : Gonzalez, T., Blais, A., Couëllan, N., & Ruiz, C..

Séminaires

- Séminaire mensuel de l'axe de recherche Intelligence Artificielle (IA) de l'ENAC,
- Présentation sous format 180 secondes à la Journée des doctorants de l'ENAC 2022 (1^{er} prix de la meilleure présentation de thèse ENAC pour l'édition 2022),
- *User Meeting Group* 2021, 2022 et 2023, séminaire industriel organisé à Oktal-SE.

Produits logiciels

- Amélioration du générateur d'images de sorties de corrélateur de l'équipe Telecom du laboratoire ENAC : nous avons apporté plusieurs modifications au générateur. Générant des données destinées à alimenter des algorithmes de détection de multitrajets, le générateur a été amélioré avec l'ajout d'une *labellisation* automatique des données pour des algorithmes d'estimation de multitrajets et la capacité de génération de plusieurs multitrajets sur une même image.
- Interface simulateur de propagation - générateur de sorties de corrélateur : avec l'objectif d'une génération de multitrajets de paramètres plus réalistes, nous avons réalisé l'interfaçage entre le générateur de sorties de corrélateur et le logiciel SE-Nav de la société Oktal-SE. Ce logiciel permet, par l'intermédiaire de simulations de propagation de signaux satellites en environnement 3D simulé, la reproduction du phénomène de multitrajet. Les paramètres de multitrajets physiquement réalistes sont ainsi utilisés comme point de départ pour la génération d'images de sorties de corrélateur,
- Création de plusieurs scénarios de simulation : construction de plusieurs scénarios de simulation. Pour diversifier les multitrajets collectés au cours des simulations, nous avons réalisé les simulations sur différents environnements 3D (centre ville, zone urbaine dense, zone rurale) avec des trajectoires différentes, des obstacles et mobiles de cinétique très importante (avion de ligne au décollage) ainsi que des constellations de satellites dans différentes configurations. Ces *datasets* pourront par la suite être utilisés pour l'estimation de plusieurs multitrajets à la fois ainsi que pour la dynamique des multitrajets (leur évolution dans le temps) qui n'a pas été prise en compte lors de nos travaux,
- Génération de plusieurs *datasets* de multitrajets, de paramètres aléatoires ou issus des simulations du simulateur SE-Nav : ces *datasets* sont différents par leur nombre de multitrajets par images, le niveau de bruit, les amplitudes des multitrajets et la dimension des images de sorties de corrélateur. Chaque *dataset* permet d'exercer des entraînements d'algorithmes de *Deep Learning* pour l'estimation des paramètres de retard, fréquence Doppler, amplitude et phase du multitrajet,
- Réseaux de neurones à convolution avec estimation de densités de probabilité : nous avons développé sous Python et Tensorflow 2.4.1 [24] des réseaux de neurones à convolution pour des tâches d'estimation de paramètres de multitrajets par régression. Nous avons mis en place un programme capable de lancer de manière automatique et séquentielle des expériences d'estimation sur des réseaux de neurones de poids initiaux différents sur des *datasets* différents (les algorithmes de *Deep Learning* ont recours à des processus aléatoires pendant leur apprentissage et sont vulnérables à des divergences de la fonction de perte et blocages dans des *minima* locaux. Avec des expériences sur des structures de poids et biais différents ainsi que des ensembles d'entraînement différents, on peut observer la stabilité de ses performances). La structure de l'algorithme a été adaptée afin d'effectuer la prédiction de distributions, avec la génération des différentes densités de distribution utilisées. La fonction de perte, propre au problème d'estimation de distribution, s'appuie sur le calcul de la divergence de *Kullback-Leibler* et a été vectorisée pour optimiser le temps de calcul lors de l'apprentissage. Le code est disponible sous le dépôt git [25].

Chapitre 1

Réception d'un signal GNSS

Sommaire

1.1 Principe de positionnement GNSS	7
1.1.1 Pseudo-distance	9
1.1.2 Constellations et bande utilisée	10
1.1.3 Format d'un signal GPS L1 C/A	10
1.2 Réception et synchronisation du signal	12
1.2.1 Réception d'un signal	12
1.2.2 Corrélateur	12
1.2.3 Synchronisation de la réplique avec le signal reçu : acquisition et poursuite du signal	14
1.3 Phénomène de multitrajet	16
1.3.1 Définition d'un multitrajet	16
1.3.2 Modélisation d'un multitrajet	18
1.3.3 Impacts d'un multitrajet au niveau du signal en réception	19
1.3.4 Méthodes actuelles déployées contre le multitrajet	19
1.4 Conclusion	25

Ce chapitre introduit le concept de positionnement par satellites et sa vulnérabilité face au phénomène de multitrajet. La première partie est consacrée aux calculs de distances entre récepteur et satellites ainsi qu'à la description du signal émis par les satellites. La deuxième partie se focalisera sur le récepteur, avec une analyse détaillée du corrélateur, permettant de détecter et de verrouiller le signal satellite attendu parmi les autres signaux et bruits reçus. Cet étage du récepteur est fondamental dans la thèse puisque toutes les données manipulées au cours des expériences seront issues des sorties du corrélateur. C'est également à cet étage précis que le multitrajet (faisant l'objet de la dernière section de ce chapitre) a un effet non négligeable, pouvant conduire à une sévère détérioration des performances de positionnement. La dernière partie introduit donc le problème du multitrajet avec sa définition et son effet sur l'étage du corrélateur. Le chapitre se conclut sur une liste non exhaustive de travaux de recherche portant sur la réduction de l'impact du multitrajet dans un récepteur GNSS.

1.1 Principe de positionnement GNSS

Le principe de positionnement par GNSS repose sur la mesure de pseudo-distances entre satellites et récepteur. La pseudo-distance n'est pas une distance directement mesurée mais

1.1. Principe de positionnement GNSS

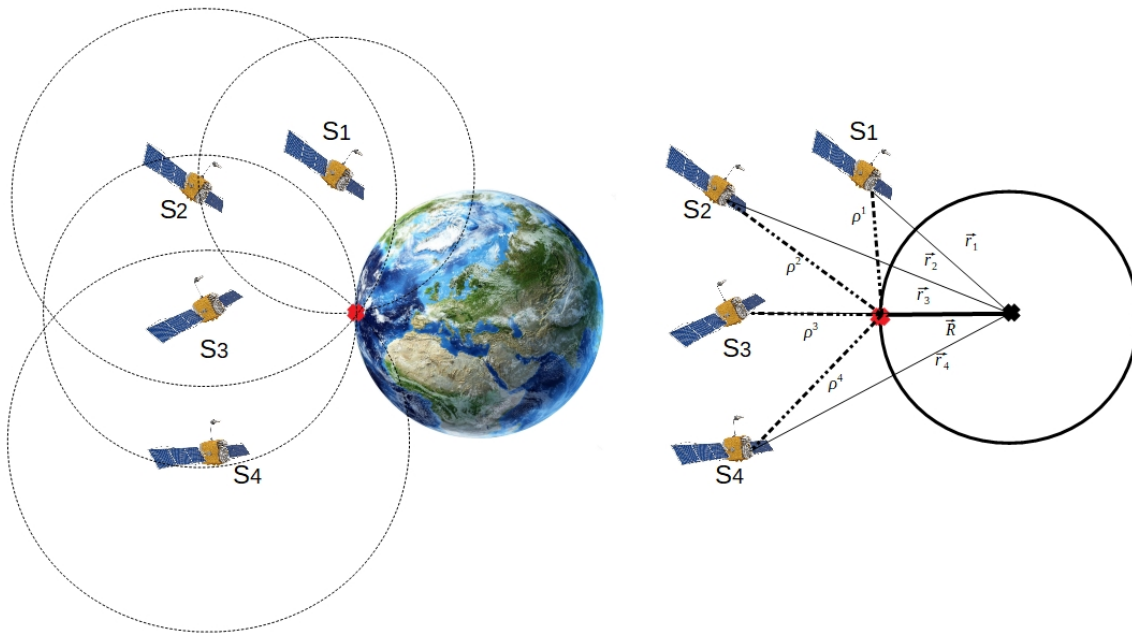


FIGURE 1.1 – Illustration du positionnement par satellites. Les pseudo-distances $(\rho^i)_{i \in \llbracket 1,4 \rrbracket}$ correspondent aux rayons des sphères des satellites $(s_i)_{i \in \llbracket 1,4 \rrbracket}$, dont l'intersection permet de déterminer la position du récepteur. En posant $\vec{R} = (x, y, z)$ la position inconnue du récepteur et $\vec{r}_j = (x_j, y_j, z_j)$ la position connue du satellite j , $\forall j \in \llbracket 1,4 \rrbracket$, $\rho^j = \|\vec{r}_j - \vec{R}\|$.

indirectement calculée par la mesure d'un temps de propagation (détails section 1.1.1). Les pseudo-distances permettent de construire des sphères dont l'intersection détermine la position du récepteur (illustration 1.1). Le signal émis par les satellites est diffusé de manière continue (on parle d'onde continue [26, 27]) sans sollicitation de la part du récepteur. La réception des divers signaux satellites sera davantage détaillée en section 1.2. Le système de positionnement se décompose en trois segments :

- le segment spatial, composé d'une ou plusieurs constellations de satellites, émetteurs des signaux utilisés pour le positionnement. Le signal contient des données renseignant sur la position du satellite ainsi que l'instant t précis auquel le signal reçu a été émis. Ces données sont nécessaires au calcul des pseudo-distances,
- le segment sol, chargé de surveiller les signaux GNSS et l'intégrité des satellites, de calculer les paramètres d'orbitographie et d'envoyer les messages de navigation vers les satellites. Ce segment est composé de stations sols et de centres de contrôle,
- le segment utilisateur, composé de terminaux comme le récepteur, chargés de collecter le signal (l'extraire des signaux parasites tels que les différents bruits et signaux en provenance d'autres satellites) afin de mesurer le temps de propagation du signal depuis le satellite émetteur jusqu'à l'entrée du récepteur. C'est cette dernière entité qui est impactée par le phénomène de multitrajet.

En plus de ces trois segments, le système de positionnement est impacté par le milieu de propagation dans lequel les signaux évoluent. Il comprend les différentes couches atmosphériques traversées comme l'ionosphère et la troposphère. Il concerne également l'environnement proche du récepteur qui est à l'origine du phénomène de multitrajet.

1.1.1 Pseudo-distance

La pseudo-distance est la distance à priori parcourue par un signal entre un satellite émetteur et un récepteur. On définit alors la distance géométrique ρ_g^i entre un satellite s_i et un récepteur r comme étant

$$\rho_g^i = c(t_{r,T}^i - t_{s,T}^i) = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}$$

où

- $t_{r,T}^i$ correspond au temps t_{GPS}^1 de réception du signal en provenance du satellite s_i ,
- $t_{s,T}^i$ correspond au temps t_{GPS}^1 d'émission du signal depuis le satellite s_i .
- (x_i, y_i, z_i) sont les coordonnées du satellite s_i dans le repère *Earth-Centered, Earth-Fixed* (ECEF),
- (x, y, z) sont les coordonnées du récepteur r dans le repère ECEF.

En réalité, les temps d'émission et de réception mesurés contiennent une erreur due à un décalage d'horloge : les horloges satellites² et récepteur ne sont pas synchronisées avec le temps t_{GPS} . Il y a le décalage d'horloge du satellite s_i que l'on notera Δt^i et le décalage d'horloge récepteur que l'on notera Δt_r . Les nouveaux temps d'émission t_s^i et de réception t_r^i sont donc :

$$\begin{aligned} t_r^i &= t_{r,T}^i + \Delta t_r \\ t_s^i &= t_{s,T}^i + \Delta t^i \end{aligned}$$

En posant $\Delta t_r^i = \Delta t_r - \Delta t^i$ le décalage de l'horloge récepteur par rapport à l'horloge à bord du satellite s_i ainsi que $dt^i = t_{r,T}^i - t_{s,T}^i$ le temps de propagation du signal du satellite s_i au récepteur, la pseudo-distance mesurée est :

$$\rho_m^i = c(dt^i + \Delta t_r^i)$$

En plus des inconnues (x, y, z) correspondant aux coordonnées 3D dans le repère ECEF [26] (voir figure 1.1), les décalages d'horloge du récepteur et du satellite doivent être déterminés afin de pouvoir effectuer un calcul de pseudo-distance correct. Le message de navigation envoyé contenant les corrections de décalage d'horloge du satellite, seul le décalage d'horloge du récepteur Δt_r reste inconnu. Finalement, le modèle de calcul de pseudo-distances est décrit par l'équation suivante :

$$\rho^i = \underbrace{c dt^i}_{\rho_g^i} + c \Delta t_r^i + \epsilon_i$$

où ϵ_i représente les erreurs de mesure autres que les décalages d'horloge sur le signal en provenance du satellite s_i (erreur troposphérique, erreur ionosphérique, effet relativiste, bruit de mesure du récepteur...) [26].

Le calcul de position d'un récepteur nécessite l'utilisation de quatre satellites, permettant de résoudre le système d'équations (P) suivant :

$$(P) : \begin{cases} \rho_1 = \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} + c \Delta t_r^1 + \epsilon_1 \\ \rho_2 = \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} + c \Delta t_r^2 + \epsilon_2 \\ \rho_3 = \sqrt{(x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2} + c \Delta t_r^3 + \epsilon_3 \\ \rho_4 = \sqrt{(x_4 - x)^2 + (y_4 - y)^2 + (z_4 - z)^2} + c \Delta t_r^4 + \epsilon_4 \end{cases}$$

Remarque : Dans le cas où le nombre de satellites utilisés est strictement supérieur à 4, une possibilité est d'appliquer une méthode de résolution par les moindres carrés [26].

1. Temps de référence utilisé pour le calcul des pseudo-distances [26]. Il est défini au sol par des horloges atomiques.

2. Les horloges satellites sont également décalées entre elles [28].

1.1.2 Constellations et bande utilisée

Le positionnement par signaux GNSS s'appuie sur plusieurs constellations comme la constellation *Global Positioning System* (GPS), GLONASS (système de positionnement par satellites d'origine soviétique et actuellement utilisé par la Russie), Beidou (système de positionnement par satellites chinois) et Galileo. Cette dernière a été développée par l'*European Space Agency* (ESA) au début des années 2000 et est composée de 30 satellites (27 en utilisation et 3 en réserve) répartis sur trois plans orbitaux (56°) de rayon 29600 km (*Medium Earth Orbit* (MEO)). Galileo permet de réaliser diverses missions parmi lesquelles un service de positionnement, de localisation d'événements de détresse ainsi que des performances de positionnement compétitives avec les autres constellations [29]. Bien que le sujet de cette thèse porte sur les signaux GNSS en général, les expériences réalisées lors de ces travaux se sont exclusivement appuyées sur l'estimation de multitrajets de signaux GPS L1 *Coarse/Acquisition* (C/A) pour plusieurs raisons :

- le principe de corrélation pour le positionnement est le même avec tous les signaux GNSS ou GPS L1 C/A,
- la bande L1, de fréquence centrale $f_{L1} = 1575.42$ MHz, est utilisée pour différents services comme le *Standard Positioning Service* (SPS) [30], fournissant des données *Positioning, Navigation and Timing* (PNT) à des fins d'applications civiles. Les signaux L1 C/A, également appelés *legacy signals*, ont l'avantage d'être transmis par tous les satellites GPS opérationnels et reçus par tous les récepteurs,
- ce signal en particulier a fait l'objet de nombreuses recherches ces dernières années, faisant du signal L1 C/A une référence en termes de positionnement par satellite [31, 32, 33].

1.1.3 Format d'un signal GPS L1 C/A

1.1.3.1 Structure du signal

La construction du signal GPS L1 C/A repose sur trois éléments [34] :

1. une porteuse de fréquence centrale $f_{L1} = 1575.42$ MHz,
2. un code pseudo-aléatoire, également appelé *Pseudo-Random Noise* (PRN) et propre à chaque satellite, d'une fréquence de 1.023 MHz. La particularité de ce code sera l'objet de la sous-section suivante,
3. un message de navigation avec un débit de 50 bits/s. Il est composé d'une trame de 1500 bits, divisée en 5 sous-trames de 300 bits. Il contient les corrections d'horloges, des informations sur la santé du satellite émetteur, les données d'éphémérides (paramètres du modèle d'orbitographie permettant le calcul de la position), l'*User Range Accuracy* (URA) (borne sur l'écart-type de l'erreur d'horloge et les données d'éphémérides), les données d'almanach (paramètres de l'ensemble de la constellation) et les données ionosphériques sur tous les satellites [35]. Ce message est transmis par le biais d'une modulation *Bi-Phase Shift Keying* (BPSK).

En réalité, comme montré dans l'équation 1.1, le signal envoyé est composé de deux éléments. Le premier correspond au code P, réservé à un usage strictement militaire, dont le décodage nécessite l'utilisation d'une clé. Le deuxième terme du signal correspond à celui du code public C/A, à l'origine destiné à permettre une pré-acquisition du code P et aujourd'hui utilisé par des applications civiles. La thèse portera uniquement sur le second terme $S_{C/A}(t)$, dont la formule est explicitée dans l'équation suivante :

$$S_{LI}(t) = \underbrace{\sqrt{2C_P}P(t)D(t) \cos(2\pi f_{L1}t + \phi_{err}(t))}_{S_{LI,P}(t)} + \underbrace{\sqrt{2C_{C/A}}C(t)D(t) \sin(2\pi f_{L1}t + \phi_{err}(t))}_{S_{C/A}(t)} \quad (1.1)$$

avec

- C_P la puissance du signal de code P émis,
- $C_{C/A}$ la puissance du signal émis de code C/A ,
- $P(t)$ le code P à l'instant t , de cadence 10.23 Mbit/s,
- $C(t)$ le code C/A à l'instant t , de cadence 1.023 Mbit/s,
- $D(t)$ le bit du message de navigation à l'instant t , de cadence 50 bits/s,
- $\phi_{err}(t)$ représentant un faible terme de bruit de phase ainsi qu'une composante de dérive d'oscillateur [36].

1.1.3.2 Code d'acquisition C/A

Le code C/A est une séquence binaire envoyée de manière continue et de périodicité d'une milliseconde [37]. La séquence binaire caractérisant un code C/A est composé de 1023 bits. Chaque bit a une durée de $\frac{1}{1023}$ ms que l'on appelle aussi période de *chip* et notée T_c [38]. Appartenant à la catégorie des codes de *Gold* [39], cette séquence PRN, propre à chaque satellite de la constellation, possède des propriétés d'auto-corrélation et de corrélation croisée permettant au récepteur d'identifier le satellite à l'origine du signal reçu et de le synchroniser avec sa réplique locale. La première de ces propriétés permet d'imposer une corrélation quasi nulle entre deux codes de *Gold* différents [40]. En posant $r_{i,j}$ la corrélation entre les codes des satellites s_i et s_j , N_s l'ensemble des satellites de la constellation et G_i le code de *Gold* du satellite s_i :

Propriété 1 : Corrélation croisée quasiment nulle

$$\forall (i, j) \in \llbracket 1, N_s \rrbracket^2 \text{ avec } i \neq j : r_{i,j}(m) = \sum_{k=1}^{1023} G_i(k)G_j(k+m) \approx 0 \quad \forall m \in \mathbb{Z}$$

Cette propriété d'orthogonalité permet l'envoi en simultané de messages différents. Une deuxième propriété concerne la corrélation quasi-nulle si la réplique locale du signal attendue n'est pas correctement alignée au code reçu [40].

Propriété 2 : Autocorrélation quasiment nulle exceptée pour un décalage à zéro

$$\forall i \in \llbracket 1, N_s \rrbracket, r_{i,i}(m) = \sum_{k=1}^{1023} G_i(k)G_i(k+m) \quad \forall m \in \mathbb{Z}$$

$$\forall i \in \llbracket 1, N_s \rrbracket, r_{i,i}(m) = \sum_{k=1}^{1023} G_i(k)G_i(k+m) \left\{ \begin{array}{ll} \approx 0 & \forall m \in \mathbb{Z}^* \\ = 1023 & \text{si } m = 0 \end{array} \right.$$

Un autre avantage à l'utilisation des code PRN est lié à la longueur de leur séquence : le gain de démodulation est très important. Grâce à la corrélation sur ces longues séquences, l'impact du bruit et des interférences est réduit à l'étape de corrélation [29]. Cependant nous constaterons à la section 1.3 que certains types d'interférences persistent et sont au cœur de la problématique de cette thèse.

1.2 Réception et synchronisation du signal

Nous avons vu dans la section 1.1 qu'un récepteur GPS était alimenté de manière continue par des signaux en provenance de satellites. Le récepteur reçoit également, en plus des signaux utiles, des interférences (dont les multitrajets) se soldant par des altérations parfois non négligeables du fonctionnement du récepteur. Afin d'extraire le signal utile des parasites, le récepteur génère une réplique locale du signal attendu. Cette réplique possède un retard, une fréquence Doppler ainsi qu'une phase propre. L'objectif du récepteur est de réaliser la synchronisation entre la réplique et le signal reçu : cela revient à aligner les paramètres de retard, fréquence Doppler et phase de la réplique sur les paramètres de retard, fréquence Doppler et phase du signal reçu. Une fois la synchronisation opérée, le décodage du message de navigation est possible. La récupération de l'information contenue dans le message de navigation (voir détail section 1.1.3.1) permettra notamment le calcul de la pseudo-distance [41]. Dans cette section, nous nous proposons d'analyser en détail le fonctionnement d'un corrélateur, outil calculant la synchronisation entre le signal entrant et une réplique locale.

1.2.1 Réception d'un signal

La chaîne de réception peut être décomposée en deux sections : une partie préliminaire préparant le signal à des opérations de traitement et la partie traitement du signal. La première section est composée d'éléments *hardware* : une antenne réceptrice, un bloc de pré-filtrage, un bloc de pré-amplification, un convertisseur de fréquence et un convertisseur *Analog to Digital Converter* (ADC) [26, 42]. Dans cette première section, le signal reçu est amplifié, filtré, converti en fréquence intermédiaire, amplifié et numérisé [43]. La deuxième section du récepteur, comme illustrée dans le schéma 1.2, est composée d'éléments traitant le signal numérisé et permettant la démodulation du message de navigation véhiculé par le signal [41]. Une fois les informations du message de navigation décodées, les corrections à appliquer comme les décalages d'horloge sont connues et permettent un calcul correct de la pseudo-distance. La démodulation est rendue possible grâce la synchronisation entre une réplique locale du code PRN et le signal reçu dans l'étage corrélateur.

1.2.2 Corrélateur

1.2.2.1 Rôle et Fonctionnement

Le corrélateur est l'étage du récepteur calculant la corrélation entre le code PRN du signal satellite reçu et sa réplique locale. Ses différentes sorties permettent d'approcher les paramètres de retard, de fréquence Doppler et de phase du signal reçu. Les opérations de corrélation ne s'effectuent pas directement sur le signal reçu. Celui est multiplié par un signal généré localement afin de supprimer la porteuse du signal reçu. Ceci a pour effet de générer deux composantes : une composante I pour In-phase signal et une composante Q pour in-Quadrature signal [40]. Les différentes opérations de corrélation s'effectuent sur ces deux composantes sur un intervalle de temps de longueur T_i (période de temps d'intégration). Le réglage de la valeur T_i détermine les performances du récepteur : une valeur de T_i grande accroît la sensibilité du récepteur, lui permettant de détecter des signaux de puissance faible, mais augmente également le temps de calcul de l'étape d'acquisition (étape de recherche des paramètres du signal reçu, explicité à la section 1.2.3) [26]. La réplique locale du signal correspondant au satellite s_i

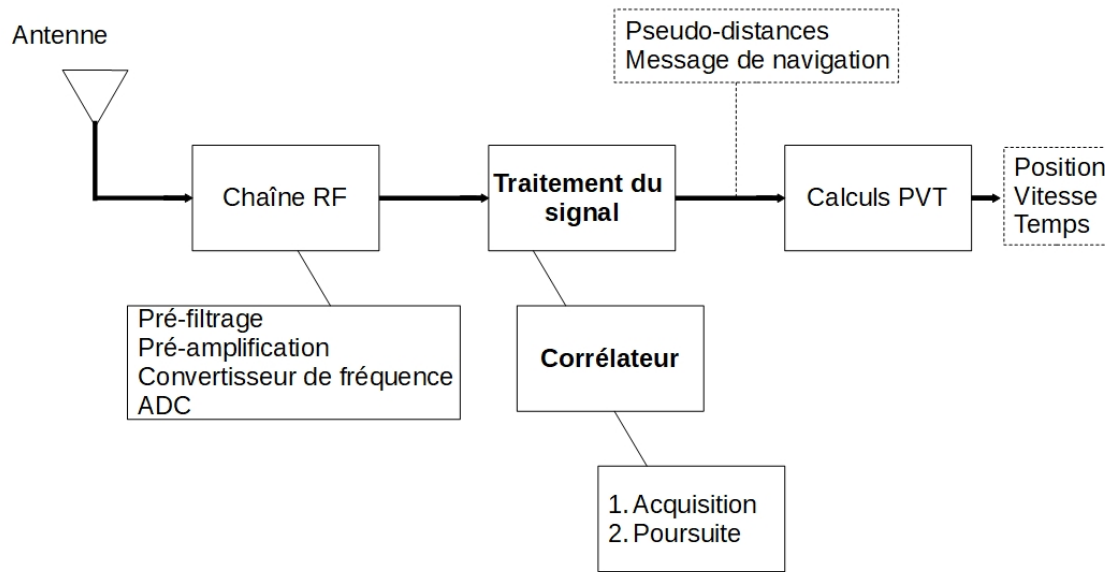


FIGURE 1.2 – Schéma simplifié d'un récepteur GPS.

avec un retard estimé $\tilde{\tau}$ générée au sein du récepteur prend la forme suivante :

$$r_{I_o}(t) = \cos(2\pi(f_0 + \tilde{f}_D)t + \tilde{\phi})c_{I_o}^i(t - \tilde{\tau})$$

avec

- f_0 une fréquence intermédiaire utilisée au sein du récepteur,
- estimation du retard du signal reçu $\tilde{\tau}$,
- estimation de la fréquence Doppler du signal reçu \tilde{f}_D ,
- estimation de la phase du signal reçu $\tilde{\phi}$,
- $c_{I_o}^i(t - \tilde{\tau})$ réplique du code PRN identique à celui du signal utile visé avec une estimation du retard du signal reçu $\tilde{\tau}$ et également corrigée de la fréquence Doppler estimée.

Remarque : Les variables $\tilde{\tau}$, \tilde{f}_D et $\tilde{\phi}$ sont également des fonctions du temps, mais considérées comme constante sur la période d'intégration T_i .

Le corrélateur est composé de plusieurs fonctions électroniques : la boucle de code ou *Delay-Locked Loop* (DLL), déterminant une estimation $\tilde{\tau}$ de la phase de code du signal reçu et la boucle de phase ou *Phase-Locked Loop* (PLL) qui détermine une estimation $\tilde{\phi}$ de la phase de la porteuse. Les sorties de la PLL permettent également de donner une estimation de la fréquence Doppler du signal reçu via le discriminateur de phase [40]. Ces deux boucles sont entretenues par les opérations de corrélation effectuées entre la réplique locale et le signal reçu.

1.2.2.2 Sorties de corrélateur

Les sorties du corrélateur affichent la corrélation entre le signal reçu et une réplique locale (de paramètres $\tilde{\tau}$, \tilde{f}_D et $\tilde{\phi}$). Généralement un récepteur alloue trois sorties de corrélateur par voie de signal. Le signal ayant une composante I (*In-Phase*) et Q (*In-Quadrature*), la DLL totalise généralement 6 sorties. Le fonctionnement de la boucle suivant le même principe sur les deux

1.2. Réception et synchronisation du signal

voies, l'explication portera uniquement sur la voie I. La DLL est alimentée par trois sorties sur la voie I détaillées ci-dessous :

- *Early* : corrélation entre le signal et la réplique locale du code de *Gold* de retard τ_E généralement égal à $\tilde{\tau} - \frac{T_c}{2}$ (ce qui est équivalent à dire que la réplique est en avance de $\frac{T_c}{2}$),
- *Prompt* : corrélation entre le signal et la réplique locale de retard nul ($\tau_P = \tilde{\tau}$),
- *Late* : corrélation entre le signal et la réplique locale de retard τ_L généralement égal à $\tilde{\tau} + \frac{T_c}{2}$.

Avec une formulation plus mathématique, les trois sorties *Early*, *Prompt* and *Late* (E-P-L) de corrélateur d'un signal reçu $s_r(t)$ à l'instant t sont :

- $Early = \int_{\tilde{\tau}}^{\tilde{\tau}+T_i} s_r(u) c_{Io}(u - \tau_E) \cos(2\pi(f_0 + \tilde{f}_D)u + \tilde{\phi}) du,$
- $Prompt = \int_{\tilde{\tau}}^{\tilde{\tau}+T_i} s_r(u) c_{Io}(u - \tau_P) \cos(2\pi(f_0 + \tilde{f}_D)u + \tilde{\phi}) du,$
- $Late = \int_{\tilde{\tau}}^{\tilde{\tau}+T_i} s_r(u) c_{Io}(u - \tau_L) \cos(2\pi(f_0 + \tilde{f}_D)u + \tilde{\phi}) du.$

Chacune des sorties est située à une distance de $\frac{T_c}{2}$ de ses plus proches voisines. La distance séparant les deux sorties les plus éloignées (soient *Early* et *Late*) est égale à la taille d'un *chip* et est appelée espace *Early-Late*, que l'on notera δ_{EL} . Lorsque le pic de corrélation n'est pas localisé au niveau de la sortie *Prompt* comme dans le cas de la figure 1.3, le décalage des répliques doit être adapté : dans le cas exposé la valeur estimée de la phase de code (point permettant d'identifier le début du code C/A, à ne pas confondre avec la phase du signal) est sous-estimée et il faut augmenter le retard estimé de ses répliques pour synchroniser la sortie *Prompt* au signal reçu. Lorsque les valeurs de corrélation sont similaires au niveau des sorties *Early* et *Late*³ (voir figure 1.4), alors la phase de code est considérée comme correctement estimée. On dit alors que la boucle est parfaitement réglée [40].

1.2.3 Synchronisation de la réplique avec le signal reçu : acquisition et poursuite du signal

Dans le processus de positionnement par satellites, un récepteur a besoin d'identifier les satellites émetteurs des signaux reçus. Suite à l'identification du code PRN correspondant, le signal est synchronisé avec une réplique locale afin de décoder le message de navigation transmis, rendant possible le calcul de la pseudo-distance. Les phase d'identification et synchronisation s'effectuent à partir du corrélateur selon deux étapes successives :

1. **L'acquisition**, permettant la détermination des signaux reçus par le récepteur. Lors de la recherche des signaux (et de leurs paramètres caractéristiques), des répliques avec différentes paires (retard, fréquence Doppler) sont générées puis corrélées avec le signal incident. Cette grille de recherche, illustrée figure 1.5, affiche la corrélation entre la réplique locale et le signal entrant. Grâce aux propriétés du code C/A, le pic de corrélation de la grille correspond à une bonne approximation des paramètres (τ , f_D) du signal. La recherche de la phase de code est effectuée en testant toutes les valeurs possibles (1-1023). Quant à la recherche en fréquence Doppler, elle s'effectue sur un intervalle de fréquence de 10 kHz dans l'objectif de couvrir toutes les fréquences, y compris celles atteintes par des avions en vitesse de croisière [26]. Afin d'effectuer la recherche dans un temps restreint, la largeur de bande est limitée. Il y a un compromis entre l'estimation précise des paramètres du signal et le temps de calcul de la grille d'acquisition. De plus, le temps d'acquisition cohérente est limité par les transitions de bits contenues dans le message de navigation [44]. En effet les bits de navigation ont une longueur de 20 ms : effectuer une intégration cohérente de plus de 20 ms

3. Lorsque le code C/A est décalé d'un delta $\pm \frac{T_c}{2}$ entre le signal reçu et la réplique, la valeur de corrélation diminue de moitié, correspondant à une perte de 6 dB en amplitude.

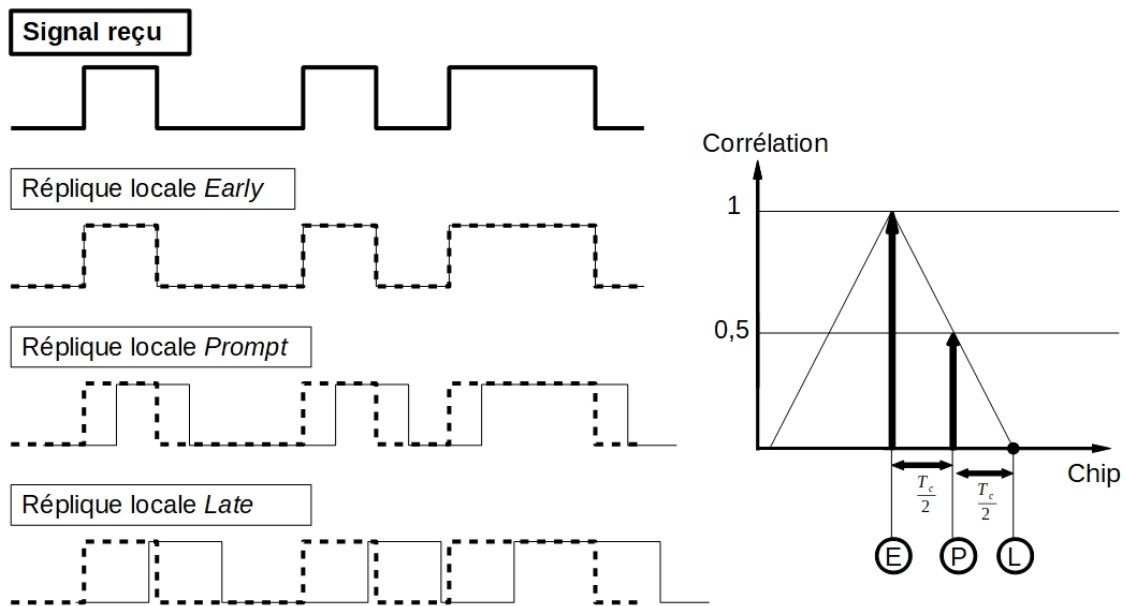


FIGURE 1.3 – Répliques locales lors d'une estimation incorrecte de la phase de code du signal. Le corrélateur compare ses répliques locales au signal incident. Dans ce cas présent, la valeur maximale de corrélation est obtenue avec la réplique *Early* : le corrélateur sous-estime la phase de code du signal.

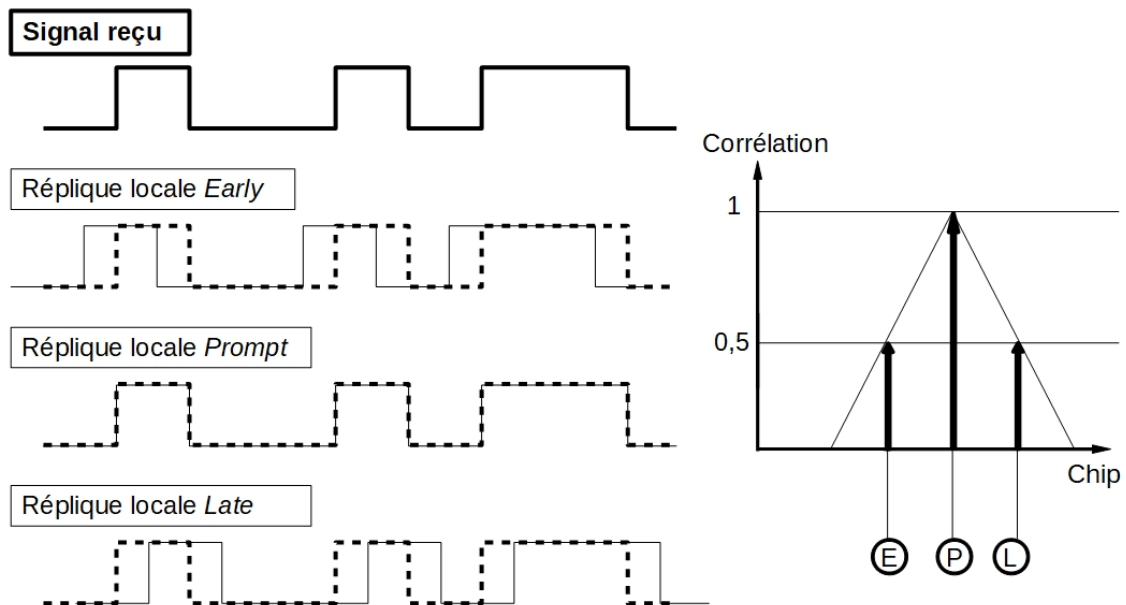


FIGURE 1.4 – Répliques locales lors d'une estimation correcte de la phase de code du signal. Le corrélateur compare ses répliques locales au signal incident. Dans ce cas présent, la valeur maximale de corrélation est obtenue avec la réplique *Prompt* et les valeurs de corrélation des sorties *Early* et *Late* sont les mêmes : le corrélateur estime correctement la phase de code du signal.

1.3. Phénomène de multitrajet

peut ainsi amener à intégrer sur des données contenant un bit de transition. Au contraire, plus le temps d'intégration est court et plus les chances d'avoir un bit de transition dans les données sont réduites, diminuant par la même occasion la sensibilité du récepteur aux signaux faibles [45],

2. La **poursuite** est l'étape succédant à l'acquisition (voir schéma 1.2). L'étape précédente a permis d'identifier le signal reçu avec une approximation des paramètres de retard et de fréquence Doppler. Lors de cette étape les sorties du corrélateur sont injectées dans des boucles de poursuite afin d'affiner l'information fournie par l'acquisition et les mettre à jour de façon régulière via la DLL pour le paramètre de retard et la PLL pour les paramètres de phase et de fréquence Doppler. Le fonctionnement des deux boucles repose sur un discriminateur calculant l'erreur d'estimation du paramètre recherché, un filtre passe-bas (F_{LP} dans la figure 1.6) réglant la réponse de la boucle et effectuant un filtrage de bruit et un oscillateur numérique⁴.

Remarques :

- Dans le cas précis où un récepteur n'arrive pas à trouver le satellite recherché, l'étape de poursuite n'est pas enclenchée et le récepteur cherche à réaliser l'acquisition d'un autre satellite,
- En cas de perte de signal, le récepteur doit recommencer les étapes d'acquisition et de poursuite dans cet ordre.

Dans le cadre de la thèse, nous considérerons la phase d'acquisition complétée et nous nous placerons toujours à l'étape de poursuite. En d'autres termes, le récepteur aura déjà effectué l'approximation des paramètres de retard, de fréquence Doppler et de phase du signal reçu.

1.3 Phénomène de multitrajet

Dans la dernière section de ce chapitre consacré au positionnement par satellite nous allons introduire le phénomène de multitrajet et définir ses caractéristiques afin de comprendre les nuisances apportées par ce type d'interférence. Nous finirons cette section par une présentation de techniques existantes visant à réduire les effets du multitrajet sur le calcul du positionnement.

1.3.1 Définition d'un multitrajet

Parmi les phénomènes d'interférences possibles concernant un signal GNSS, le multitrajet est considéré comme étant une source d'importantes dégradations [44]. Le multitrajet est la somme du signal direct (également appelé *Line-Of-Sight* (LOS)) et d'échos provoqués par des effets tels que la réflexion sur le relief ou la diffraction d'un signal sur un obstacle comme un bâtiment. Ce sont des répliques du LOS avec des paramètres différents tels que le retard et l'amplitude (illustration figure 1.7). Un phénomène de multitrajet apparaît lorsqu'il y a superposition d'un signal direct avec au moins un signal indirect au niveau de l'antenne réceptrice. La particularité de ce type d'interférence réside dans leur ressemblance avec le signal d'intérêt et leur forte dépendance vis-à-vis de l'environnement dans lequel ils se propagent : en effet l'écho dépend de l'objet réflecteur qui l'a engendré, comme le matériel le composant [47]. Différentes alternatives classiques de correction d'erreur comme les méthodes différentielles [3],

4. Génère une porteuse de fréquence instantanée $f_{NCO} = f_0 + \Delta f(V_c)$ à partir d'un signal d'entrée numérique V_c [38].

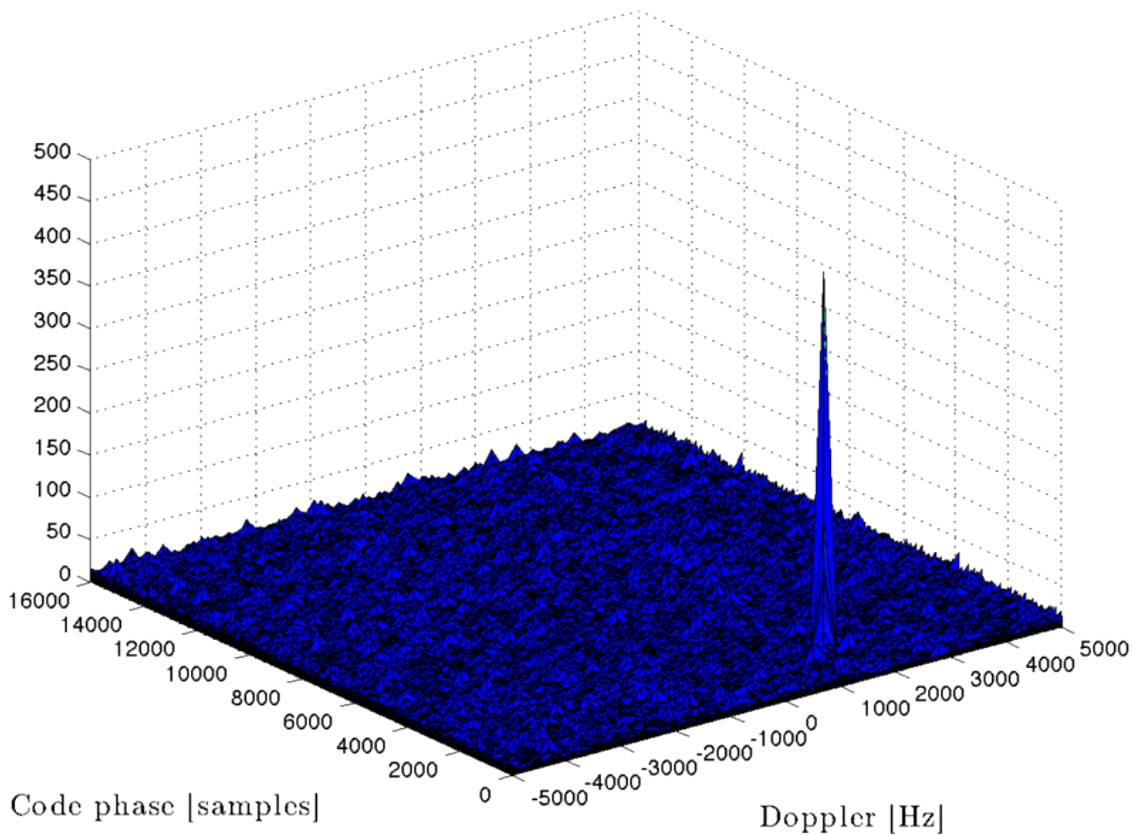


FIGURE 1.5 – Matrice d’acquisition sur le code PRN 21 [46].

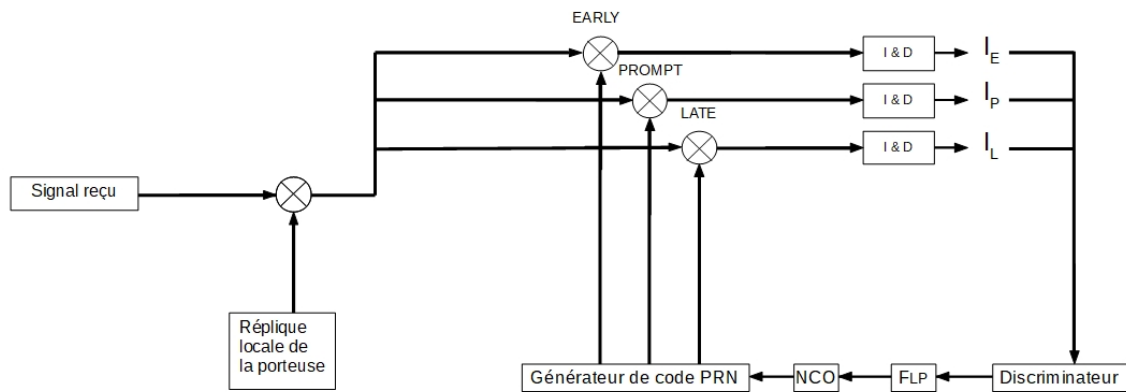


FIGURE 1.6 – Schéma d’une DLL sur la composante I d’un signal au sein d’un récepteur GPS. La sortie I_E pour I_{Early} correspond à la valeur de corrélation entre le signal reçu et la réplique locale de retard $\tilde{\tau} - \frac{T_c}{2}$ sur un intervalle d’intégration T_i . De façon similaire I_P et I_L sont respectivement les valeurs de corrélation du signal reçu multiplié par les répliques de retard $\tilde{\tau}$ et $\tilde{\tau} + \frac{T_c}{2}$. Les sorties sont calculées par les blocs d’intégration et de filtrage I & D (pour *Integrate & Dump*) [38] et alimentent la boucle de poursuite avec le discriminateur, le filtre de boucle noté F_{LP} , et un NCO avant d’être réinjectées dans le générateur de code.

1.3. Phénomène de multitrajet

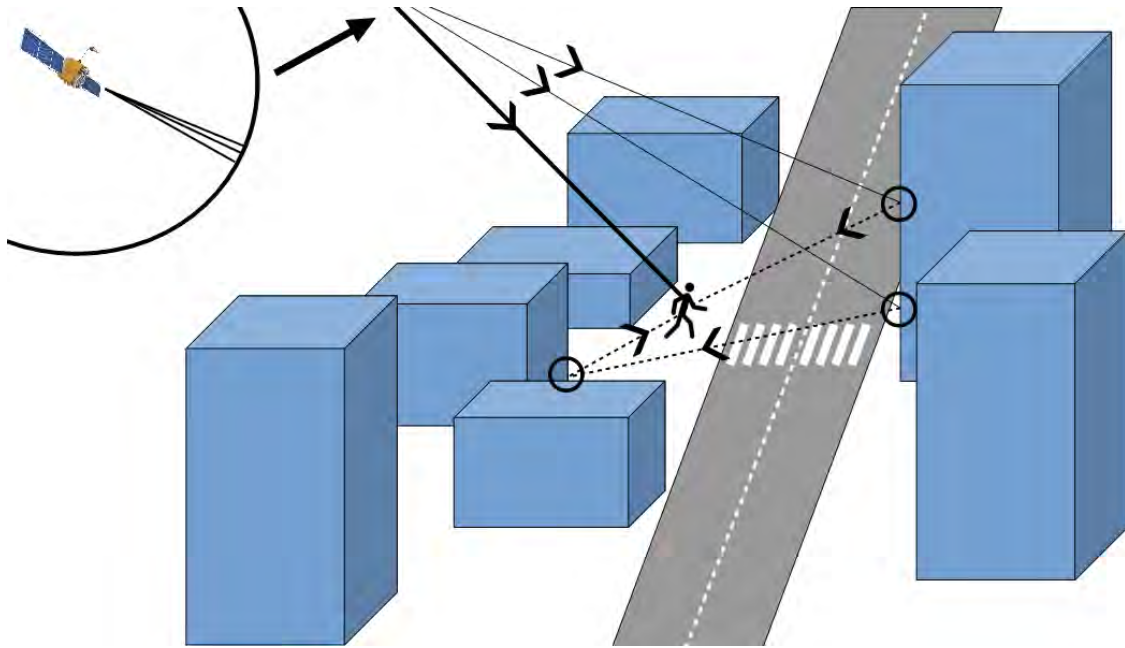


FIGURE 1.7 – Phénomène de multitrajet. Les multitrajets sont des signaux originaires du même satellite émetteur que le signal LOS (en gras). Ils subissent une ou plusieurs interactions physiques (signal en tirets après réflexion) avec l’environnement local du récepteur. Le phénomène est illustré ici avec une réception s’effectuant au niveau d’un smartphone situé dans la poche d’un piéton évoluant dans une zone urbaine composée d’immeubles. Plusieurs multitrajets apparaissent par réflexion/diffraction sur les bâtiments de la zone.

Copyrights image piéton : libre de droits issue de Vecteezy

un design particulier du récepteur [48] ou un allongement de la période d’intégration [49] ne permettent pas de complètement éliminer les répliques. La méconnaissance concernant les composantes d’un écho ainsi que du nombre d’échos présents à l’entrée du récepteur font de la réduction de ce phénomène un véritable défi, d’autant que les erreurs provoquées par ce phénomène sont prépondérantes dans l’erreur totale pour les calculs de positionnement au niveau du récepteur [47].

Remarque : Dans la suite du manuscrit, le terme multitrajet fera référence à la fois au phénomène d’interférence ainsi qu’à l’écho du signal LOS.

1.3.2 Modélisation d’un multitrajet

Par rapport au signal LOS, le multitrajet subit une atténuation, un changement de phase, un changement fréquentiel ainsi qu’un retard sur une dimension temporelle [50]. La fréquence Doppler du multitrajet peut notamment être affectée par le mouvement des objets environnants. Dans le cadre de la thèse, nous considérerons qu’un multitrajet peut être entièrement caractérisé par les quatre paramètres suivants :

- α l’atténuation par rapport au LOS,
- $\Delta\phi$ le décalage de phase par rapport au LOS,
- $\Delta\tau$ le retard temporel entre le signal LOS et le signal réfléchi,
- δf le décalage en fréquence Doppler par rapport au LOS.

Un récepteur pouvant subir plusieurs multitrajets en même temps, le signal reçu au niveau de l'antenne du récepteur est la somme d'un trajet direct et de $M \in \mathbb{N}^*$ multitrajets. Dans cette situation le signal reçu est régi par l'équation suivante :

$$s_r(t) = A_c \left[C(t)D(t) \sin(2\pi(f_{L1} + f_D)t + \phi) + \sum_{i=1}^M \alpha_i C(t - \Delta\tau_i)D(t - \Delta\tau_i) \sin(2\pi(f_{L1} + \delta f_i)(t - \tau_i) + \Delta\phi_i) \right] + b(t)$$

avec :

- α_i l'atténuation du multitrajets i ,
- $\Delta\tau_i$ le retard du multitrajets i par rapport au signal LOS,
- δf_i le décalage en fréquence Doppler du multitrajets i par rapport au signal LOS,
- $\Delta\phi_i$ la différence de phase entre le multitrajets i et le signal LOS,
- $b(t)$ le terme de bruit.

Les paramètres α_i , $\Delta\tau_i$, δf_i et $\Delta\phi_i$ dépendent du temps. Dans le cadre de nos travaux, leur variation est considérée suffisamment négligeable sur une période d'intégration T_i : ils sont ainsi considérés constants sur T_i .

Dans les cas d'études rencontrés des chapitres 4, 5 et 6, nous analyserons exclusivement des cas avec un seul multitrajets, soit $M = 1$. Le chapitre 7 abordera des cas avec $M > 1$.

Remarque : Lorsque le coefficient d'atténuation α d'un multitrajets est faible (c'est-à-dire lorsque le multitrajets est fortement atténué après réflexion), l'influence du bruit est plus importante que celle du multitrajets [50].

1.3.3 Impacts d'un multitrajets au niveau du signal en réception

Le récepteur est capable de détecter les signaux GNSS en dessous du plancher du bruit [51] ainsi que de différencier les signaux en provenance d'autres satellites. En revanche il n'est pas en mesure de distinguer des signaux en provenance d'un même satellite : ce qui est le cas lors d'un phénomène de multitrajets. Un signal LOS et un multitrajets (dit *Non Line-Of-Sight* (NLOS)) partagent le même code PRN, le même message de navigation mais avec des décalages au niveau de ses paramètres⁵. Les multitrajets provoquent des interférences tantôt constructives, tantôt destructives avec les signaux directs, résultant en une déformation des sorties de corrélateur du récepteur [47]. Comme le montre l'illustration en figure 1.8, le phénomène de multitrajets impacte directement au niveau des sorties du corrélateur en déformant le pic de corrélation, conséquence de l'influence de chacun des signaux, direct et indirect. Le paramètre de retard intervenant dans le calcul de la pseudo-distance entre le satellite et le récepteur, un multitrajets peut induire une erreur de calcul de pseudo-distance pouvant atteindre les 150 mètres pour le code C/A d'après [52].

1.3.4 Méthodes actuelles déployées contre le multitrajets

Étant l'une des menaces les plus sérieuses à la précision des mesures de signaux GNSS, le phénomène de multitrajets a fait l'objet de nombreuses recherches afin de réduire voir de supprimer son influence. Des techniques comme la multiplication de sorties de corrélateur ainsi que

5. En réalité, le récepteur n'est pas impacté par des multitrajets de retard supérieur à $1.5 T_c$. Cela est dû au triangle de corrélation des sorties du corrélateur. Un multitrajets de retard supérieur à $\tau_{MP} \geq 1.5 T_c$ ne perturbe plus aucune sortie de corrélateur car son triangle de corrélation apparaît au-delà de la sortie *Late*. C'est une caractéristique que l'on prendra en compte lors de la génération des données dans le chapitre 2.

1.3. Phénomène de multitrajet

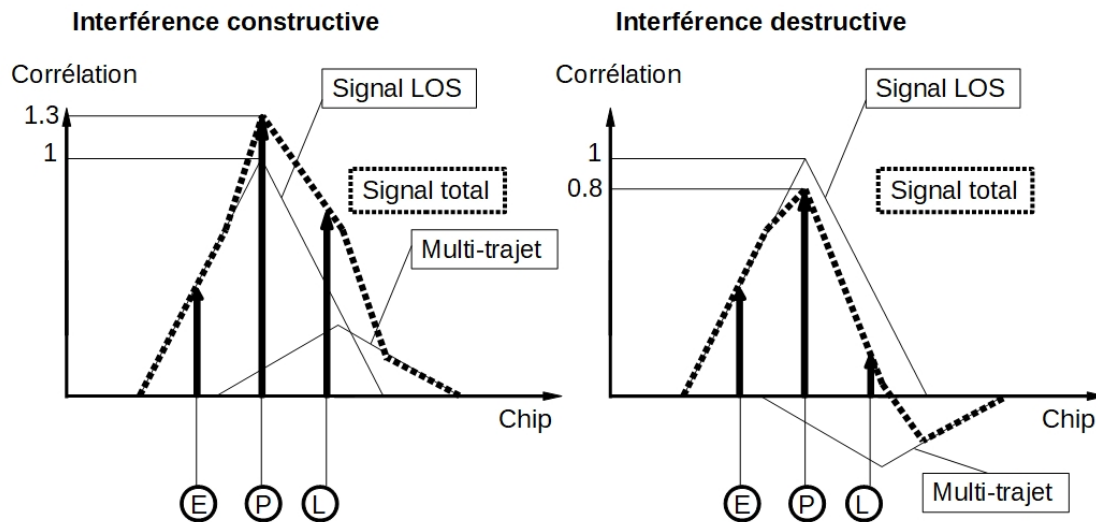


FIGURE 1.8 – Effets d'un multitrajet sur un corrélateur. Les deux cas exposés sont les effets de la présence d'un seul multitrajet au niveau des sorties du corrélateur. Les schémas illustrent respectivement le cas d'une interférence constructive (multitrajet en phase) et d'une interférence destructive (multitrajet en opposition de phase). La différence entre les sorties *Early* et *Late* fausse la boucle de calcul alimentée par les sorties, ce qui *in fine* fausse l'estimation de la phase de code.

l'interpolation de ces sorties ont démontré des résultats satisfaisants et seront l'objet de cette section. La liste présentée ci-dessous n'est pas exhaustive et exclut les techniques faisant appel à des moyens physiques comme des modifications au niveau de l'antenne [53]. Nous exposons leur principe, et lorsque l'information est disponible, le nombre de sorties de corrélateur utilisées ainsi que le nombre de multitrajets pouvant être traités.

1.3.4.1 Multipath Estimating Delay Lock Loop

La méthode du *Multipath Estimating Delay-Locked Loop* (MEDLL), présentée dans [54], consiste à effectuer une décomposition de la fonction de corrélation en séparant les composantes du LOS et des multitrajets. En estimant les paramètres d'amplitude, de retard et de phase de chaque multitrajet, la MEDLL va pouvoir donner une estimation de la fonction de corrélation du multitrajet concerné. Cette estimation sera ensuite soustraite à la fonction de corrélation mesurée. En appliquant ce principe de manière itérative, la fonction de corrélation obtenue est une estimation de la fonction de corrélation du signal LOS. Ce résultat sera ensuite injecté dans la DLL afin d'effectuer les opérations de poursuite. La MEDLL s'applique en modifiant la structure *hardware* du corrélateur : celui-ci n'est pas cantonné aux trois sorties classiques *Early*, *Prompt* et *Late* mais de N sorties avec $N > 10$. Le calcul de la fonction de corrélation s'effectue sur l'intervalle $[-T_c, T_c]$. Les N sorties sont réparties de manière uniforme dans cet intervalle. Dans le même article, un maximum de deux multitrajets était considéré, d'amplitudes $\alpha = 0.5$ avec des retards $\tau_{MP} \in [0.02 T_c, 1.1 T_c]$. Les auteurs calculent une *Root-Mean-Square* (RMS) sur les résidus (erreurs) de double-différence (méthode soustrayant les mesures de pseudodistances de deux récepteurs observant deux mêmes satellites, ce qui a

pour conséquence d'éliminer les décalages d'horloge des deux récepteurs ainsi que le décalage satellite [41]) des pseudo-distances mesurées par les récepteurs. Les performances montrent une amélioration de la RMS des résidus de double-différence de 10 à 40% pour le récepteur appliquant la méthode MEDLL. La méthode permet également de réduire les effets du multitrajét jusqu'à 90% par rapport à un récepteur standard utilisant la technique du *narrow correlator*. Cependant plusieurs désavantages sont liés à cette méthode comme le nombre de sorties de corrélateur (implémentées sur la structure *hardware*) limitant le nombre de multitrajets traités en entrée. Van Nee et al. dans [55] concluent sur les difficultés d'estimation de la MEDLL lorsque les multitrajets ont un faible retard, d'autant que la MEDLL réduit les effets des multitrajets sans totalement les supprimer.

1.3.4.2 Narrow Correlator Spacing

La technique du *Narrow Correlator Spacing* intervient au niveau de l'espace de recherche de la DLL en appliquant une variation de l'espacement δ_{EL} entre les sorties *Early* et *Late*. Traditionnellement fixé à $1 T_c$ (illustration figure 1.3) pour les récepteurs GPS modernes [4], ici des espacements de 0.05 à $1 T_c$ ont été testés. Cette variation est rendue possible en agissant sur l'horloge permettant de générer les différents codes E-P-L. Les expériences ont montré des performances supérieures lorsqu'un espacement inférieur à $1 T_c$ était appliqué à la DLL. Les performances du *narrow correlator spacing* sur code C/A en situation de multitrajét approchent celles d'un récepteur utilisant un code P (avec une largeur de bande passante fixée à 8 MHz pour un $\delta_{EL} = 0.1 T_c$). L'article conclut également sur une augmentation des performances avec une bande passante augmentée à 20 MHz pour un espacement $\delta_{EL} = 0.05 T_c$. Cependant, des inconvénients ressortent de cette technique : une étape de pré-corrélation avec une bande de fréquence⁶ définie est nécessaire, le taux d'échantillonnage doit être élevé et le traitement numérique du signal doit être plus rapide.

1.3.4.3 Early-Late slope

La technique du *Early-Late slope* (ELS) introduite dans [56] est une extension de la technique du *Narrow Correlator Spacing*. La technique repose sur l'utilisation de 2 sorties de corrélateur supplémentaires, placées en dehors de l'espace couvert par les sorties *Early* et *Late*. L'idée est de calculer les pentes du triangle de corrélation de part et d'autre de son pic et de les injecter dans un paramètre T appelé erreur de poursuite. Si les sorties de corrélation ne sont pas équidistantes, alors la valeur T ne sera pas nulle et le récepteur recentrera les sorties *Early* et *Late* autour du pic de corrélation. Des tests sur une bande passante de 8 MHz avec des multitrajets d'amplitude $\alpha = 0.5$ et de retard compris dans l'intervalle $[0, 1.5 T_c]$ ont permis de montrer une amélioration des performances par rapport à une utilisation du *narrow correlator* sans calcul de pente. L'article montre une RMS des résidus des doubles différences divisée par deux avec l'utilisation de cette extension. Elle réduit le biais introduit par le multitrajét sur la pseudo-distance calculée de 25 à 50% par rapport à la technique du *narrow correlator* seule. Les performances du récepteur avec *Early-Late slope* en présence de multitrajét approchent celles d'un récepteur avec une antenne à anneau d'arrêt.

6. Suffisamment grande afin de ne pas altérer le pic de la fonction d'autocorrélation.

1.3.4.4 Double-Delta correlator et Extended Double-Delta correlator

En présence de multitrajét, la fonction de corrélation (et donc l'estimation du retard du signal) est altérée et les pentes de part et d'autres du pic de corrélation ne sont plus symétriques. L'idée de la technique du *Double-Delta correlator* est de compenser ces différences de pentes avec un terme de correction pour l'estimation du retard [57]. La technique introduit l'utilisation d'une paire de sorties de corrélateur supplémentaire afin de calculer un terme de correction τ_{corr} définie comme étant

$$\tau_{corr} = \frac{\delta_{E_1 L_1}}{2} \frac{2(E_1 - L_1) + (E_2 - L_2)}{E_1 - E_2 + L_1 - L_2}$$

avec :

- $\delta_{E_1 L_1}$ la distance séparant les sorties de corrélateur (E_1, L_1) ,
- (E_1, L_1) et (E_2, L_2) les paires de sorties de corrélateur calculées,
- E_1 est une sortie correspondant au retard $\tilde{\tau} - \frac{T_c}{4}$,
- E_2 est une sortie correspondant au retard $\tilde{\tau} - \frac{T_c}{2}$,
- L_1 est une sortie correspondant au retard $\tilde{\tau} + \frac{T_c}{4}$,
- L_2 est une sortie correspondant au retard $\tilde{\tau} + \frac{T_c}{2}$.

La méthode montre des limites lors de la gestion de plusieurs multitrajets ainsi que de multitrajets d'intensités plus importantes que celles du LOS⁷. L'article [57] montre que dans le cas d'un multitrajét plus fort que le LOS, les estimations du *Time Of Arrival* (TOA), c'est-à-dire le temps absolu de propagation d'un signal du satellite au récepteur, se dégradent fortement. Ceci est principalement dû au choix erroné du pic de corrélation comme étant celui correspondant au LOS. L'*Extended Double-Delta correlator* [58] introduit des sorties de corrélateur supplémentaires, permettant le calcul d'autres termes de compensation. Il donne également la possibilité de modification de l'espacement entre les différentes paires de sorties de corrélateur. Dans l'article [58], la technique *Extended Double-Delta correlator* a démontré de meilleures performances d'estimation de l'enveloppe d'erreur du multitrajét comparée aux techniques de *Early-Minus-Late*, *Narrow correlator* et *Double-Delta correlator* (expériences uniquement sur des cas avec un seul multitrajét). En contrepartie, l'erreur de poursuite est plus élevée avec l'*Extended Double-Delta correlator* comparée au *Double-Delta correlator*.

1.3.4.5 Multipath Insensitive Delay Lock Loop

Présentée dans l'article [5], la MIDLL est une technique s'appuyant sur la recherche d'un point invariant pour le discriminateur au sein du corrélateur. Plutôt que de chercher à synchroniser la sortie *Prompt* avec le signal direct, la synchronisation va se réaliser au niveau de la sortie *Very Late* (VL), correspondant à une corrélation entre le signal et une réplique de retard $\tau_{VL} = T_c + \frac{\delta_{EL}}{2}$. Les trois sorties de corrélateur deviennent *Early*, *Very Late* et *Late* (E-VL-L). Cette propriété d'invariance en sortie *Very Late* pour le discriminateur est intéressante car conservée quel que soit le multitrajét en entrée. Testée pour différents espaces *Early-Late*, la MIDLL montre une meilleure stabilité dans les performances en poursuite lorsque $\delta_{EL} < 1 T_c$. La synchronisation avec la réplique VL permet de limiter à 1 mètre l'erreur de poursuite due à un multitrajét. L'article compare la MIDLL avec la méthode du *Narrow Correlator*. Pour une bande passante en pré-corrélation de 2 MHz, la MIDLL approche les performances d'erreur en poursuite d'un *Narrow Correlator* pour une bande passante de 8 MHz : sur bande passante

7. Phénomène pouvant se produire lorsque la combinaison des multitrajets dépasse l'amplitude du signal LOS, ou que le signal LOS a subi une altération suite à la traversée d'un feuillage

équivalente, la MIDLL montre une supériorité dans les performances. Après implémentation sur un logiciel de réception GPS, la MIDLL a démontré une enveloppe d'erreur proche de celle atteinte par un *Narrow Correlator*. D'une simplicité d'implémentation soulignée comme plus aisée comparée aux techniques de MEDLL ou de *Narrow Correlator*, la MIDLL est cependant plus sensible au bruit comparée aux méthodes précédemment citées. La MIDLL entraînant également une perte de puissance du signal, son utilisation est recommandée sur des signaux puissants.

1.3.4.6 Multipath mitigation in the frequency domain

Le principe évoqué dans l'article [59] parle d'une réduction de l'impact du multitrajet en partant du postulat que toute source d'erreur des mesures GPS comme celles apportées par la traversée des différentes couches atmosphériques, le bruit récepteur et le multitrajet inclus possèdent un spectre fréquentiel qui leur est propre. Le processus présenté est une séquence se déroulant en trois étapes : estimation du multitrajet, atténuation et correction. Dans la première étape, le récepteur cherche à estimer le retard du multitrajet en synchronisant le multitrajet avec une modélisation locale de celui-ci. Dans la deuxième phase, le retard estimé est transféré dans l'espace fréquentiel et intégré dans un calcul de résidus *Code-minus-Carrier* (CmC). La mesure de CmC est construite comme la différence entre les mesures de pseudodistance ρ et de phase ϕ [60] :

$$CmC = \rho - \phi$$

Le CmC est une métrique qui est utilisée afin d'extraire les retards des multitrajets Elle a par exemple été appliquée sur des calculs en multibandes fréquentielles afin de quantifier les erreurs de pseudo-distances provoquées par les multitrajets [61]. La troisième étape applique une correction de la pseudo-distance mesurée suite aux calculs de résidus à l'étape 2. Les expériences ont été conduites sur des données réelles contenant éphémérides et observables. Les résidus de CmC permettent d'exposer la présence de multitrajet dans les mesures. Ils sont utilisés dans un calcul de ratio de compression, mesurant l'amélioration de l'écart-type du CmC du multitrajet atténué. Les expériences ont montré une amélioration de l'ordre de 50% (et atteignant dans certains cas 70%) du ratio de compression entre un CmC avec le multitrajet atténué et un CmC sans aucune atténuation. L'article étudie des cas de signaux en présence d'un seul multitrajet et ayant subi une seule réflexion. La taille du bloc *Fast Fourier Transform* (FFT) utilisé requiert l'utilisation d'au moins 256 points pour atténuer le multitrajet de manière efficace. La taille de la fenêtre fréquentielle utilisée introduit un compromis entre l'efficacité d'atténuation et le chevauchement des composantes fréquentielles des différents facteurs d'erreur.

1.3.4.7 Fast Iterative Maximum-Likelihood Algorithm

L'algorithme itératif FIMLA [62] est une adaptation de l'estimation du maximum de vraisemblance par méthode Newtonienne appliquée au problème de multitrajet. Le problème d'estimation du paramètre de retard τ_{MP} peut être assimilé à un problème de maximum de vraisemblance dans le domaine temporel. La fonction de vraisemblance utilisée est construite à partir des sorties de corrélateur des voies I et Q à deux instants différents t_1 et t_2 . Dans les différentes expériences effectuées, le temps d'intégration était fixé à $T_i = 100ms$ pour un *Signal-to-Noise Ratio* (SNR)=-20dB avec un ratio entre le signal LOS et le multitrajet égal à 5 dB. Différentes valeurs d'espace *Early-Late* ont été testées, les valeurs $\delta_{EL} < 0.25 T_c$ ont permis d'obtenir de bonnes performances d'estimation. La diminution de ce paramètre permet de réduire le biais du multitrajet mais au détriment de la variance d'estimation. Les différentes simulations ont

montré que l'algorithme FIMLA offrait de meilleurs résultats que les DLL « classiques », y compris avec la technique du *Narrow Correlator*. À titre d'exemple, le FIMLA a des performances en environnement de multitrajét similaires aux DLL conventionnelles avec réception d'un signal direct seul. L'algorithme réduit l'effet du multitrajét au niveau de l'estimation du retard de code et propose de meilleures performances que les DLL conventionnelles dans toutes les situations. En revanche l'algorithme montre des difficultés lorsque le retard du signal LOS et du multitrajét sont proches, traduisant une difficulté à effectuer la séparation. De plus l'article souligne le problème des futurs signaux GNSS avec des fonctions de corrélation multi-pics et propose de faire appel aux méthodes de *Double-Delta correlator* comme solution au problème.

1.3.4.8 Multipath estimation in multicorrelator GNSS receivers using the maximum likelihood principle

La technique présentée dans l'article [63] s'appuie sur l'utilisation d'une fonction de maximum de vraisemblance pour l'estimation du couple de paramètres (τ, Θ) ⁸ d'un signal LOS ou NLOS. La recherche s'effectue sur une grille explorant les retards possibles des signaux reçus, dont le pas de grille est déterminé par $\delta = \frac{T_c}{N}$, avec $2N + 1$ le nombre de sorties de corrélateur du récepteur. Dans l'article, N varie de 10 à 30. Un processus d'interpolation entre les sorties de corrélateur est introduit dans l'article afin d'avoir une meilleure approximation de la corrélation entre les sorties. Lors des expériences, l'amplitude du multitrajét est fixée à $\frac{A_c}{2}$, avec A_c l'amplitude du trajet principal, le retard n'excède jamais $1.2 T_c$ et la phase est uniformément répartie entre $[-\pi, \pi]$. L'algorithme a montré de bonnes performances en situation de plusieurs multitrajets. Cependant la *Root-Mean-Square Error* (RMSE) augmente de manière significative avec une utilisation de moins de 41 sorties de corrélateur. L'utilisation de méthodes d'interpolation permet cependant de réduire ce phénomène. L'algorithme montre également des limites en cas de sous-estimation du nombre de multitrajets, affectant l'estimateur par l'ajout d'un biais.

1.3.4.9 GNSS multipath estimation and mitigation based on particle filter

Les filtres particuliers sont un modèle d'estimation paramétrique reposant sur l'utilisation d'informations de variables d'observation. On possède toutes les observations d'un système pour $k \in \mathbb{N}^*$ instants avec y_i l'observation du système à un instant $i \in \llbracket 1, k \rrbracket$. Les observations $(y_i)_{1 \leq i \leq k}$ sont issues d'un système dont l'état est inconnu sur tous les instants précédents : soit x_i l'état du système à un instant $i \in \llbracket 1, k \rrbracket$, x_i est inconnu $\forall i \in \llbracket 0, k \rrbracket$ (x_0 étant l'état initial du système). L'idée des filtres particuliers est d'établir une estimation de l'état initial du système (donc x_0), de la dynamique de celui-ci (c'est-à-dire la transition entre un état x_{i-1} et un état x_i) ainsi qu'une relation entre l'état x_i du système et l'observation y_i : l'objectif principal étant d'être capable de prédire l'état x_i du système pour $\forall i \geq k$ [64]. Cette méthode est appliquée dans le cadre de l'article [6] afin d'estimer le triplet de paramètres (τ, α, ϕ) caractérisant les multitrajets. À un instant k le système implémenté est composé des variables d'état $(\tau_k, \alpha_k, \phi_k)$ auxquelles s'ajoutent le vecteur d'observation z_k composé des sorties du corrélateur des canaux I et Q à l'état k , au nombre de 42 (21 par canal). Plusieurs cas de figures ont été testés parmi lesquels la présence d'un seul multitrajét (d'amplitude 0.8 et de retard 0.3 T_c) et de deux multitrajets en simultanés (respectivement d'amplitude 0.5 et 0.3 et de retard 0.3 T_c et 0.7 T_c). Les filtres particuliers ont montré une supériorité au niveau de l'erreur moyenne d'estimation du retard du multitrajét obtenue par rapport aux techniques MEDLL et Teager *and*

8. Dans l'article [63], Θ est un paramètre complexe combinant l'amplitude et la phase d'un signal.

Kaiser *Multipath Estimating Delay-Locked Loop* (TK-MEDLL)⁹. Le filtre particulière développé montre une supériorité en performances par rapport aux deux autres méthodes en présence de plusieurs multitrajets en simultané. Cependant dans [6] Qi et al. mettent en garde sur le problème de complexité calculatoire lié aux filtres mais également aux informations à priori (état des variables observables), conseillant son utilisation dans un cadre de positionnement de grande précision mais à éviter pour une utilisation en temps réel.

1.4 Conclusion

Le multitrajet est un facteur prépondérant dans l'erreur de positionnement par satellites. Étant la réplique biaisée du signal utile attendu, les déformations des sorties de corrélateur qu'il provoque peuvent fortement dégrader les performances du corrélateur et *in fine* altérer le calcul de pseudo-distance, primordial dans la détermination de la position vraie de l'utilisateur. Les diverses techniques exposées dans la dernière section se focalisent principalement sur une action post-corrélation, avec une préférence pour l'ajout de sorties de corrélateur supplémentaires par rapport aux classiques sorties E-P-L [54, 62, 63, 6]. Dans le cadre de la thèse, nous présenterons une méthode s'appliquant également en post-corrélation. Nous multiplierons le nombre de sorties calculées à l'étage du corrélateur qui seront mises sous la forme d'images, dont le calcul et la mise en forme font l'objet du chapitre suivant.

La majorité des articles de la section 1.3.4 montre une génération des multitrajets assez restrictive. Certaines expériences comme celles menées dans les articles [6, 63] considèrent uniquement des multitrajets de fortes amplitudes (au minimum $50\%A_c$ avec A_c l'amplitude du signal LOS), ce qui est problématique lorsque Van Nee et al. dans [55] ont conclu sur une difficulté plus prononcée lors de l'estimation de multitrajets de faibles amplitudes. De même les modélisations des différents multitrajets ne semblent pas prendre en compte le décalage en fréquence Doppler pourtant réel des multitrajets [21]. Nos différents systèmes de génération de multitrajets prendront en compte une variation des paramètres de retard, de fréquence Doppler, d'amplitude et de phase plus représentative, qu'ils soient aléatoirement générés selon des lois de distribution (détaillées chapitre 2) ou par le biais d'un simulateur de propagation de rayons déterministes en environnement 3D dans le chapitre 6.

9. L'application de l'opérateur Teager-Kaiser [65] à la MEDLL [66]. Cet opérateur, qui estime l'énergie d'un signal, permet de détecter les modulations d'amplitude ou de fréquence des signaux.

1.4. Conclusion

Chapitre 2

Génération de données synthétiques sous forme d'images

Sommaire

2.1 Présentation du logiciel	27
2.2 Modélisation du signal dans le générateur	29
2.3 Génération du multitrajet	30
2.4 Ajout du bruit et mise à l'échelle des images au C/N_0	32
2.5 Génération de sorties de corrélateur	33
2.6 Construction des différentes images des canaux I & Q	36
2.7 Ensemble de données des expériences	39
2.7.1 Dataset régulé par l'amplitude du multitrajet	39
2.7.2 Dataset régulé par le niveau de C/N_0	40
2.8 Conclusion	40

Ce chapitre présente la première chaîne de génération de données de multitrajets adoptée pendant la thèse, dont le processus est illustré figure 2.1. L'objectif principal de cette première approche est la génération de données à grande échelle contenant un signal LOS en poursuite, un multitrajet aléatoirement généré ainsi qu'une composante de bruit caractéristique des sorties de corrélateur d'un récepteur GPS. Ces données prendront la forme de couples d'images auxquels seront appliqués les algorithmes d'estimation de paramètres présentés au chapitre 4. Pour comprendre comment ces sorties de corrélateur du signal interféré sont recrées, nous allons dans un premier temps définir les modèles de construction des signaux utiles (LOS) et des multitrajets (NLOS). Nous détaillerons ensuite l'élaboration des images I & Q avec un focus particulier sur le calcul des sorties de corrélateur supplémentaires. Le chapitre se termine par une présentation des ensembles de données qui seront utilisés lors des expériences des chapitres 4, 5 et 6.

2.1 Présentation du logiciel

Actuellement, il n'existe pas d'ensemble de données *labellisées* en grand nombre (disponible de manière publique) portant sur les multitrajets. Le générateur de sorties de corrélateur a été conçu afin de pallier ce manque de données. Développé par l'équipe Telecom au sein du laboratoire de l'École Nationale de l'Aviation Civile (ENAC), son objectif est la génération d'une

2.1. Présentation du logiciel

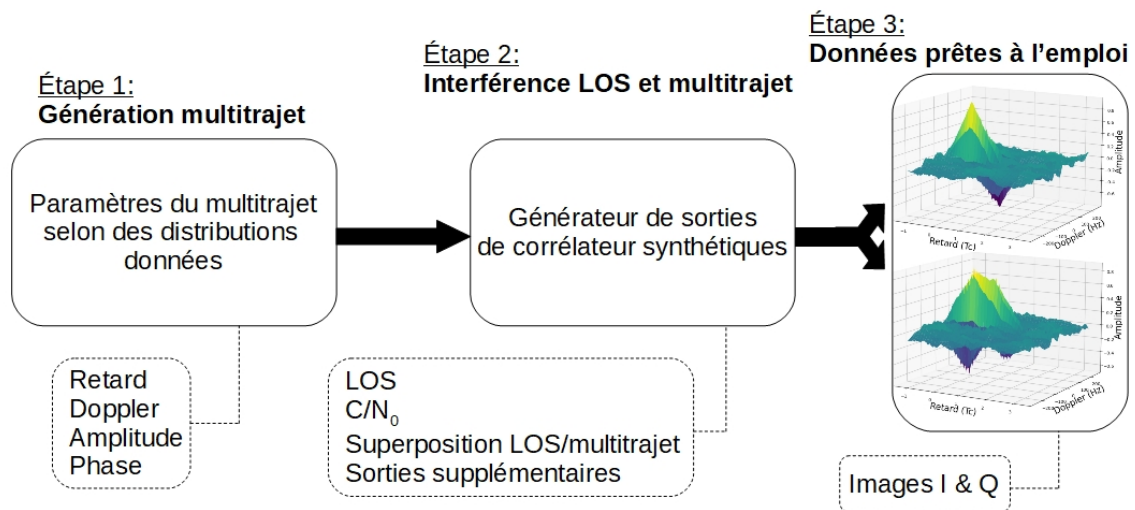


FIGURE 2.1 – La première chaîne de génération avec des données contenant des multitrajets est un processus se déroulant en trois étapes. La première étape consiste à générer le multitrayet, construit à partir du signal LOS dont les paramètres (τ , f_D , ϕ) ont été perturbés (ainsi que son amplitude). Les sorties de corrélateur sont ensuite calculées séparément pour le LOS et le multitrayet. Enfin la dernière étape consiste à construire les images des composantes I et Q du signal total en utilisant la linéarité de la chaîne de réception par rapport aux composantes du signal, avec l'ajout d'un bruit thermique typique d'un récepteur.

grande quantité de données (*labellisées*) de signaux avec des multitrajets riches en diversité au niveau des paramètres le caractérisant (retard, fréquence Doppler, amplitude, phase) mais également du bruit présent aux sorties du corrélateur. Le générateur offre ainsi une alternative au manque de données de multitrajets réels par la génération de données de signaux et multitrajets synthétiques. Cette contrainte de génération de signaux synthétiques nous conduira à améliorer la chaîne de génération de multitrajets par l'ajout d'un simulateur apportant des caractérisations plus réalistes aux données (notamment aux paramètres du multitrajet) et étudié chapitre 6. Le générateur permet en outre le calcul de sorties de corrélateur supplémentaires dans l'optique d'enrichir les informations présentées à l'algorithme d'estimation.

2.2 Modélisation du signal dans le générateur

Les signaux générés au sein du générateur sont entièrement recréés à partir d'équations les modélisant. Ils ne correspondent donc pas à la numérisation d'un signal réel reçu à l'entrée de l'antenne du récepteur, contrairement au bruit issu de mesures réelles (les raisons de ce choix seront explicitées dans la section 2.4). Le signal LOS est le signal utile qu'un récepteur cherche à extraire du bruit et des autres signaux par le processus de corrélation. C'est ce signal qui permet le calcul de pseudo-distances entre le récepteur et le satellite. Dans le générateur de sorties de corrélateur, le signal arrivant au niveau de l'antenne (voir figure 2.2) est modélisé selon l'équation suivante :

$$r(t) = \sqrt{2CD}(t - \tau)c(t - \tau) \cos(2\pi(f_c + f_D)t + \phi) + b_0(t)$$

avec :

- C la puissance du signal reçu,
- $D(t)$ le bit de navigation à l'instant t ,
- $c(t)$ le bit de la séquence PRN à l'instant t ,
- f_c la fréquence de la porteuse,
- f_D le décalage en fréquence Doppler du signal,
- ϕ la phase du signal,
- $b_0(t)$ le bruit additif blanc gaussien (*Additive White Gaussian Noise* (AWGN) en anglais).

Remarque : Le modèle du signal dans le générateur est simplifié. Le retard, la fréquence Doppler et la phase sont considérés comme constants sur la période d'intégration T_i .

Suite aux mélangeurs présents dans la chaîne de réception (figure 2.2) qui respectivement multiplient le signal utile avec une réplique locale de la porteuse puis avec une réplique locale du code PRN, les composantes I & Q du signal aux sorties du corrélateur à la suite des intégrateurs sont :

$$I = \frac{DT_i}{2} \sqrt{\frac{C}{2}} K(\tau - \tilde{\tau}) \cos(\pi(f_D - \tilde{f}_D)T_i + (\phi - \tilde{\phi})) \text{sinc}(\pi(f_D - \tilde{f}_D)T_i) + b_I \quad (2.1)$$

$$Q = \frac{DT_i}{2} \sqrt{\frac{C}{2}} K(\tau - \tilde{\tau}) \sin(\pi(f_D - \tilde{f}_D)T_i + (\phi - \tilde{\phi})) \text{sinc}(\pi(f_D - \tilde{f}_D)T_i) + b_Q \quad (2.2)$$

avec

- la fonction sinus cardinal $\text{sinc}(x) = \begin{cases} 1 & \text{si } x = 0 \\ \frac{\sin(x)}{x} & \text{si } x \neq 0 \end{cases}$
- D le bit de navigation supposé constant pendant une durée d'intégration cohérente T_i ¹,

1. T_i est à la fois un multiple de la période de code PRN et un diviseur entier de la période de bit de navigation.

2.3. Génération du multitrajét

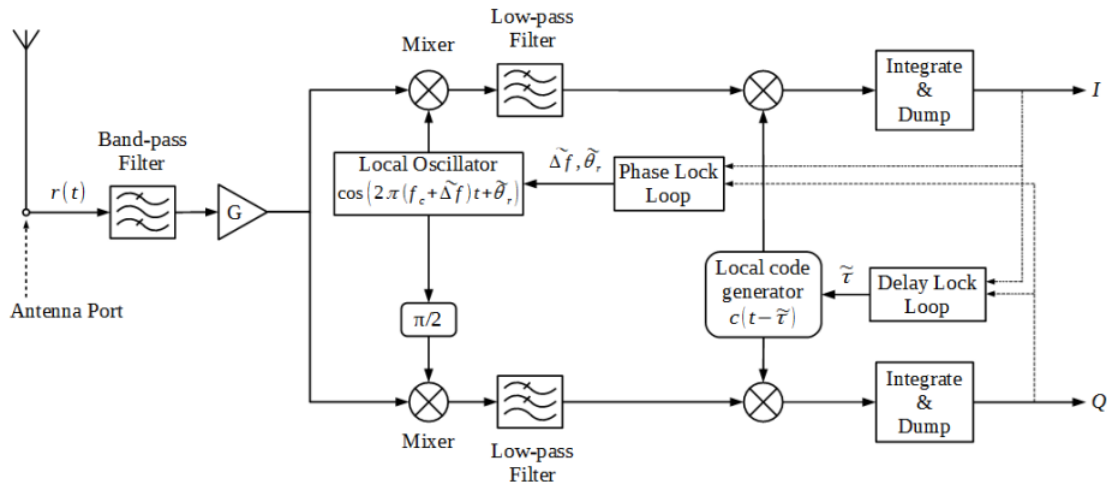


FIGURE 2.2 – Schématisation de la chaîne de réception GPS. La corrélation entre le signal et la réplique est le résultat des multiplications du signal avec les répliques de code et de porteuse plus l'intégration (Bloc *Integrate & Dump*) sur une période T_i . La PLL s'appuie sur les sorties de corrélateur pour la synchronisation des paramètres de phase et de fréquence Doppler. La DLL exerce la synchronisation de la réplique locale du code PRN pour l'estimation du retard du signal.

- $K(\tau - \tilde{\tau})$ la fonction d'auto-corrélation du code à l'instant $\tau - \tilde{\tau}$,
- $\tilde{\tau}$ l'estimation du retard du signal par le récepteur,
- \tilde{f}_D l'estimation du décalage en fréquence Doppler du signal par le récepteur,
- $\tilde{\phi}$ l'estimation de la phase du signal par le récepteur,
- b_I et b_Q les composantes de bruit sur I & Q respectivement.

Dans le cadre de la thèse, le signal est considéré en poursuite (section 1.2.2), ce qui signifie que l'étape d'acquisition a déjà été effectuée. En d'autres termes, les paramètres estimés par le récepteur sont synchronisés avec les vrais paramètres du signal : $\tilde{\tau} \approx \tau$, $\tilde{f}_D \approx f_D$ et $\tilde{\phi} \approx \phi$.

2.3 Génération du multitrajét

Le multitrajét est une réplique du signal direct avec une altération au niveau de ses paramètres. Le retard lui correspondant est plus élevé que celui du LOS puisque par définition le multitrajét parcourt une distance supplémentaire (voir section 1.3). L'amplitude du multitrajét est inférieure à celle du signal LOS². La fréquence Doppler est influencée par plusieurs facteurs comme la vitesse de l'objet réfléchissant ou la vitesse du récepteur, aboutissant à une variation de sa valeur par rapport au signal direct (c'est pourquoi les expériences du chapitre 6 prendront en compte des jeux de données contenant un récepteur mobile ainsi que des mobiles environnants de cinétiques différentes). Le multitrajét n'arrivant pas au même moment que le signal LOS au niveau de l'antenne réceptrice, la phase du multitrajét n'a pas de raison évidente d'être égale à celle du signal direct. La construction du multitrajét au sein du générateur

2. Dans la réalité, il peut exister des cas où l'amplitude du multitrajét est plus importante que celle du signal direct, par exemple si l'on se place dans un cas de réception de NLOS uniquement. Dans le cadre de la thèse, les données contiennent toujours les composantes du signal direct et du multitrajét, ce qui exclut les cas de réception uniquement composés de NLOS.

est simplement la reconstitution d'un LOS en poursuite avec des perturbations au niveau des quatre paramètres (τ, f_D, A_c, ϕ) . En posant τ_{MP} le retard du multitrajet, δf son décalage en fréquence Doppler, α_{MP} son atténuation par rapport au signal direct et ϕ_{MP} sa phase, on peut définir les disparités entre les paramètres des signaux direct et indirect :

- $\Delta\tau_{MP} = \tau_{MP} - \tau$ est le retard du multitrajet par rapport au LOS. Il sera généré selon une loi uniforme $\mathcal{U}([0, 1.5 T_c])$. Le multitrajet parcourt plus de distance que le signal LOS, impliquant $\Delta\tau_{MP} \geq 0$. Le triangle de corrélation d'un multitrajet mesurant une largeur de $2 T_c$, un multitrajet de retard supérieur à $1.5 T_c$ n'impactera pas les sorties de corrélateur impliquées dans la poursuite du signal (cas illustré dans le schéma 2.3 : le début de la composante du multitrajet sur le corrélateur est situé après la dernière sortie de corrélateur, *Late*). La génération des retards des multitrajets est donc limité à $1.5 T_c$,
- $\Delta f_D = \delta f - f_D$ est le décalage entre la fréquence Doppler du multitrajet et celle du signal LOS. Ce paramètre sera engendré selon une loi normale $\mathcal{N}(0, \frac{125}{3})$. Au-delà d'un décalage en fréquence Doppler de $\frac{2.5}{T_i}$, la perturbation apportée par le multitrajet aux sorties E-P-L est inférieure à $\frac{1}{10}$ ème de la valeur du sinus cardinal du signal LOS. Le temps d'intégration T_i étant fixé à 20 ms, un multitrajet de décalage en fréquence Doppler en dehors de l'intervalle $[-\frac{2.5}{T_i}, \frac{2.5}{T_i}] = [-125 \text{ Hz}, 125 \text{ Hz}]$ est considéré comme négligeable. La génération des décalages en fréquence Doppler suit donc une loi de Gauss dont l'écart-type a été fixé à $\sigma = \frac{125}{3}$ afin de garantir que 99,7% des échantillons respectent $|\Delta f_D| \leq 125 \text{ Hz}$,
- $\Delta\alpha_{MP} = \frac{\alpha_{MP}}{\alpha_{LOS}} = \alpha_{MP}$ car par définition on construit le coefficient d'atténuation sur un ratio par rapport à l'amplitude du signal direct ($\alpha_{LOS} = 1$). Il suivra une génération selon une loi uniforme $\mathcal{U}([0.1, 0.9])$. Le multitrajet a une amplitude plus faible que le LOS en raison de la perte de puissance subie lors de la réflexion du signal [56]. Les cas NLOS, c'est-à-dire uniquement composés de trajets indirects, étant exclus, $\alpha_{MP} \leq 1$. Dans le cadre de nos recherches, une limite égale à 0.9 a été appliquée à la génération des amplitudes de multitrajets afin d'assurer des multitrajets d'amplitudes inférieures à celles des signaux LOS. L'amplitude minimale des multitrajets générés aléatoirement par le générateur a été fixé à 10% de l'amplitude du signal LOS,
- $\Delta\phi_{MP} = \phi_{MP} - \phi$ la différence entre la phase du multitrajet et celle du signal direct, aléatoirement générée selon une loi uniforme $\mathcal{U}([0, 2\pi])$.

Par hypothèse, nous avons considéré la génération des variables aléatoires $\Delta\tau_{MP}$, Δf_D , $\Delta\alpha_{MP}$ et $\Delta\phi_{MP}$ s'effectuant de manière indépendante, c'est-à-dire sans aucun lien de causalité entre les paramètres. On observera par la suite que, pour des multitrajets plus réalistes car générés à partir d'équations de l'optique géométrique (chapitre 6), des dépendances apparaissent entre certains paramètres. Finalement dans le générateur, le multitrajet se retrouve modélisé par l'équation suivante :

$$mp(t) = \alpha_{MP} \sqrt{2C} D(t - \tau_{MP}) c(t - \tau_{MP}) \cos(2\pi(f_c + \delta f)t + \phi_{MP})$$

donnant ses composantes I & Q en sortie de corrélateur :

$$I_{MP} = \alpha_{MP} \frac{DT_i}{2} \sqrt{\frac{C}{2}} K(\Delta\tau_{MP}) \cos(\pi\Delta f_D T_i + \Delta\phi_{MP}) \text{sinc}(\pi\Delta f_D T_i) + b_I \quad (2.3)$$

$$Q_{MP} = \alpha_{MP} \frac{DT_i}{2} \sqrt{\frac{C}{2}} K(\Delta\tau_{MP}) \sin(\pi\Delta f_D T_i + \Delta\phi_{MP}) \text{sinc}(\pi\Delta f_D T_i) + b_Q \quad (2.4)$$

Remarque : Les paramètres décrivant le modèle du multitrajet sont son retard τ_{MP} , son décalage en fréquence δf , son amplitude α_{MP} et sa phase ϕ_{MP} . Le signal LOS étant en situation de

2.4. Ajout du bruit et mise à l'échelle des images au C/N_0

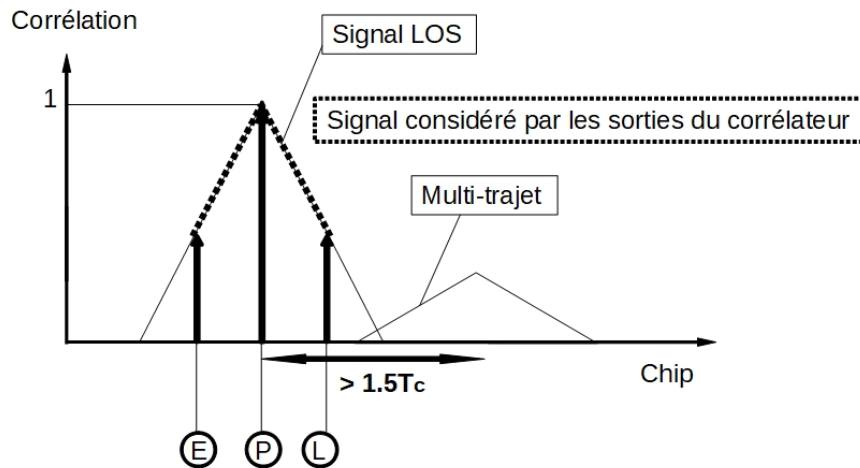


FIGURE 2.3 – Les sorties E-P-L ne sont pas impactées dans le cas d'un multitrajet de retard supérieur à $1.5 T_c$. Dans le cas d'une poursuite, le pic de corrélation du signal est localisé au niveau de la sortie *Prompt*. La corrélation aux sorties *Early* et *Late*, toutes deux à une distance de $\frac{T_c}{2}$ est de valeur 0.5. La sortie *Late* n'est pas impactée par le triangle de corrélation du multitrajet. La construction des sorties de corrélateur implique qu'un multitrajet de tel retard n'a pas d'incidence sur les performances de poursuite du signal.

poursuite, l'estimation des paramètres $\Delta\tau_{MP}$, Δf_D , $\Delta\alpha_{MP}$ et $\Delta\phi_{MP}$ revient à effectuer l'estimation de τ_{MP} , δf , α_{MP} et ϕ_{MP} . Dans la suite, on fera référence à l'estimation des paramètres du multitrajet comme étant l'estimation du quadruplet $(\Delta\tau_{MP}, \Delta f_D, \Delta\alpha_{MP}, \Delta\phi_{MP})$.

2.4 Ajout du bruit et mise à l'échelle des images au C/N_0

La modélisation analytique du bruit présent aux sorties du corrélateur présente plusieurs difficultés parmi lesquelles la corrélation croisée entre les composantes I & Q. Pour palier à ce problème, comme dans le processus de génération des images de corrélateur dans [21] et [18], nous utiliserons une banque de données contenant des bruits réels capturés en sortie de corrélateur d'un récepteur. Les échantillons de bruit ont été récoltés en suivant un processus séquentiel en deux étapes :

1. Une poursuite initiale d'un signal GNSS par un récepteur IFEN SX3 connecté à une antenne réelle. Cette expérience a été conduite sur le toit d'un bâtiment du site ENAC par l'équipe de recherche *SIGNAL processing and NAVigation* (SigNav) et a permis l'estimation et l'enregistrement des paramètres τ , f_D et ϕ du signal poursuivi,
2. Une fois la poursuite terminée, le récepteur est déconnecté de l'antenne qui est remplacée par une charge d'impédance 50Ω . On rejoue la poursuite avec les paramètres τ , f_D et ϕ enregistrés (soient $\tilde{\tau}$, \tilde{f}_D et $\tilde{\phi}$). Cette charge faisant office de « bouchon » empêche de recevoir un signal quelconque afin d'uniquement récolter le bruit thermique du récepteur en sortie du corrélateur.

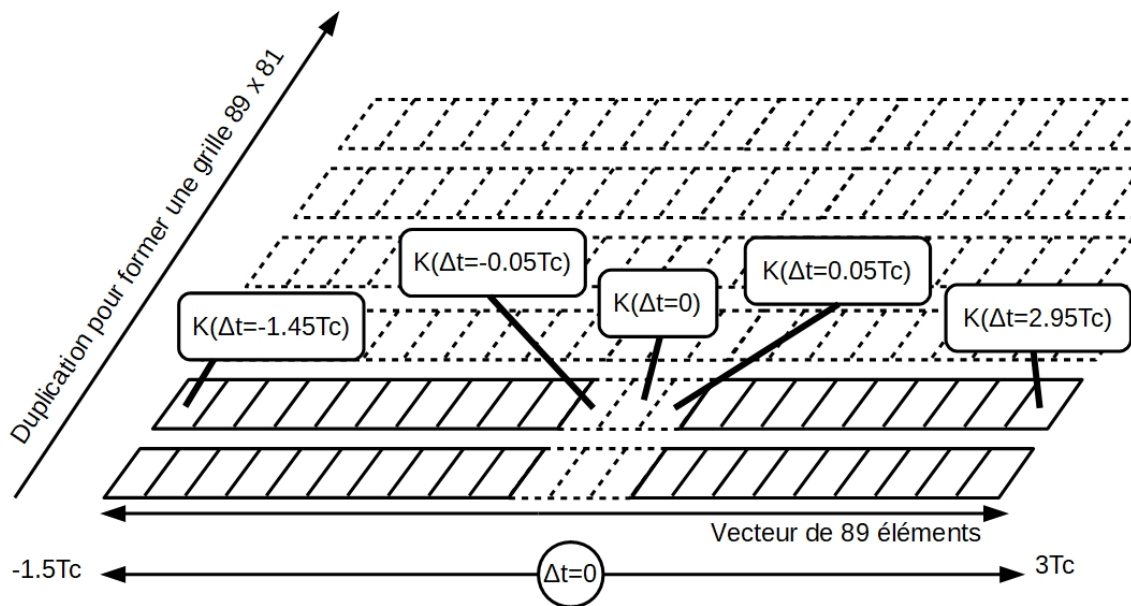


FIGURE 2.4 – Une première grille de 89x81 est construite à partir d'un vecteur d'autocorrélation de 89 éléments calculés entre $-1.5 T_c$ et $3 T_c$. Il est ensuite dupliqué 81 fois afin de former une grille de 89x81 éléments. La grille est la même pour les voies I & Q.

2.5 Génération de sorties de corrélateur

Comme expliqué dans la section 1.2.2 du chapitre 1, l'une des étapes réalisées au sein du corrélateur est la multiplication du signal reçu avec la réplique locale du signal attendu. On a également vu dans cette même section l'utilisation de six sorties de corrélateur (trois par composante du signal). L'objectif du générateur est de reproduire les sorties de corrélateur d'un récepteur GPS conventionnel avec la capacité de fournir beaucoup plus de sorties qu'un corrélateur « classique ».

Le calcul des sorties supplémentaires s'effectue à partir des équations 2.3 et 2.4 des voies I & Q. La construction des images repose sur la multiplication de deux grilles de calculs. La première grille formée est le résultat du calcul d'autocorrélation $K(\Delta\tau)$ pour des retards $\Delta\tau \in [-1.5 T_c, 3 T_c]$ (figure 2.4). Le signal LOS étant considéré en poursuite, le pic de la fonction de corrélation pour le signal LOS est localisée au niveau de la sortie *Prompt*, c'est-à-dire pour un $\Delta\tau = \tau - \tilde{\tau} \approx 0$. Le triangle de corrélation étant de largeur égale à $2 T_c$, l'information du signal LOS est répartie sur l'intervalle $\Delta\tau \in [-T_c, T_c]$. Le multitrajet pouvant atteindre un retard maximal de $\tau_{MP} = 1.5 T_c$ (section 2.3), l'information des signaux LOS et multitrajet est donc répartie sur l'intervalle $[-T_c, 2.5 T_c]$. Des observations au sein de récepteurs réels ont montré que des multitrajets très puissants pouvaient modifier la valeur de $\tilde{\tau}$ de l'ordre de $\pm \frac{T_c}{2}$ [21]. En prenant en compte cette marge d'erreur, l'intervalle de variation de $\Delta\tau$ devient $[-\frac{3}{2} T_c, 3 T_c]$. L'intervalle est divisé selon une période d'échantillonnage choisie à $f_{ech} = 20$ MHz : ainsi, 89 valeurs d'autocorrélation sont générées selon des valeurs $\Delta\tau$ différentes, parmi lesquelles une correspond à la valeur $\Delta\tau = 0 T_c$.

La deuxième grille est le calcul des composantes des voies I & Q pour différents décalages en fréquence Doppler Δf générés entre -250 Hz et 250 Hz (illustration figure 2.5). L'intervalle est divisé en 81 points dont un point correspond à $\Delta f = 0$ Hz.

2.5. Génération de sorties de corrélateur

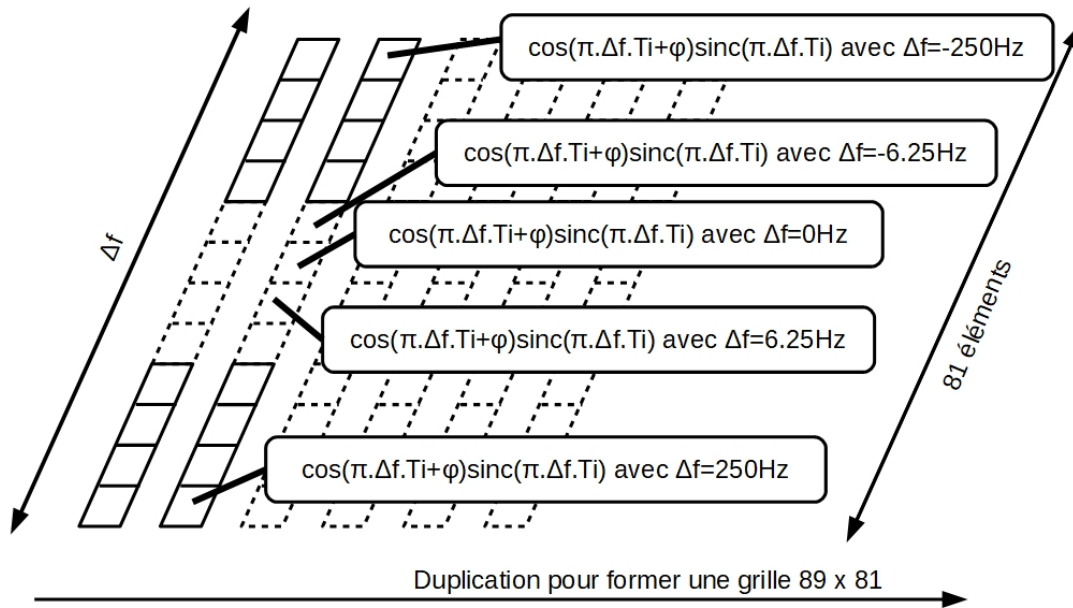


FIGURE 2.5 – Une deuxième grille de 89x81 est construite à partir du calcul sur 81 décalages en fréquence Doppler différents et répartis entre -250 Hz et 250 Hz. La grille de la voie I est construite à partir du modèle décrit par l'équation 2.3 et la grille de la voie Q est construite à partir du modèle décrit par l'équation 2.4. L'exemple illustré ci-dessus concerne la construction de la grille de décalage Doppler pour la voie I.

Les deux grilles de calculs provisoires sont fusionnées afin de former une grille de sorties par voie. Ces grilles de sorties forment un couple d'images de taille respective égale à 89x81 pixels dont chaque élément correspond à un couple $(\Delta\tau, \Delta f)$. Ainsi les images construites contiennent un total de 89x81 couples $(I(\Delta\tau, \Delta f), Q(\Delta\tau, \Delta f))$. Comme illustrées à la figure 2.6, les différentes tailles d'images sont formées à partir d'une sélection précise des différentes sorties. La sortie $\Delta\tau = 0 T_c$ et $\Delta f = 0$ Hz est prise comme point de référence de l'image formée et la décimation³ de la grille de sorties 89x81 permet de compléter le reste de l'image. Par exemple pour une image de taille 10x10 (dont on illustre un exemple à la figure 2.7), on sélectionne les sorties correspondant à $\Delta\tau \in \{-0.8, -0.4, 0, 0.4, 0.8, 1.2, 1.6, 2, 2.4, 2.8\} T_c$ et $\Delta f \in \{-250, -200, -150, -100, -50, 0, 50, 100, 150, 200\}$ Hz. Les images de différentes tailles sont ainsi construites à partir de la même grille de sorties de corrélateur. Par construction, la sortie $(\Delta\tau = 0 T_c, \Delta f = 0 \text{ Hz})$ est toujours présente dans les images générées, quelles que soient leurs dimensions. Dans l'exemple d'une image I en vision 2D ne présentant qu'un signal LOS avec présence de bruit à la figure 2.8, on observe que le pixel localisé par les coordonnées $(\Delta\tau = 0 T_c, \Delta f = 0 \text{ Hz})$ est le plus coloré. C'est une observation du pixel correspondant au pic du signal LOS en poursuite et ici en absence de multitrajet.

Chaque axe de l'image 2D correspond à la variation d'un paramètre : les deux axes horizontaux correspondent à la variation du retard $\Delta\tau$ et du décalage en fréquence Doppler Δf , l'axe vertical correspondant à l'amplitude des signaux. Une image étant composée de N sorties de corrélateur et une donnée étant un couple d'images I & Q, finalement les $2N^2$ sorties de corrélateur correspondant au signal LOS et au multitrajet prennent la forme d'un tenseur 3D de taille $N \times N \times 2$. Le bruit présent sur les images est reproduit avec l'ajout d'images de bruit issues

3. sélection de certains pixels dans une liste de pixels définie.

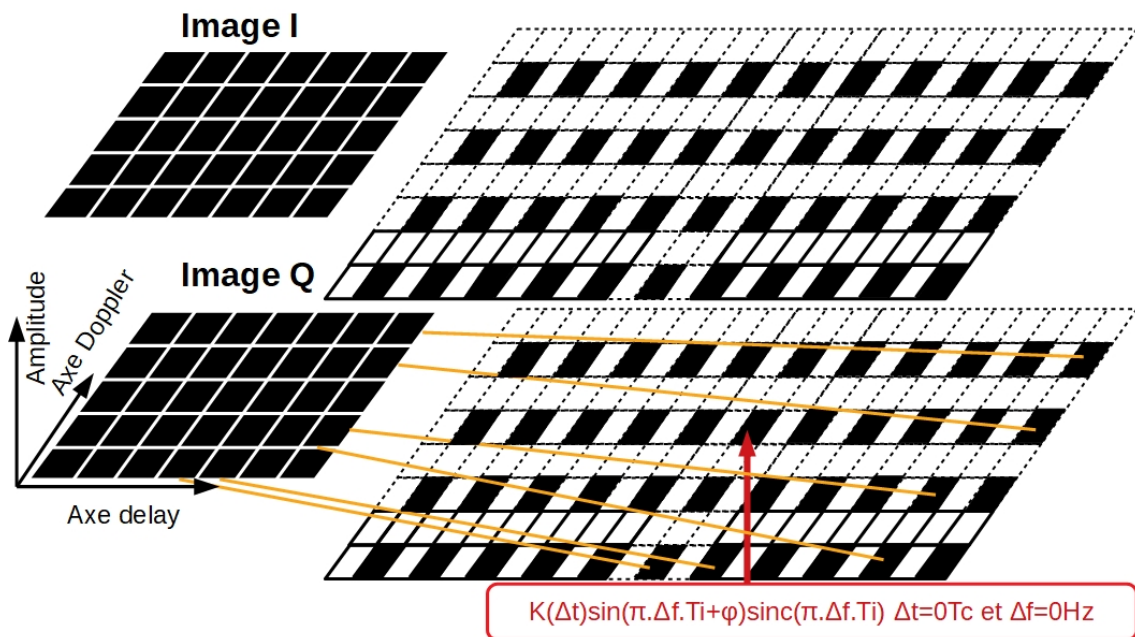


FIGURE 2.6 – Les grilles de retards et de décalages en fréquence Doppler sont multipliées entre-elles : chaque élément de la grille nouvellement formée correspond à un retard précis $\Delta\tau$ ainsi qu'à un décalage en fréquence Doppler Δf . Les images I & Q sont ensuite construites par une sélection de certaines valeurs de $\Delta\tau$ et Δf . Les images de dimensions 80x80, 40x40, 20x20 et 10x10 sont donc issues des mêmes grilles de sorties de corrélateur. Cependant les sorties utilisées pour former les images de différentes dimensions ne sont pas forcément les mêmes. Néanmoins la sortie de corrélateur correspondant au cas ($\Delta\tau = 0 T_c, \Delta f = 0 \text{ Hz}$) est toujours utilisée, quelles que soient les dimensions de l'image construite.

2.6. Construction des différentes images des canaux I & Q

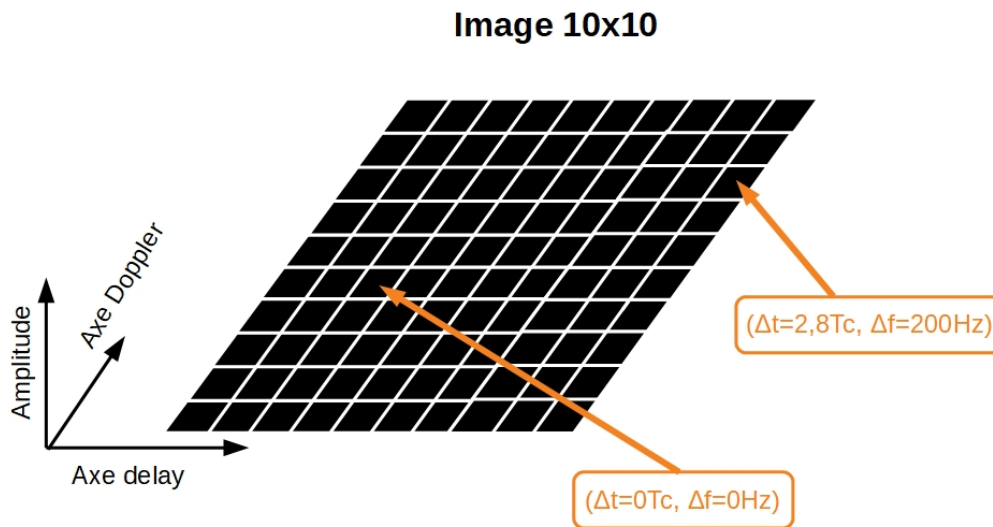


FIGURE 2.7 – Exemple de construction d’une image I de dimension 10x10. Chaque pixel de l’image correspond à un retard $\Delta\tau$ et un décalage en fréquence Doppler Δf donnés.

de mesures provenant d’un récepteur réel (section 2.4). Sa puissance est calculée pour ensuite être nivelée afin d’obtenir le niveau de C/N_0 souhaité.

2.6 Construction des différentes images des canaux I & Q

La construction des images est un processus séquentiel se déroulant en quatre étapes :

1. La première étape consiste à recréer les sorties de corrélateur d’un signal LOS. Supposées en étape de poursuite, les images I & Q créées correspondent à la modélisation du signal donnée par les équations 2.1 et 2.2,
2. La deuxième étape réalise le calcul des sorties correspondant au multitrajet uniquement. Dans l’exemple pris dans cette sous-section, nous avons généré un fort multitrajet inversant le bit de navigation du LOS, soit avec les paramètres ($\tau_{MP} = 0.75 T_c$, $\delta f = 50$ Hz, $\alpha_{MP} = 0.8$, $\phi_{MP} = \pi$),
3. Une fois les images des deux signaux construites, par linéarité des composantes du signal aux sorties du corrélateur, les images résultantes sont la somme des contributions de chacun des signaux sur les sorties. En d’autres termes, le couple d’images I & Q du signal interféré correspond à la somme des deux images I et des deux images Q générées aux étapes 1 et 2,
4. Un terme de bruit réel est rajouté sur chacune des images.

Les figures 2.9 et 2.10 donnent un aperçu des images à la fin de chacune de ces étapes.

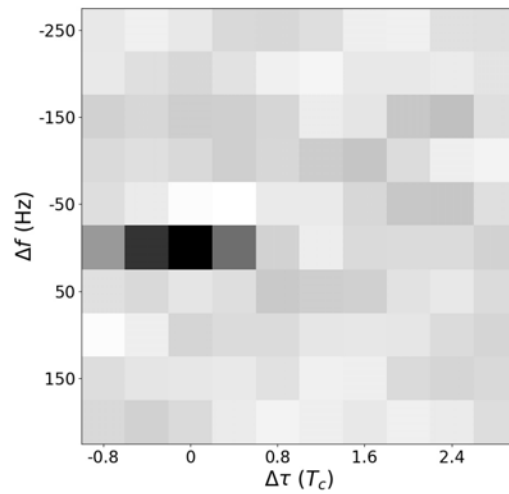
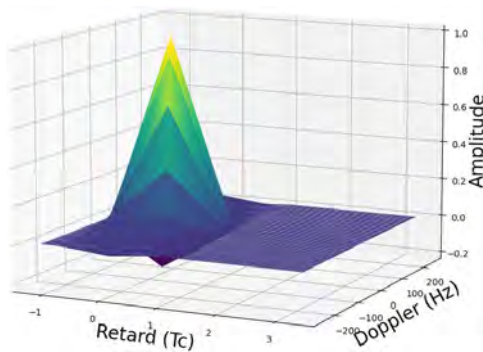
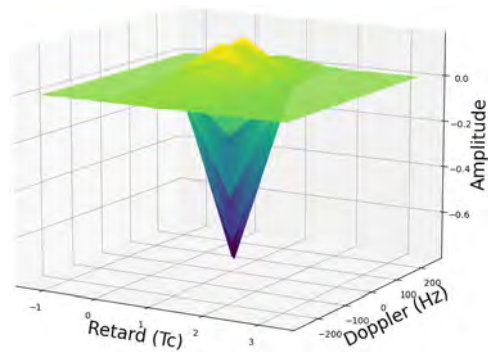


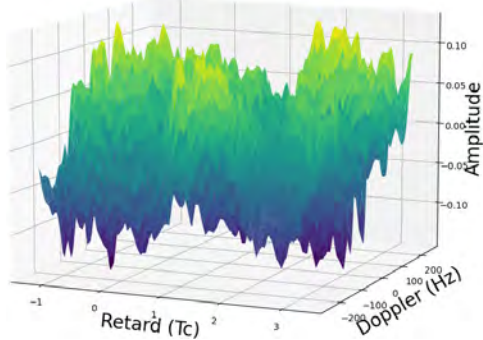
FIGURE 2.8 – Image I de 10x10 pixels affichée avec des contrastes accentués pour une observation du pic principal. C'est l'image qui alimente l'algorithme d'estimation.



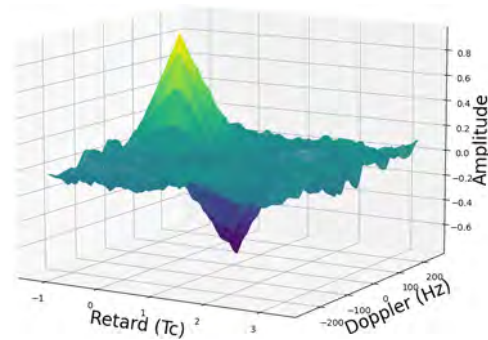
A. Image I pour le signal LOS



B. Image I pour le multirajet



C. Bruit réel aux sorties I



D. Image I finale

FIGURE 2.9 – Génération des différentes sorties de corrélateur pour la composante I d'un signal subissant un multirajet de paramètres $\tau_{MP} = 0.75 T_c$, $\delta f = 50$ Hz, $\alpha_{MP} = 0.8$ et $\phi_{MP} = \pi$.

2.6. Construction des différentes images des canaux I & Q

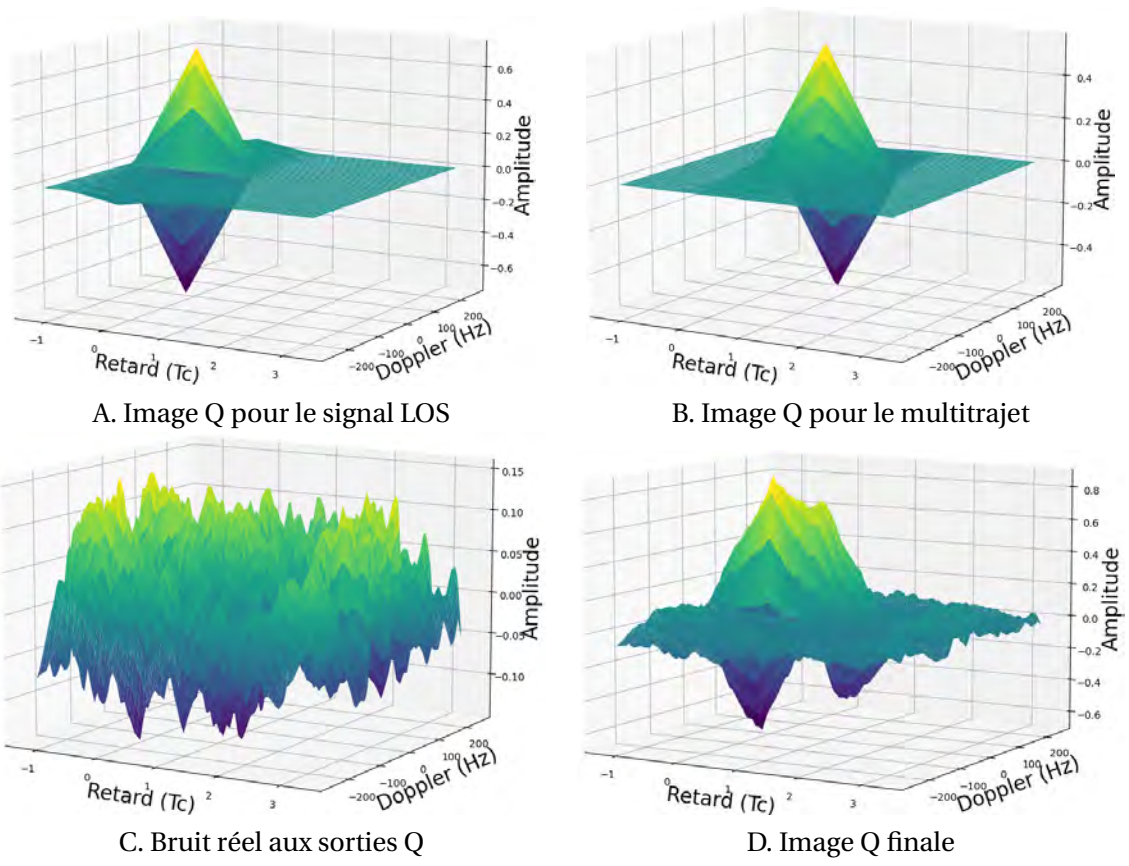


FIGURE 2.10 – Génération des différentes sorties de corrélateur pour la composante Q d'un signal subissant un multitrajet de paramètres $\tau_{MP} = 0.75 T_c$, $\delta f = 50$ Hz, $\alpha_{MP} = 0.8$ et $\phi_{MP} = \pi$.

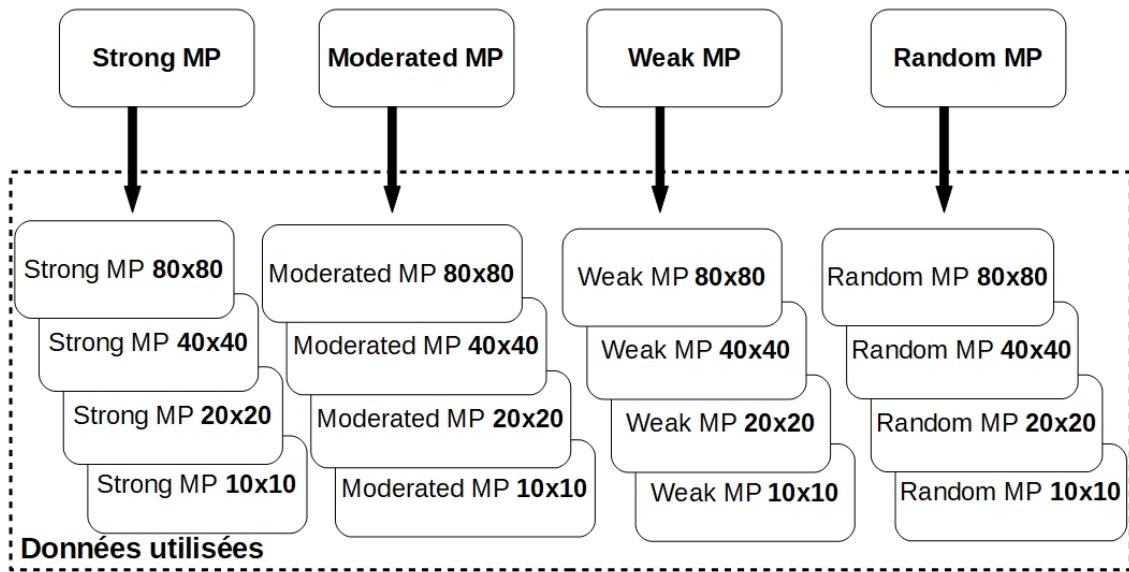


FIGURE 2.11 – 16 différents ensembles de données générés à partir des différentes catégories de multitrajets. Dans un *dataset* donné une seule dimension d’images est disponible. L’algorithme est entraîné puis généralisé sur un seul *dataset* à la fois.

2.7 Ensemble de données des expériences

2.7.1 Dataset régulé par l’amplitude du multitrajet

Les séries d’expériences suivantes ont été divisées en deux jeux de données distincts. Le premier jeu de données trie les multitrajets selon leur puissance. Leur retard, décalage en fréquence Doppler et phase sont aléatoirement générés selon les lois de distribution introduites en section 2.3. Le niveau de C/N_0 des images est fixé à 40 dBHz. Les multitrajets sont ainsi classés selon les trois catégories suivantes :

- Les multitrajets considérés de forte intensité, caractérisés par un coefficient d’atténuation $0.6 \leq \alpha_{MP} \leq 0.9$. Cet ensemble de données sera désigné sous l’appellation de *Strong Multipath* (Strong MP),
- Les multitrajets de puissance « modérée » avec $0.4 \leq \alpha_{MP} \leq 0.6$, appelés *Moderated Multipath* (Moderated MP),
- Les multitrajets de faible amplitude avec $0.1 \leq \alpha_{MP} \leq 0.4$, appelés *Weak Multipath* (Weak MP).

Un ensemble de données regroupant les différentes amplitudes des multitrajets testées est créé ($\alpha_{MP} \in [0.1, 0.9]$) et est référé sous la dénomination *Random Multipath* (Random MP). Pour chaque ensemble de multitrajets, quatre sous-ensembles ont été générés, chacun correspondant à une résolution précise pour les images. Comme illustrée à la figure 2.11, la dimension des images dans une sous-catégorie donnée reste constante. Une image pouvant être composée de 80x80, 40x40, 20x10 ou 10x10 pixels.

2.7.2 Dataset régulé par le niveau de C/N_0

Le deuxième jeu de données répond à une volonté de mesurer la robustesse des performances de l'algorithme par rapport à une dégradation du C/N_0 , c'est-à-dire une augmentation du bruit vis-à-vis des amplitudes des signaux reçus. Les multitrajets sont ici triés selon le niveau de C/N_0 auquel ils ont été générés :

- $C/N_0 = 43$ dBHz : conditions de réception bonnes,
- $C/N_0 = 40$ dBHz : conditions de réception satisfaisantes,
- $C/N_0 = 37$ dBHz : conditions de réception mitigées,
- $C/N_0 = 34$ dBHz : conditions de réception dégradées.

Les multitrajets sont générés selon leurs lois de distribution introduites en section 2.3. Comme pour le jeu de données précédent, chaque ensemble est décliné en quatre sous-ensembles afin de former un ensemble par taille d'images, allant aussi de 80×80 à 10×10 pixels.

2.8 Conclusion

Ce chapitre a présenté le générateur de sorties de corrélateur développé au sein du laboratoire ENAC. Il permet de créer des multitrajets de paramètres aléatoires et produit les images correspondant aux sorties du corrélateur en situation de poursuite et sous influence du multitrajet. Le générateur est capable de produire un nombre de données *labellisées* arbitraire avec une résolution pouvant aller de 80×80 pixels à 10×10 . Le principe du générateur s'appuie sur une subdivision uniforme des espaces de recherche des paramètres de retard $\Delta\tau$ et de décalage en fréquence Doppler Δf , subdivisions qui subissent une décimation afin de sélectionner le nombre de pixels adéquat. Chaque sortie de l'espace de recherche correspond à la valeur de l'opération de corrélation entre le signal total (LOS + multitrajet + bruit) et une réplique locale du signal LOS animée d'un retard et d'une fréquence Doppler précis. Nous avons généré plusieurs ensembles de données (notamment avec des tailles d'images distinctes) à partir du générateur pouvant se résumer à un jeu de données trié selon les amplitudes des multitrajets et un jeu de données trié selon le rapport C/N_0 du signal. Ces différentes données vont être utilisées dans les chapitres 4 et 5 afin de mesurer les performances d'estimation des algorithmes utilisés. La capacité du générateur à produire une grande quantité de données nous a permis d'appliquer des méthodes de *Machine Learning* (ML) notamment des réseaux de neurones, présentés dans le chapitre suivant.

Chapitre 3

Estimation de paramètres par CNN

Sommaire

3.1 Introduction au Deep Learning	42
3.1.1 Machine Learning	42
3.1.2 Deep Learning	42
3.2 Réseaux de neurones	43
3.2.1 Architecture	43
3.2.2 Couches cachées	44
3.2.3 Couche de sortie	45
3.3 Apprentissage	45
3.3.1 Ensembles de données d'apprentissage	48
3.4 Réseaux de neurones à convolution	49
3.4.1 Bloc de convolution	50
3.4.2 Structure globale	50
3.5 Régression par CNN	54
3.6 Utilisation du <i>Machine Learning</i> dans le domaine GNSS	56
3.6.1 Données	56
3.6.2 <i>Features</i> ou caractéristiques	57
3.6.3 Algorithmes de <i>Machine Learning</i>	58
3.7 Conclusion	60

L'objectif de la thèse est l'estimation des paramètres de multitrajets à partir d'images construites par le générateur de sorties de corrélateur introduit au chapitre précédent. En s'inspirant des travaux réalisés par Munin et al. dans [21], [18] et [19], nous appliquons des réseaux de neurones à convolution sur les images I & Q afin de réaliser l'estimation des paramètres de multitrajets. Le chapitre présente les concepts généraux concernant les réseaux de neurones à convolution. Après un préambule portant sur le paradigme de *Machine Learning*, le chapitre introduit l'architecture et le fonctionnement d'un réseau de neurones à partir de l'exemple d'un *Multi-Layer Perceptron* (MLP) : c'est un réseau de neurones simple dont l'architecture est la plus utilisée [67]. La deuxième partie du chapitre est consacrée aux particularités du réseau de neurones à convolution (ou CNN), architecture particulièrement adaptée à l'analyse d'images et utilisée dans les chapitres 4-7 pour les tâches de régression de paramètres sur les images I & Q. La dernière partie du chapitre est un état de l'art portant sur l'utilisation d'algorithmes de *Machine Learning* dans le domaine GNSS, plus particulièrement sur les tâches de détection, estimation et élimination des multitrajets et leurs effets.

3.1 Introduction au Deep Learning

Cette section introduit le concept de *Deep Learning*. Dans le cadre de la thèse, nous allons appliquer des algorithmes de *Deep Learning* sur des images¹ 2D. Plus précisément nous utiliserons des réseaux de neurones à convolution (également appelés CNN) dont nous présenterons les différences par rapport à un perceptron multicouche conventionnel.

La section en annexe D donne davantage de détails autour de l'architecture et du fonctionnement des réseaux de neurones. Elle est destinée à un public non habitué aux concepts de *Machine Learning* et/ou souhaitant plus de précision concernant les réseaux de neurones.

3.1.1 Machine Learning

L'apprentissage automatique, ou *Machine Learning* en anglais, est une branche de l'intelligence artificielle regroupant les différentes approches mathématiques et statistique visant à reproduire un processus d'apprentissage à partir de données. Le *Machine Learning* se différencie d'une programmation habituelle par le paradigme présenté en figure 3.1 : un algorithme classique calcule les sorties de données d'entrée à partir d'une fonction connue et définie. Au contraire, l'algorithme de *Machine Learning* n'a pas connaissance de cette fonction. Pour calculer les sorties, l'algorithme de *Machine Learning* doit retrouver la fonction qui a permis d'obtenir ces sorties à partir des données d'entrée. L'approximation de la fonction recherchée est réalisée lors d'une phase d'apprentissage. Une fois la reconstruction de la fonction effectuée, l'algorithme est en mesure de prédire les sorties.

3.1.2 Deep Learning

Le *Deep Learning*, ou apprentissage profond en français, est une sous-branche du *Machine Learning* désignant des algorithmes, principalement des réseaux de neurones, destinés à analyser des données à travers l'utilisation de couches cachées successives [68]. Les algorithmes de *Deep Learning* sont sollicités dans des domaines comme la reconnaissance de langage [69] ou l'analyse d'images avec la restauration de pixels [70]. L'une des particularités de ces algorithmes réside dans leur flexibilité vis-à-vis du problème posé : contrairement à des algorithmes spécifiques à une tâche précise, les algorithmes de *Deep Learning* s'adaptent en fonction des données les alimentant [71]. Cette particularité est liée à un avantage apporté par cette famille d'algorithme qui est l'autonomie vis-à-vis de l'extraction d'information [72]. Désigné sous le nom de *features extraction* ou également de *features engineering* [73], l'extraction des *features* consiste à récupérer les éléments informatifs d'une donnée d'entrée : par exemple reconnaître les pixels liés aux doigts tout en identifiant leur position 3D sur une image 2D [74]. La construction de ces *features* est directement effectuée par le réseau : c'est donc un processus automatique [73], permettant l'adaptation du réseau en fonction de la tâche accordée (classification chien/chat [75], régression sur les coordonnées d'un corps [76]...). Ainsi pour une même structure *hardware* (c'est-à-dire l'architecture d'un réseau de neurones), les tâches demandées peuvent être complètement différentes.

1. Représentations des signaux en sortie de corrélateur dont le processus de génération initial est explicité chapitre 2.

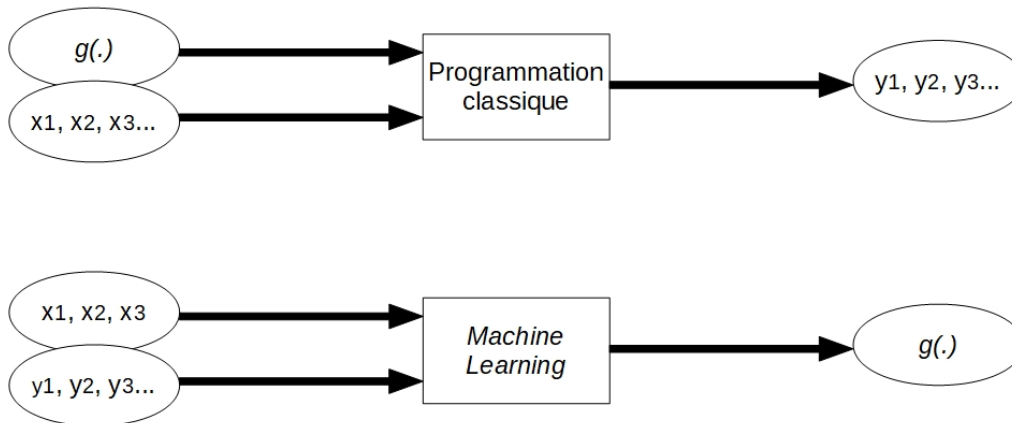


FIGURE 3.1 – Un programme informatique « classique » applique une fonction (ici g) sur des données en entrée $(x_i)_{1 \leq i \leq N} \in X$ afin d'en obtenir les sorties $(y_i)_{1 \leq i \leq N} \in Y$, $N > 0$ étant le nombre de données à disposition, X l'ensemble de données et Y l'ensemble image des données d'entrée. Le modèle de *Machine Learning* a connaissance des données d'entrée et de sorties et cherche à retrouver la fonction g qui a permis d'obtenir les sorties (également appelées étiquettes ou *labels*) à partir des données d'entrée.

3.2 Réseaux de neurones

3.2.1 Architecture

Les réseaux de neurones sont une famille d'algorithmes. Leur fonctionnement s'appuie sur une analyse en *forward* (sens entrée \rightarrow sortie) de données d'entrée à travers une succession de couches. Chacune de ces couches est composée d'un nombre fixé de neurones (également appelés noeuds). L'objectif du réseau de neurones est la construction d'une fonction f capable d'approcher le modèle ayant permis de générer les données $x \in X$ d'un problème (X étant l'ensemble des données du problème). À chaque donnée x correspond un *label* $y \in Y$ où Y est l'ensemble des *labels* des données de X . Dans l'exemple donné figure 3.2 extrait du problème de reconnaissance de chiffres manuscrits sur le *dataset* MNIST, la donnée x est une image numérisée d'un chiffre manuscrit de *label* $y = 9$. Le réseau est entièrement défini par ses paramètres $\theta \in \Theta$ avec Θ un espace paramétrique. θ est composé des poids neuronaux, contenant à la fois les liaisons entre neurones de couches successives et les biais présents à chaque couche. La construction de la fonction f passe par la recherche d'une valeur de θ capable d'approximer au mieux les *labels* des données soit :

$$\forall x \in X, f(x; \theta) \approx y$$

La recherche du meilleur θ s'effectue lors d'une phase d'apprentissage, processus détaillé à la section 3.3.

Les MLP ou *Deep feedforward networks* sont une architecture de réseau de neurones très répandue. Leur architecture peu complexe peut être représentée par la composition de plusieurs



FIGURE 3.2 – Image du *dataset* MNIST.

fonctions. Soit une donnée d'entrée $x \in X$ et la fonction f construite par le réseau à partir de son vecteur de paramètres θ . Alors f est le résultat de la composition des fonctions, chacune correspondant à l'analyse exercée par une couche. En posant $f^{(i)}$ la fonction représentant la couche $i \in [1 ; d]$ du réseau où $d \in \mathbb{N}^*$ est la profondeur du réseau, on obtient :

$$f(x; \theta) = f^{(d)} \left(\dots f^{(i)} \left(f^{(i-1)} \left(\dots f^{(2)} \left(f^{(1)}(x) \right) \right) \right) \right)$$

$f^{(1)}$ est la couche analysant directement la donnée d'entrée x : c'est la couche d'entrée du réseau. $f^{(d)}$ est la dernière couche du réseau : c'est la couche de sortie. La sortie des neurones de la couche $f^{(d)}$ sont également les sorties du réseau de neurones.

3.2.2 Couches cachées

Les couches intermédiaires $(f^{(i)})_{1 < i < d}$ appliquent une analyse aux informations présentes à leur entrée découlant des couches précédentes et renvoient en sortie une représentation de ces mêmes informations pour les couches suivantes [77]. Plus connues sous le nom de couches cachées, les sorties de ces couches sont inconnues. Pour une donnée $x \in X$, il n'y a pas connaissance des sorties $y^{(i)}$ pour chaque couche $f^{(i)}$: seul le résultat $y^{(d)}$ issu de la couche de sortie est connu. Un neurone d'une couche cachée est relié à tous les neurones de la couche précédente, avec une pondération propre à chaque bras neuronal. Cette configuration est plus connue sous le nom de couche dense, ou *fully-connected layer*. La liaison neurone à neurone n'est pas toujours valable : dans des structures comme les CNN, un neurone d'une couche de convolution est relié à un sous-ensemble de neurones de la couche suivante [78]. En revanche un neurone n'est pas connecté aux autres neurones partageant sa couche. Dans le cas des réseaux de neurones récurrents ou *Recurrent Neural Network* (RNN) un neurone peut reboucler sur lui-même : sa sortie est aussi une de ses entrées avec sa propre pondération.

Au sein d'un neurone d'une couche cachée $i \in [1 ; d]$, une transformation affine est appliquée sur les entrées du neurone. Soit x l'entrée de la couche, alors la sortie d'un neurone de la couche est de la forme :

$$z^{(i)} = (W^{(i)})^t x + b^{(i)}$$

où

- $(W^{(i)})^t$ est la transposition du vecteur poids de la couche i ,
- $b^{(i)}$ est le biais de la couche i .

$z^{(i)}$ est linéaire par rapport à x en tant que combinaison d'opérations linéaires par rapport à x . Afin de rajouter de la non-linéarité dans la couche, une fonction d'activation non-linéaire est appliquée à $z^{(i)}$ afin de briser le caractère linéaire des sorties de neurones de cette couche. Des exemples de fonctions d'activation classiques sont donnés à l'annexe D. Notons h la fonction non-linéaire appliquée aux sorties de la couche. Alors la sortie d'un neurone de la couche est :

$$y^{(i)} = h(z^{(i)}) = h\left(\left(W^{(i)}\right)^t x + b\right)$$

3.2.3 Couche de sortie

Le résultat du traitement *forward* des données par le réseau donne la prédiction du réseau correspondant à la donnée d'entrée x . Située au niveau de la dernière couche, la sortie d'un réseau de neurones prend la forme suivante :

$$y^{(d)} = f(x)$$

avec $y^{(d)} \in \mathbb{K}$, \mathbb{K} étant un ensemble de dimension quelconque. En effet, la couche de sortie d'un réseau de neurones peut être composé de plusieurs sorties. Un cas particulier de sorties multiples est celui de la sortie *softmax*, illustré en figure 3.3. Cas souvent rencontré pour des tâches de classification par réseaux de neurones, cette sortie généralise la régression logistique pour des classes multiples : elle permet de prédire l'appartenance à une classe parmi $K \in \mathbb{N}^*$ classes. Elle attribue une probabilité d'appartenance de la donnée traitée $x \in X$ pour chaque classe $k \in \llbracket 1, K \rrbracket$. Le vecteur de sortie $y^{(d)} = (y_1^{(d)}, \dots, y_k^{(d)}, \dots, y_K^{(d)}) \in Y$ (Y un espace de dimension K) contient les probabilités d'appartenance à chaque classe :

$$\forall k \in \llbracket 1, K \rrbracket, y_k^{(d)} = \frac{e^{n_i^s}}{\sum_{k=1}^K e^{n_k^s}}$$

avec

- $y_k^{(d)}$ le $k^{\text{ième}}$ élément du vecteur de sortie $y^{(d)}$,
- n_i^s la sortie du $i^{\text{ième}}$ neurone (situé dans la couche de sortie).

Remarque : La somme des éléments du vecteur $y^{(d)}$ est égale à 1 par théorème des probabilités totales. Cette propriété sera notamment exploitée dans les chapitres 5, 6 et 7 pour obtenir des distributions de probabilités en sortie de la couche *softmax*.

3.3 Apprentissage

Le réseau de neurones, comme tout algorithme de *Machine Learning*, fonctionne en deux phases. La première est la phase d'entraînement pendant laquelle il y a une prise de connaissances des données et une reconstruction de la fonction f permettant de relier les données d'entrées x à leur *labels* y . Une fois cette phase terminée, l'algorithme est dit entraîné et prêt à donner des prédictions précises sur des données : cette deuxième étape est connue comme la phase de généralisation. Il existe une catégorie d'algorithme de *Machine Learning* continuant leur processus d'apprentissage lors de la phase d'inférence. Appelé apprentissage *on-line*, ce type d'apprentissage proactif vise à améliorer les performances avec une correction du réseau après chaque donnée traitée pendant la phase de généralisation. Dans le cadre de la thèse, l'apprentissage est exclusivement *off-line* : le réseau cesse d'évoluer et fige ses poids et biais

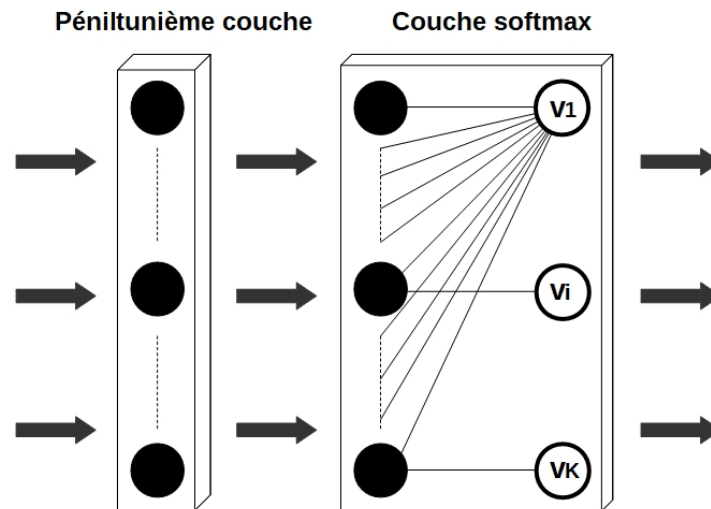


FIGURE 3.3 – Couche *softmax* dans le cas d’une classification de données parmi $K \in \mathbb{N}^*$ classes. Elle calcule un vecteur de probabilité de dimension correspondant à la taille de sa couche (nombre de neurones la composant). Ici le vecteur $v = (v_1, \dots, v_K)$ renvoyé est de taille K .

une fois l’étape d’entraînement terminée. Parmi les types d’inférence, il y a la déduction, l’abduction (faire des hypothèses sur les données) et l’induction, qui consiste à généraliser suite à une observation sur plusieurs exemples. C’est ce dernier type d’inférence que l’on cherche à réaliser.

L’apprentissage s’appuie sur deux processus : le *forward* et *backward* des informations. L’étape de *forward*, illustrée à la figure 3.4, permet d’obtenir des prédictions $y^{(d)}$ des données x , le traitement des données s’effectuant successivement par chacune des couches du réseau. Dans le cadre d’un apprentissage supervisé, une donnée x possède un *label* y avec lequel la différence avec la prédiction du réseau peut être quantifiée : c’est l’erreur de prédiction du modèle. Dans le contexte d’un problème de régression, par exemple lors d’une estimation de chiffre à partir d’une image (tâche de régression), l’erreur du modèle peut être calculée de différentes manières : par exemple avec la valeur absolue de la différence entre le *label* et la prédiction $|y - y^{(d)}|$ ou avec l’erreur quadratique $(y - y^{(d)})^2$.

Lors d’un apprentissage, l’erreur de prédiction (ou pertes) du modèle est calculée sur un ensemble de données appelé *batch* d’apprentissage. L’impact du *batch* sur l’apprentissage du réseau est détaillé dans l’annexe D. En plus des différentes erreurs de prédiction du modèle peut être ajouté un terme de pénalisation, sur la valeur des pondérations du réseau par exemple. C’est une technique visant à pénaliser la complexité du réseau afin de prévenir le phénomène de sur-apprentissage, symptôme d’une adaptation exagérée du réseau vis-à-vis des données d’apprentissage et se traduisant par une altération des performances en phase de généralisation. Elle fait partie des méthodes de régularisation qui feront l’objet de la section 5.1.1. La perte d’un réseau de neurones est ainsi définie comme étant la somme de ses erreurs de prédiction

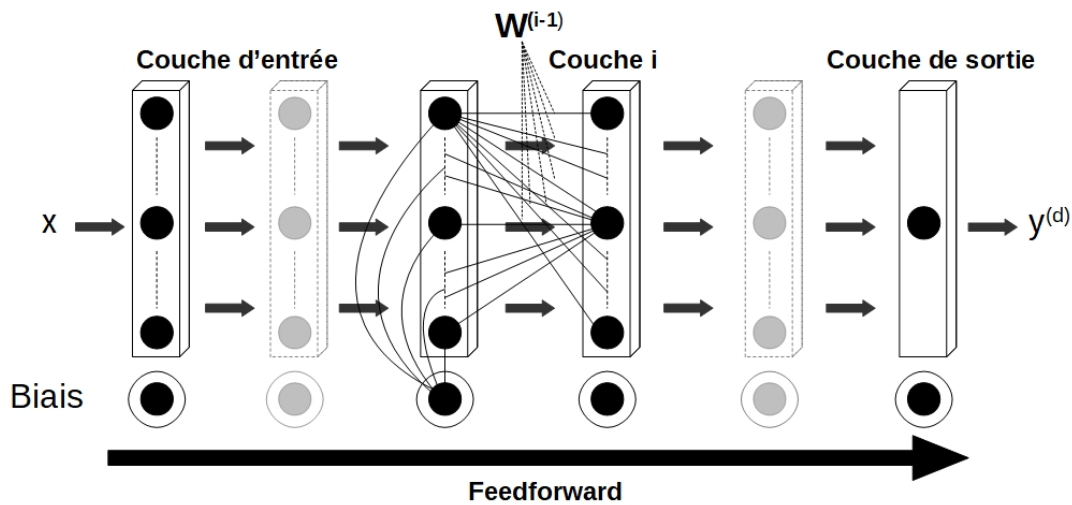


FIGURE 3.4 – *Feedforward / Forward* : la donnée d'entrée x est analysée de manière séquentielle par les couches du réseau. La sortie d'une couche $i - 1$ est l'entrée d'une couche i . La sortie du réseau $y^{(d)}$ correspond à la sortie de la dernière couche du réseau (pour une profondeur d dans cet exemple) : c'est la prédiction correspondant à la donnée x .

sur les données du *batch* plus du terme de pénalisation :

$$E(B, \theta) = \underbrace{\sum_{x \in B} \text{loss}(y, y^{(d)})}_{\text{erreurs de prédiction}} + \underbrace{\lambda \text{reg}(\theta)}_{\text{pertes régularisées}}$$

où

- $\text{loss}(y, y^{(d)})$ est la fonction de perte calculant la différence entre le *label* y d'une donnée x et sa prédiction $y^{(d)}$,
- B le *batch* d'apprentissage,
- $\text{reg}(\theta)$ une fonction de régularisation,
- λ le taux de régularisation.

Au cours de l'apprentissage, des *feedforward* de *batch* d'apprentissage différents vont permettre au réseau de calculer des pertes. La rétropropagation du gradient adapte les poids du réseau en fonction des pertes mesurées : c'est le principe de *back-propagation* avec une remontée des informations sortie \rightarrow entrée. L'apprentissage est une alternance de processus de *forward* et *backforward* d'informations. Le traitement des données est effectué un nombre de fois arbitrairement fixé et appelé *epochs*. Chaque correction des poids et biais du réseau vise à diminuer l'erreur entre la prédiction et le *label* pour les prochains traitements *forward* de la donnée. Le paramètre θ du réseau est ainsi mis à jour tout au long de l'apprentissage afin de réduire les pertes du modèle au fur et à mesure des *epochs*. La phase d'apprentissage est ainsi définie par plusieurs éléments comme le nombre d'*epochs* consacrées à l'entraînement du réseau, le nombre de données contenues dans un *batch* d'apprentissage, la profondeur du réseau ou la valeur du pas d'apprentissage utilisé pour la rétropropagation du gradient : tous ces éléments appartiennent à la catégorie des hyper-paramètres d'un réseau. Contrairement

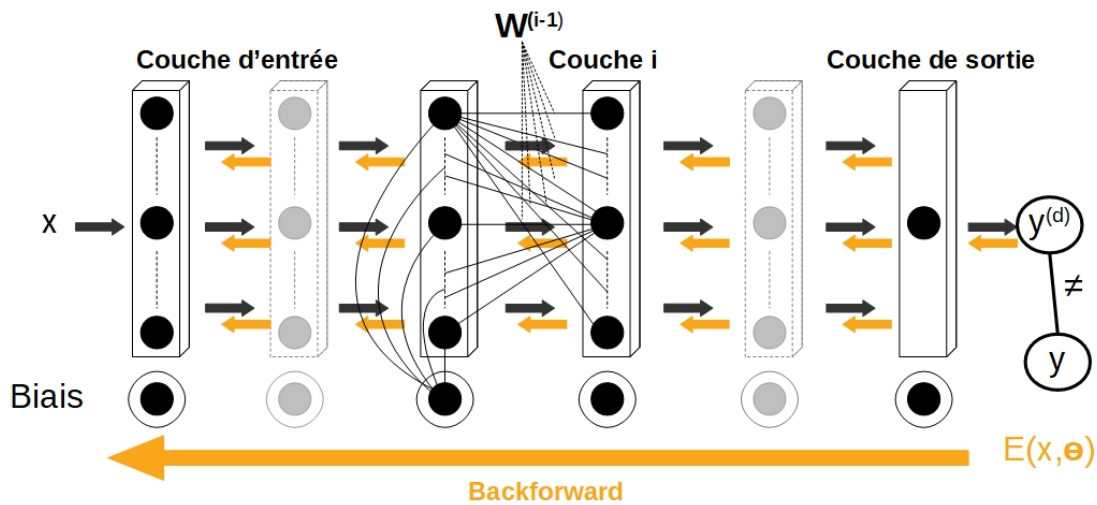


FIGURE 3.5 – *Backward* : la prédiction $y^{(d)}$ est comparée au *label* y de la donnée d'entrée x . Après le calcul de la perte sur l'ensemble X , la perte est propagée dans le réseau afin de corriger les poids $W^{(i)}$ et biais $b^{(i)}$ des différentes couches du réseau, afin d'obtenir une perte moins élevée au prochain *forward* de la donnée x .

aux poids et biais du réseau, les hyper-paramètres n'évoluent pas au cours de l'apprentissage. Par leur impact non négligeable sur l'apprentissage du réseau, des méthodes plus ou moins efficaces ont été développées afin d'effectuer la recherche des hyper-paramètres optimaux. Malgré l'existence de ces méthodes, la recherche des hyper-paramètres optimaux d'un réseau reste un problème difficile. Une fois la phase d'apprentissage terminée, le réseau peut entamer la phase de généralisation. Lors de cette phase, seul le processus de *feedforward* est exécuté avec des données inédites, c'est-à-dire non rencontrées lors de la phase d'apprentissage et n'ayant donc ainsi pas servi à la mise à jour du paramètre θ du réseau. L'étape de généralisation est celle pendant laquelle les prédictions des données inédites est effectuée. Les résultats des chapitres suivants sont les performances de généralisation des différents réseaux entraînés.

En résumé, l'apprentissage d'un réseau de neurones est un problème d'optimisation avec la recherche d'une minimisation des facteurs de pertes et de régularisation par rapport aux poids du réseau à partir des données d'apprentissage :

$$\min_{\theta \in \Theta} E(X, \theta)$$

3.3.1 Ensembles de données d'apprentissage

Lors de l'apprentissage du réseau, le *dataset* entier est divisé en plusieurs ensembles de données qui seront utilisés aux différentes phases. Dans le cadre de la thèse, nous diviserons le *dataset* en trois sous-ensembles comme décrits ci-dessous :

1. l'ensemble d'**entraînement** : contenant le plus de données parmi les trois sous-ensembles, il sert au calcul des erreurs de l'algorithme et à l'ajustement des poids et biais du réseau lors de la phase d'entraînement,

2. l'ensemble de **validation** : servant de mini généralisation, les données de validation permettent d'effectuer des prédictions afin d'évaluer les performances après chaque mise à jour des poids et biais pendant la phase d'entraînement,
3. l'ensemble de **test** : ensemble de données inédites pour l'algorithme. Il permet l'évaluation des capacités de généralisation du réseau à la fin du processus d'entraînement.

Remarque : Il est fondamental que ces ensembles de données soient disjoints sous peine de fausser les résultats obtenus. Si les ensembles sont disjoints, les données de validation et de tests permettent la détection d'éventuels phénomènes de sous-apprentissage ou sur-apprentissage [79].

Les réseaux de neurones en profondeur sont sensibles au bruit présent dans les caractéristiques (variables) d'entrée. Des caractéristiques de grande dimension induisent un risque plus élevé de sur-apprentissage [80]. Une grande quantité de données (notamment lors de l'apprentissage) est alors préconisée afin de devenir robuste face à cette perturbation [77] en plus de diminuer le risque de sur-apprentissage pouvant survenir lorsque les données d'apprentissage sont en nombre insuffisant [81]. Cependant l'augmentation du nombre de données demandera un nombre d'itérations (*epochs*) plus élevé pour converger [79].

3.4 Réseaux de neurones à convolution

Depuis 2012, les réseaux de neurones à convolution ont gagné en popularité dans le traitement d'images par technologie *Deep Learning* (DL) [79]. La principale différence entre un CNN et un réseau de neurones conventionnel comme le MLP étudié à la section précédente se situe au niveau de leur structure en basses couches (voir figure). Les couches denses de plus bas niveaux sont remplacées par des blocs de convolution (couches de convolution et de sous-échantillonnage) qui filtrent les images en entrée avec des matrices de poids. Un neurone d'une couche de convolution est relié à un sous-ensemble de neurones de la couche précédente. Ce sous-ensemble de neurones est relié au neurone de la couche par une matrice de poids. Dans une même couche, tous les neurones partagent la même matrice de poids, seul le sous-ensemble de neurones auquel ils sont reliés diffère. L'unicité de pondération pour tous les neurones d'une même sous-couche, également appelée *weight sharing*, permet de considérablement réduire l'empreinte mémoire du réseau [81]. Le nombre de poids à gérer est plus beaucoup plus faible puisque que l'on passe d'une liaison neurone à neurone d'une couche à une couche suivante à une liaison par matrice commune de taille excédant rarement les 5×5 pixels [82]. Cette structure en *weight sharing* a montré une augmentation des performances par rapport à des algorithmes neuronaux plus classiques [83, 84, 85].

L'idée du CNN est d'appliquer un balayage de l'image d'entrée avec un noyau de convolution afin de détecter des lignes, des bordures... L'analyse d'images est différente entre un perceptron multicouche conventionnel et un réseau de neurones à convolution. Dans le cas d'un MLP conventionnel, l'algorithme effectue une approche globale de l'image en entrée alors que dans celui du CNN, l'approche est locale avec plusieurs analyses sur des régions restreintes de l'image. Dans le cas du CNN, les neurones du bloc de convolution peuvent être assimilés à des unités de détection de *features* sur zone locale [78]. Cette approche permet notamment la prise en compte des informations de voisinage entre pixels. Une multitude d'informations sont extraites au cours du *forwarding* de la donnée à travers le réseau parmi lesquelles la corrélation entre pixels d'une même zone locale.

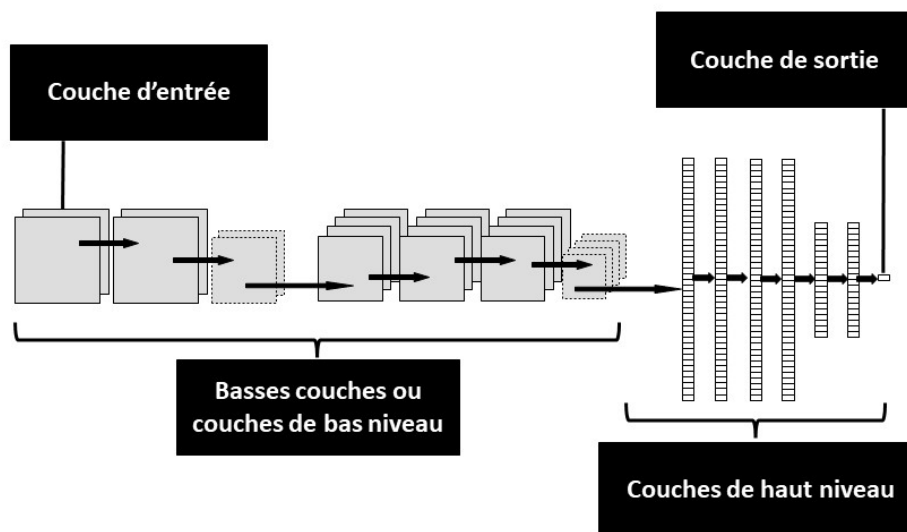


FIGURE 3.6 – Schéma d'un réseau de neurones à convolution. Sa structure est composée d'une première section contenant uniquement de couches de convolution et de sous-échantillonnage. La deuxième section est composée d'une succession de couches neuronales denses. Par rapport à un réseau de neurones classiques comme le MLP, seule la structure en basses couches est modifiée.

3.4.1 Bloc de convolution

Le bloc de convolution est composé d'une succession de couches de convolution de même taille suivies d'une couche de sous-échantillonnage. Comme illustré sur la figure 3.7, les blocs de convolution sont placés les uns à la suite des autres afin de travailler sur les analyses d'images des blocs précédents.

3.4.2 Structure globale

Un réseau CNN traditionnel a une structure en deux parties : la partie convolutive et la partie dense. La première partie est consacrée à l'extraction des *features* à partir des images en entrée et est composée de blocs de convolution successifs. Les blocs de convolution tendent à réduire la dimension des images analysées (on parle aussi de *feature maps*) avec une augmentation de leur nombre avec la profondeur du réseau [86]. La deuxième partie est uniquement composée de couches cachées entièrement connectées avec une couche de sortie. Cette partie est consacrée à l'analyse des données hautement informatives construites par la partie convolutive du réseau. La liaison entre la partie convolutive formée d'images et la partie *fully-connected* est effectuée par une couche *Flatten* qui adapte la carte convolutive (le résultat obtenu après plusieurs couches de convolution) en vecteur compatible avec la structure des couches denses [87].

Couche de convolution

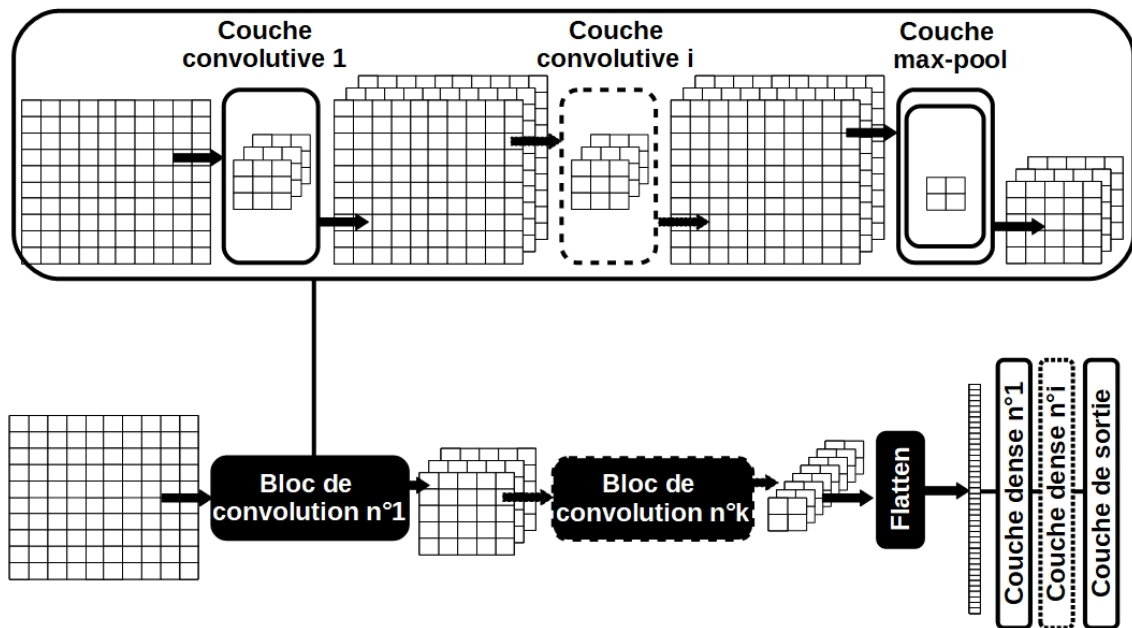


FIGURE 3.7 – La première partie est composée de bloc de convolution avec chacun leur nombre de filtres et de fonctions d'activation propres. Les *features maps* diminuent en dimension mais augmentent en nombre au fur et à mesure des blocs de convolution. Les cartes de convolution finales après le bloc de convolution $k \in \mathbb{N}^*$ sont traitées par la couche de *Flatten* pour rendre leur traitement compatible par les couches denses dans la deuxième partie du réseau. Elles analysent les données hautement informatives extraites et transformées par la partie convolutive du réseau. La propagation *forward* des données se termine par la couche de sortie, pouvant prendre une multitude de formes : neurone unique ou multi-sorties (ex : *softmax*).

3.4. Réseaux de neurones à convolution

La particularité des couches de convolution est de considérer une matrice de poids plutôt que des connexions neurones à neurones comme dans les couches denses vu précédemment. La matrice de poids, également appelée filtre ou détecteur de caractéristiques, est l'élément qui permet l'extraction d'informations sur l'image en entrée. L'opération de convolution exercée par le filtre sur l'image consiste en l'analyse d'une zone délimitée de l'image (la zone recouverte par le filtre dans la figure 3.8) afin d'extraire les informations contenues sous la forme d'un flottant, et formant un pixel de l'image en sortie de couche. L'opération est répétée selon les axes de l'image d'entrée (ex : horizontal et vertical pour une image en 2D) de façon à couvrir l'entièreté de l'image d'entrée. Le résultat des différentes opérations de convolution du filtre avec les différentes zones de l'image en entrée de la couche forme une nouvelle image appelée *feature map*. Le balayage de l'image d'entrée dépend de plusieurs paramètres :

- la taille du filtre à appliquer,
- le *stride* choisi : le déplacement en nombre de pixels du filtre sur l'image (un *stride* de 2 revient à décaler la tuile de 2 pixels). Plus celui-ci est élevé et plus la *feature map* en sortie sera de taille réduite. . . Le *stride* peut être différent selon les axes de l'image afin d'appliquer une analyse différente en fonction des axes,
- le *padding* ou remplissage par zéros : ajout de pixels à la valeur numérique nulle en bordure d'images. L'opération de convolution par une matrice différente de 1×1 donne une *feature map* de dimension inférieure à l'image en entrée de couche. Le *padding* peut être utilisé pour conserver les mêmes dimensions qu'en entrée sans perturber l'analyse des pixels aux frontières de l'image par les filtres de convolution.

Le calcul des pixels formant les *features maps* est illustré figure 3.8 et décrit par l'équation suivante :

$$z_{ijk} = b_k + \sum_{u=1}^{f_h} \sum_{v=1}^{f_w} \sum_{k'=1}^{f_{n'}} x_{i'j'k'} w_{uvk'k}$$

avec

- z_{ijk} la valeur du pixel ligne i colonne j de la *feature map* k ,
- b_k le biais de la couche pour la *feature map* k ,
- $i' = us_h + f_h - 1$,
 - s_h le *stride* vertical du filtre de convolution,
 - f_h la taille du filtre selon l'axe vertical (combien de lignes sont multipliés),
- $j' = vs_w + f_w - 1$
 - s_w le *stride* horizontal du filtre de convolution,
 - f_w la taille du filtre selon l'axe horizontal (combien de colonnes sont multipliés),
- $x_{i'j'k'}$ le pixel ligne i' colonne j' de la *feature map* k' dans la couche précédente,
- $w_{uvk'k}$ poids de connexion ligne u colonne v du filtre de convolution entre des *features maps* k' et k ,
- $f_{n'}$ le nombre de *features maps* dans la couche précédente.

Les filtres sont des matrices de poids mises à jour lors des descentes de gradient pendant l'apprentissage au même titre que les neurones des couches denses en vue de diminuer les pertes sur le *dataset* d'apprentissage. Le balayage de l'image d'entrée par le filtre octroie au CNN des propriétés comme la résistance au décalage et à la déformation des images. À mesure que les images traversent les différentes couches de convolution, les *features maps* générées sont de plus en plus informatives. En effet, elles représentent l'extraction de *features* à travers différents filtres. Un filtre recherche une *feature* spécifique [87] : plus il y a de filtres et plus le nombre de

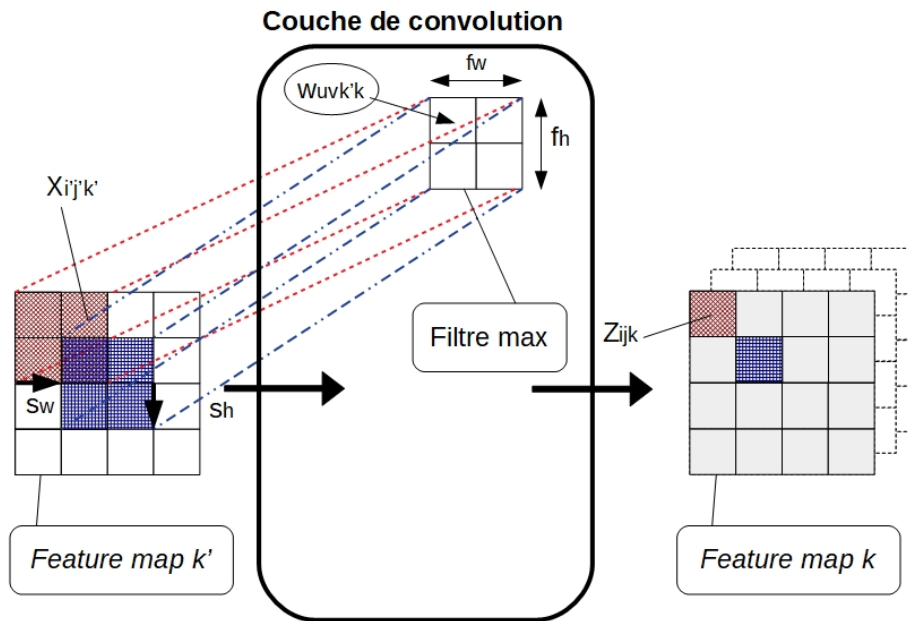


FIGURE 3.8 – Détail du calcul d'une *feature map* générée à partir d'une *feature map* d'une couche précédente. Le résultat de la *feature map* dépend des poids du filtre, de sa dimension, des *strides* horizontaux et verticaux.

features constructibles est élevé [88]. Le nombre de filtres est donc un hyper-paramètre d'un réseau CNN [86].

Remarque : les matrices de convolution peuvent se chevaucher si la valeur de *striding* est faible (par exemple dans la figure 3.8 avec un *striding* de 1 pour des filtres de dimensions 2×2). Un même pixel va être traité plusieurs fois par la même matrice de poids mais avec des pixels voisins différents et des poids matriciels potentiellement différents (n'étant plus multiplié par le même coefficient de la matrice).

Couche de sous-échantillonnage

La couche de sous-échantillonnage (ou *pooling layer*) diminue la dimension des *features maps*. Ressemblant à la couche de convolution, sa fonctionnalité est complètement différente. Une couche de *pooling* ne possède pas de poids : leur but n'est pas d'analyser l'image via une transformation linéaire (comme dans les couches de convolution [73]) mais d'y appliquer une opération non-linéaire avec l'extraction d'une seule valeur parmi une zone de pixels (illustration figure 3.9). Les filtres de sous-échantillonnage, également appelés *kernels*, permettent d'extraire les informations les plus critiques des *features maps* [81]. Ils sont généralement de faibles dimensions (2×2 ou 3×3) car le regroupement effectué par l'opération de *pooling*, bien que conservant l'information la plus importante de la tuile, reste synonyme de pertes d'informations. En diminuant les dimensions des *features maps*, ils limitent le nombre de coefficients des matrices de convolution à calculer dans l'ensemble du CNN et permettent une phase d'apprentissage plus rapide [73]. Plusieurs types de *pooling* existent : les plus fréquemment utilisées sont le *max-pooling*, qui extrait le pixel de valeur numérique maximale et l'*average pooling* qui calcule la moyenne des valeurs numériques des pixels extraits par le *kernel*.

3.5. Régression par CNN

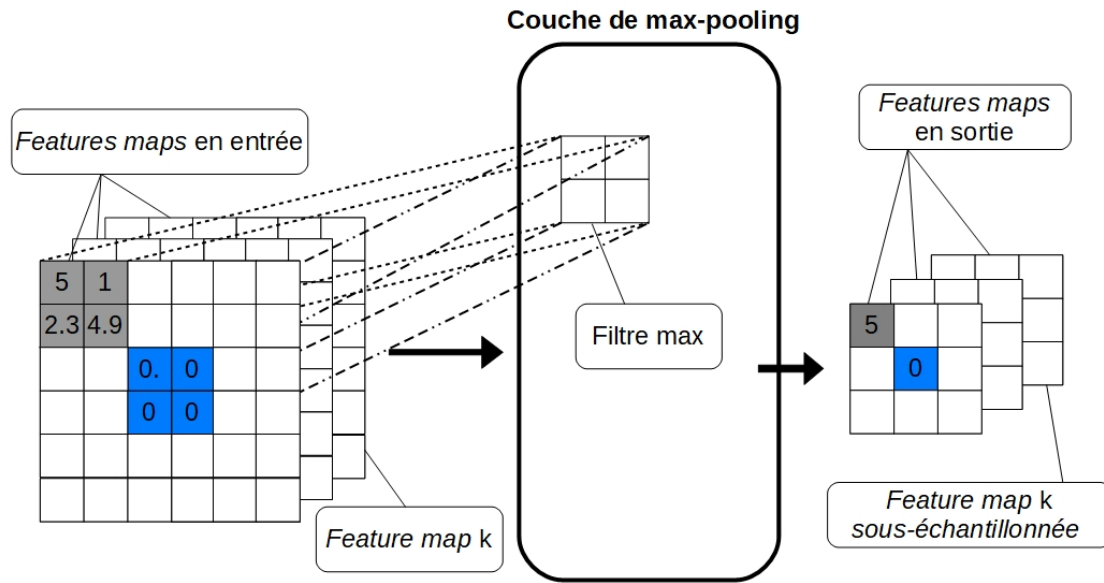


FIGURE 3.9 – Couche de *max-pooling*. Le *kernel* de taille 2×2 balaie l'image en entrée en s'appliquant qu'une fois à chaque pixel (les tuiles, c'est-à-dire les zones vues par le filtre de sous-échantillonnage, ne se chevauchent pas contrairement à certains cas en convolution, la *stride* est égal à la taille du *kernel*). Toutes les *features maps* en entrée sont filtrées par la couche de *max-pooling* et sont toutes de dimensions réduites en sortie.

3.5 Régression par CNN

La tâche de régression par CNN consiste à profiter de l'extraction automatique des *features* par les blocs de convolution situés dans les couches basses du réseau pour effectuer des estimations de paramètres. L'analyse des images d'entrée par les différentes couches des blocs de convolution permet la construction des *features map* synthétisant les informations utiles extraites par le réseau. Les *features maps* informatives sortant du dernier bloc de convolution sont ensuite traitées par les couches denses situées en couches hautes du réseau. C'est ici qu'est reconstruit la fonction permettant de relier le *label* de la donnée aux *features maps* représentant les images en entrée. Soit x les images en entrée de *label* y , les *features maps* créées à l'issue des blocs de convolution du CNN respectent l'équation suivante :

$$m = f_p^{(d_c)} \left(\underbrace{f_c^{(d_c, k)} (\dots f_c^{(1, 1)} (x))}_{f_e(x)} \right)$$

où

- m est un ensemble de *features maps*,
- $f_c^{(i, j)}$ la couche de convolution $i \in [1 ; d_{c, j}]$ du bloc de convolution $j \in [1 ; d_c]$ avec $d_{c, j}$ le nombre de couche de convolution dans un bloc et d_c le nombre de bloc de convolution dans le CNN,
- $f_p^{(j)}$ la couche de sous-échantillonnage du bloc j ,
- f_e la fonction d'extraction de *features* du réseau formée par la composition des différentes couches de convolution et de sous-échantillonnage.

La deuxième partie du CNN uniquement composé de couches denses entièrement connectées est alimentée par la fonction d'extraction de *features* f_e et fournit une prédiction découlant des *features maps* :

$$y^{(d)} = f^{(d)} \left(\underbrace{f^{(d-1)}(\dots f^{(1)}(m))}_{f_d(m)} \right)$$

où

- $y^{(d)}$ est la sortie de la dernière couche du réseau,
- $f^{(i)}$ est la couche dense $i \in \llbracket 1 ; d \rrbracket$ avec d le nombre de couches denses du CNN,
- f_d la fonction analyse des *features* formée par la composition des différentes couches denses du réseau

Le CNN est ainsi formé par la composition de la fonction d'extraction f_e et d'analyse f_d et permet de construire une fonction $f = f_d \circ f_e$ capable de calculer une prédiction à partir d'images d'entrée x . L'apprentissage du CNN consiste ainsi à résoudre le problème d'optimisation suivant :

$$\min_{\theta \in \Theta} \left(\text{loss}(y, f(x)) + \lambda \text{reg}(\theta) \right)$$

La tâche de régression par CNN a été moins étudiée que la tâche de classification par CNN. Cependant les travaux de recherche ont mis en avant l'amélioration des performances en régression avec l'utilisation de CNN. Dans [76], Toshev et al. comparent les performances de CNN disposés en cascade par rapport à des modèles graphiques définis sur arbre, des combinaisons de classificateurs linéaires, des forêts d'arbres décisionnels et un algorithme de *Neighbourhood Component Analysis* (NCA). Pour l'estimation des coordonnées spatiales d'un corps humain dans un espace 3D, le CNN permet d'obtenir de meilleures performances que les techniques citées. D'autres travaux se sont également intéressés à la problématique de la localisation d'un corps humain dans un espace 3D avec des CNN, concluant sur une amélioration des résultats par rapport aux autres algorithmes de régression existants [14] avec la mise en lumière des dépendances structurelles entre des points comme le reliement d'un avant-bras à un bras sur une image à partir d'images monoculaires [13]. Parmi les autres utilisations connues du CNN pour de la régression il y a l'estimation d'âge par analyse d'images dans [12] et l'estimation de profondeur dans une image à des fins d'utilisation pour des mesures d'holographies digitales dans [89]. Les CNN et plus particulièrement ceux possédant une couche de sortie linéaire (aussi appelés *vanilla deep regression*) montrent des performances compétitives avec des méthodes existantes dédiées aux problèmes d'estimation de coordonnées de visage sur images. L'avantage des *vanilla deep regression* étant que contrairement aux méthodes existantes, ceux-ci ne sont pas spécifiques à un problème donné [90].

Le CNN a aussi démontré une meilleure efficacité face à d'autres algorithmes de *Machine Learning* : comparé à un SVM dans le diagnostic d'électromyographie dans [91] ou le problème de netteté d'images floues par estimation du noyau *blur* [92]. Il a aussi été comparé face à un SVR dans le cadre de la modélisation spatiale des nappes phréatiques dans [7]. La régression par CNN montre également de meilleures performances comparée à d'autres algorithmes de *Deep Learning* (y compris des CNN) pour certaines tâches : dans [93], Chougule et al. transforment la détection de ligne sur route par une tâche de régression : l'estimation des coordonnées de la segmentation de la ligne à détecter. Le CNN développé arrive à reconstruire une représentation de la ligne à détecter plus précise que des CNN restreints à une détection simple par segmentation de l'image (LaneNet et U-Net).

3.6 Utilisation du *Machine Learning* dans le domaine GNSS

L'intérêt de la communauté du GNSS envers le *Machine Learning* est croissant depuis les années 2000 et a suscité de nombreuses recherches. Différents axes d'études ont été explorés afin de diminuer l'altération de la précision du positionnement apportée par les multitrajets, en milieu urbain principalement. L'état de l'art dressé par Jagiwala et al. dans [94] montre que des algorithmes de *Machine Learning* tels que des SVM et des CNN ont été appliqués à des données issues des différents étages du récepteur : au niveau de l'antenne *Radio-Frequency* (RF) mais également à l'étage du corrélateur en pré-corrélation ainsi qu'aux sorties des boucles de poursuite. Cette section consacrée aux algorithmes de *Machine Learning* appliqués à la dégradation par interférences de la précision de positionnement GNSS est divisée en trois parties. La première est consacrée aux données utilisées. Leur nombre, leur *labels* et leur type conditionnent les algorithmes qui leur seront appliqués. La deuxième partie concerne les *features* utilisées, nécessitant une extraction manuelle des données pour un certain nombre d'algorithmes implémentés ou étant automatique dans le cas d'algorithmes de *Deep Learning* (avec quelques exceptions qui seront étudiées). La dernière partie est une analyse concentrée sur deux algorithmes ayant montré les meilleures performances de détection d'interférences (dont majoritairement des multitrajets) qui sont les SVM et les CNN.

Remarque : Les multitrajets évoluent dans le temps. Cependant dans le cadre de cette thèse, nous avons conduit des travaux de recherche s'appuyant sur une vision purement statique du multitrajet dans une démarche de preuve de concept.

3.6.1 Données

Parmi les différents travaux effectués, les données analysées prennent différentes formes : une représentation de la fonction d'onde sous la forme d'images appelées *scalogram* [95], l'utilisation de la matrice d'autocorrélation [96] ainsi que d'images de sorties de corrélateur [19]. La forme la plus fréquente dans les travaux étudiés est celle de mesures brutes de signaux GNSS sous format *Receiver Independent Exchange Format* (RINEX).

Malgré la richesse du nombre de données utilisées, celles-ci peuvent être classées en deux catégories : les données issues de campagnes de collectes et les données synthétiques. Dans le premier cas, il s'agit d'utiliser un récepteur GNSS placé dans un environnement propice aux multitrajets (comme un centre urbain dense ou un milieu *indoor*) et d'enregistrer les signaux incidents pendant un laps de temps défini [22]. Dans [97] par exemple, les auteurs réalisent une collecte des signaux sur plusieurs bandes de fréquence (L1, L2, L5, E1, E5a et E5b) afin de multiplier les informations enregistrées au cours de l'expérience. Les campagnes de mesures ont été réalisées à partir de scénarios dits statiques (lorsque le récepteur reste immobile) et d'autres dits dynamiques (le récepteur est en mouvement pendant toute la durée de la collecte). Dans [98] notamment, Sun et al. comparent les performances résultantes sur des ensembles de données issues des réceptions statiques et dynamiques afin d'observer les différences de comportement de l'algorithme, avec l'exemple d'un apprentissage sur des *datasets* en scénarios statiques pour une généralisation sur des données de scénarios dynamiques.

La principale difficulté des données issues de collectes concerne leur *labellisation* pour le cas d'un apprentissage supervisé. Dans [99], un modèle physique de l'environnement proche en 3D est utilisé pour la *labellisation* des données collectées. D'autres, dans un cas de figure de scénarios statiques, passent par la simulation de signaux reçus par lancer de rayons et attribuent des *labels* en fonction de la visibilité ou du masquage des satellites [8].

Une seconde approche pour la construction des ensembles de données consiste à simuler les signaux en réception, comme dans [96] avec la simulation d'une réception sur plusieurs antennes. D'autres choisissent de directement générer les données sans passer par la reproduction du contexte physique des signaux comme dans [18]. Dans ces cas précis, la *labellisation* est générée en même temps que les données sont créées.

Type de données utilisées dans la thèse

Dans notre cas, nous travaillons avec le même type de données utilisées dans [18] : des images 2D générées en sortie de corrélateur à partir de simulations de signaux synthétiques. La *labellisation* des images s'effectue de manière automatique et le nombre d'images pouvant être créées par le générateur permet de construire des ensembles de données conséquents, de l'ordre de 10 000 images I et 10 000 images Q par *dataset*.

3.6.2 *Features* ou caractéristiques

En fonction des algorithmes utilisés le traitement des données d'entraînement/généralisation ne sera pas le même. En effet on peut différencier les algorithmes nécessitant une extraction manuelle des *features* et ceux effectuant l'extraction de manière automatique. Le premier cas concerne les algorithmes de *Machine Learning* comme les SVM, les arbres de décision, algorithmes de k plus proches voisins et de régression logistique. Ces algorithmes ne sont pas capables d'extraire les *features* importantes des données et nécessitent un pré-traitement externe des données afin de leur fournir les *features* sur lesquelles travailler. Différentes *features* ont été utilisées parmi lesquelles l'angle d'élévation du satellite émetteur [99], l'angle d'azimut [98], le temps de verrouillage de la poursuite [97] et la pseudo-distance [22]. Les travaux étudiés font parfois appel à des combinaisons de *features* comme la cohérence entre pseudo-distance (calculée à partir des mesures de pseudo-distances et de décalage Doppler entre deux instants [99]), le taux de variation du C/N_0 [8], les résidus de pseudo-distance [8] ou leur version normalisée [99]. Sur tous les travaux cités, la *feature* la plus fréquemment utilisée est le ratio entre le signal et le bruit ou SNR. Il permet notamment la comparaison des algorithmes de *Machine Learning* avec des méthodes de classification de LOS/NLOS plus classiques, comme la comparaison du SNR via un seuil fixé comme dans [100].

La difficulté liée à une extraction manuelle concerne la pertinence des *features* utilisées. En effet dans [100], Zhang et al. indiquent que certaines *features* comme les résidus de pseudo-distance normalisés apportent une instabilité au niveau des performances, affectant de manière positive ou négative en fonction de l'algorithme utilisé et du scénario de collecte des signaux étudié. Un autre problème apparaît au niveau de la multiplication des *features* différentes en entrée et concerne les différences d'ordres de grandeur des *features* : dans [101], les auteurs effectuent une normalisation du vecteur d'entrée afin de mettre les différentes *features* sur une même échelle (la différence pouvant atteindre un facteur 10^5).

Afin de contourner les difficultés liées aux extractions manuelles de *features*, des recherches ont eu recours à l'utilisation d'algorithmes de *Deep Learning* comme les CNN qui automatisent l'extraction. Connus pour leurs bonnes performances dans l'analyse d'images, ceux-ci ont été utilisés à diverses reprises dans la détection de multitrajet : sur des images 1D (vecteurs de paramètres) ou sur des images RGB représentant des *scalogram* [95]. Dans [96], un CNN est appliqué sur la matrice d'autocorrélation afin de régresser un vecteur de poids appliqué à un réseau d'antennes en vue d'un filtrage spatial des interférences. La réponse impulsionnelle des

3.6. Utilisation du *Machine Learning* dans le domaine GNSS

chaînes de réception, représentée sous la forme d'un vecteur de *features* est utilisée en tant qu'entrée d'un CNN [102]. Enfin dans [18], les différentes images utilisées par le CNN sont des sorties de corrélateur calculées à partir d'estimation du retard et du décalage en fréquence Doppler. Les CNN permettent donc de travailler sur des données de types différents dans un même objectif de réduction de l'impact d'interférences.

Features considérées dans la thèse

Dans notre cas les images construites sont issues des sorties de corrélateur [18]. De dimensions $N \times N$ avec $N \in \{10, 20, 40, 80\}$, les deux axes des images représentent les variations des paramètres de retard et de décalage en fréquence Doppler des signaux synthétiques générés. Dans un premier temps sans contexte physique, les signaux générés ont ensuite été construits à partir de simulations de propagation en milieu simulé dans le chapitre 6, permettant d'apporter plus de réalisme (au sens physique) aux données manipulées. La *labellisation* des images s'effectue de manière automatique lors de la génération des multitrajets.

3.6.3 Algorithmes de *Machine Learning*

3.6.3.1 *Support Vector Machine*

De nombreux algorithmes ont été testés sur les différents types de données possibles, quelles soient extraites au niveau des antennes, avant l'étape de corrélation ou en sortie du corrélateur. Parmi les algorithmes nécessitant une extraction manuelle des *features*, le SVM est celui qui obtient les meilleurs résultats. Dans [99], le SVM est appliqué à un vecteur de *features* composé du SNR, de l'angle d'élévation du satellite émetteur, de la cohérence de pseudo-distance et de la pseudo-distance normalisée. Comparé à une classification simple sur un critère de seuil sur le SNR, le SVM apporte une amélioration des performances. La moyenne de précision de classification sur les 8 différents ensembles de données de l'article est de 77.36% pour le SVM contre 71.77% pour la classification simple sur SNR, avec une erreur moyenne de positionnement en environnement urbain atteinte par l'algorithme de *Machine Learning* de 3.03 m. Dans [8] le SVM est utilisé pour distinguer les signaux LOS, NLOS et multitrajets (les auteurs font une distinction entre les cas où le signal direct est absent, cas NLOS, et la situation avec des signaux réfléchis (multitrajets) en présence du signal direct). Cette classification s'effectue sur un vecteur composé de quatre éléments : le C/N_0 , le taux de variation du C/N_0 , les résidus de pseudo-distances et le taux de variation des pseudo-distances. Dans [100], le SVM permet d'atteindre dans le cas des données considérées une précision de classification de 75%. Comparé à des algorithmes de k plus proches voisins, d'arbres de décision et d'un réseau de neurones dense (*fully-connected layer*) avec comme *features* le SNR, le nombre de satellites identifiés, l'angle d'élévation de ces satellites, les résidus de pseudo-distances et leur version normalisé, le SVM est l'algorithme de *Machine Learning* affichant les meilleures performances. Par rapport à une classification simple sur SNR, tous les algorithmes de *Machine Learning* testés permettent une réduction de l'erreur de classification de l'ordre de 60%. Dans [103], les *features* alimentant le SVM ne sont pas des mesures brutes issues des signaux GNSS mais extraites de la fonction d'autocorrélation. Les auteurs effectuent la comparaison des performances du SVM en variant les *features* données : l'utilisation des *features* issues de la fonction d'autocorrélation permet d'améliorer la précision de classification de 25% par rapport à des mesures en format *National Marine Electronics Association* (NMEA) ou RINEX.

3.6.3.2 Combinaison d'algorithmes de *Machine Learning*

Le cas étudié dans [98] n'est pas une utilisation individuelle de l'algorithme de SVM. En effet, la « machine » réalisée est composée de plusieurs étages d'algorithmes de *Machine Learning*. Dans le premier niveau, un SVM traite les *features* en entrée, vecteurs de dimension 9 contenant des informations telles que les angles d'élévation et d'azimut ainsi que les résidus et cohérence de pseudo-distances. Les résultats sont ensuite analysés par un *extreme gradient boosting* et la classification finale donnée par un algorithme de *logistic regression* (que l'on pourrait assimiler à une fonction seuil). Ce bloc d'algorithme de *Machine Learning* appelé *Stacking Ensemble Learning* (SEL) permet d'obtenir de meilleures performances de détection que les algorithmes de *Machine Learning* pris individuellement ainsi que les algorithmes de détection de SNR simple. A titre d'exemple, la précision moyenne de classification sur 10 ensembles de données différents est de 75.83% pour une classification simple sur SNR, 77.29% pour des *Random Forest*, 79.90% pour la méthode de classification par *Gradient boosting* sur arbre de décision, 82.11% pour les SVM et 84.92% pour la classification avec SEL.

3.6.3.3 CNN

Des algorithmes de *Deep Learning* ont également démontré de bonnes performances concernant la classification de multitrajets. Une attention notable a été apportée sur les CNN, particulièrement adaptée à l'analyse d'images. Dans [95], un CNN est appliqué sur des *scalogram* à des fins de détection de brouilleurs. Les auteurs ont remarqué que l'algorithme était non seulement capable de détecter les interférences mais également de les classer selon le type de brouillage utilisé (en onde continue, en ondes continues multiples, réponse impulsionnelles...).

Plusieurs architectures de CNN ont été testées parmi lesquelles le VGG. Les modèles ont été testés sur des ensembles de données, classés en fonction de la puissance des signaux délivrés par les brouilleurs. Les architectures AlexNet et GoogleNet offrent des précisions de classification respectivement de 72-91% et de 69-92%. Le réseau VGG-16 monte à une précision de 97% de détection pour des brouilleurs de *Jamming to Signal Ratio* (JSR) compris entre 9-13dB. Le réseau ResNet18 atteint un taux de détection de 97% pour un JSR de 19-23dB. L'algorithme MobileNetv2 développé par les auteurs est celui montrant les meilleures performances de détection pour tous les JSR testés avec un pourcentage de détection de 85% sur les brouilleurs les plus faibles testés et atteignant un taux de détection de 99.8% pour un JSR>9dB. Dans [96], le CNN est appliqué à la matrice d'autocorrélation des signaux reçus afin de retourner un vecteur de poids optimal pour la suppression des interférences par filtrage spatial. Les auteurs justifient le choix d'une architecture en CNN pour sa complexité computationnelle réduite ainsi que sa capacité d'extraction de *features* automatique et ne nécessitant pas d'information a priori comme la direction d'arrivée des signaux attendus. Le CNN étudié permet d'obtenir une valeur de *Signal to Interference plus Noise Ratio* (SINR) proche du SINR maximum obtainable par les poids optimum attribués aux antennes pour effectuer le filtrage spatial : la différence entre le SINR obtenu et le SINR optimal est inférieure à 0.5dB. Dans [17], le CNN analyse des images construites en fonction du retard et du temps (images 2D) en sortie de corrélateur et permet d'atteindre un taux de détection de multitrajet dépassant les 90%.

Dans [101], Liu et al. comparent les performances d'un CNN avec des SVM. Les images d'entrée sont construites à partir de vecteurs de *features* issues des boucles de poursuite (valeur de la corrélation, erreur sur la phase de la porteuse...). Comparé aux SVM, le CNN apporte une amélioration de la précision de détection atteignant les 45%. Dans [102], les images alimentant le CNN représentent les mesures brutes d'une réponse impulsionnelle sur Ultra Large Bande.

3.7. Conclusion

Comparé à d'autres architectures SVM et MLP, le CNN montre une meilleure précision dans le nombre de classifications correctes, le nombre de NLOS correctement classés ainsi qu'au niveau du ratio de NLOS détectés sur le nombre de NLOS détectés (vrais positifs) ajouté au nombre de NLOS détectés comme LOS (faux négatifs). Dans [18], les performances entre un CNN et un SVM sont comparés : le CNN est appliqué sur des images de sorties de corrélateur construites en fonction du retard et de la fréquence Doppler. Le SVM est alimenté par des *features* composées du nombre de maximums locaux aux sorties du corrélateur ainsi que de la distribution du retard à la sortie de corrélateur de valeur maximale. Le CNN obtient de meilleures performances de classification dans la majorité des données testées, notamment pour une simulation de réception de signal avec un temps d'intégration égal à 1ms.

CNN pour l'estimation

Travaillant sur des données sous forme d'images de sorties de corrélateur à des fins d'estimation de paramètres, nous avons convergé vers une utilisation d'un algorithme CNN. Le CNN est plus performant que le SVM dans la majorité des cas de détections effectuées dans [18]. De plus, les travaux effectués sur d'autres types de données démontrent une supériorité du CNN dans la tâche de classification d'interférences, multitrajets, signaux NLOS voire de brouilleurs. Nous considérons aussi l'extraction automatique de *features* des algorithmes de *Deep Learning*, notamment du CNN sur des données sous forme d'images, comme un atout majeur pour la tâche d'estimation des paramètres du multitrajet à partir des sorties de corrélateur sur les canaux I et Q.

3.7 Conclusion

La problématique de la thèse porte sur l'estimation des quatre paramètres décrivant entièrement un multitrajet à partir de sorties de corrélateur E-P-L ainsi que des sorties supplémentaires calculées pour d'autres valeurs de retard et de fréquence Doppler. Parmi les méthodes d'estimation de paramètres explorées dans ce chapitre, notre attention s'est portée sur l'utilisation des réseaux de neurones à convolution à des fins de régression. Comme tout réseau de neurones, l'extraction de *features* à partir des données brutes est automatique. Comparé à un réseau de neurones de structure peu complexe comme le MLP, un CNN possède moins de poids neuronaux à régler par *weight sharing*. Ses blocs de convolution lui permettent d'être robuste à la déformation et au décalage au sein des images [78]. En plus d'avoir démontré des capacités d'analyses d'images avec des prédictions plus performantes que d'autres algorithmes ML [91, 7], le CNN a aussi été mis en avant pour sa capacité de détection de multitrajets sur images et plus particulièrement sur des images de sorties de corrélateur dans [18, 19, 21]. Pour toutes ces raisons, nous avons choisi d'utiliser une structure en réseau de neurones à convolution pour estimer le quadruplet de paramètres du multitrajet. Le chapitre suivant décrit l'architecture CNN élaborée et détaille les résultats des expériences de régression sur les ensembles de données introduits au chapitre 2.

Chapitre 4

Estimation de paramètres de multitrajets sur des images de sorties de corrélateur

Sommaire

4.1 Architecture en VGG	62
4.2 Puissance de calcul mise en oeuvre	64
4.3 Protocole expérimental	64
4.4 Calcul MAE moyennée	65
4.4.1 Résultats selon les différentes amplitudes	65
4.4.2 Résultats selon les différents niveaux de C/N_0	69
4.4.3 Discussion	73
4.5 Conclusion	73

Ce chapitre est consacré à l'estimation des paramètres des multitrajets à partir des images produites par le générateur de sorties de corrélateur synthétiques, dont la production a été détaillée au chapitre 2. Comme introduit dans le chapitre 3, l'utilisation du réseau de neurones est justifiée par de précédentes recherches effectuées sur la détection de multitrajets, notamment à partir d'images de sorties de corrélateur. Nous nous proposons de prendre la suite de ces travaux de recherche à des fins d'estimation de paramètres de multitrajets en vue d'une reconstruction de l'interférence pour la suppression de sa composante aux sorties du corrélateur. Ce changement de tâche de détection à celle d'estimation est justifié par les performances satisfaisantes de détection de multitrajets issus du générateur de sorties de corrélateur par l'utilisation d'algorithmes de *Machine Learning*. Par rapport aux articles [21], [18] et [19] où les *features* du CNN étaient construites à des fins de détection, dans notre cas d'étude les *features* extraites par le CNN serviront à la prédiction d'une valeur continue à partir d'un tenseur de dimension $N \times N \times 2$, avec $N \times N$ la résolution des images produites par le générateur. Pour la recherche de ces nouvelles *features*, nous avons surdimensionné le réseau en utilisant davantage de couches de convolution par rapport aux structures utilisées pour la détection des multitrajets.

Dans un premier temps nous présenterons l'architecture du réseau utilisé. Dans un second temps nous analyserons les performances de l'algorithme suite aux expériences effectuées par ce réseau sur les ensembles de données définis section 2.7. Les données varieront selon trois facteurs. Le premier sera l'amplitude des multitrajets à estimer : cela permettra d'observer l'amplitude minimale d'un multitrajet pouvant être analysé par le CNN. Le deuxième facteur

VGG

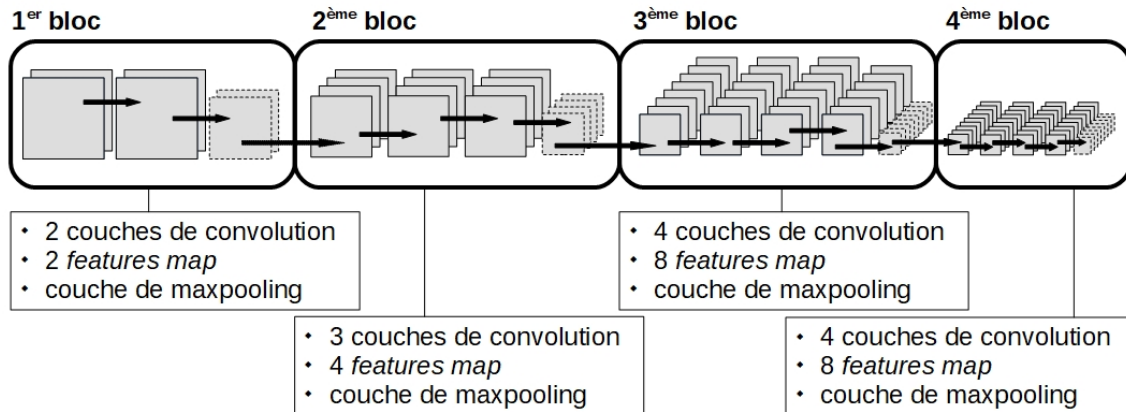


FIGURE 4.1 – Principe des blocs au sein d’un réseau CNN de type VGG. Un bloc est formé par une succession de couches de convolution suivie d’une couche de *pooling* (dans notre cas de *max-pooling*). Au fur et à mesure des blocs, la dimension des *features map* est diminuée tandis que leur nombre est démultiplié. Au-delà d’une certaine profondeur, le nombre de *features map* n’augmente plus d’un bloc au suivant mais en revanche la dimension de celles-ci continue d’être réduite.

concernera le niveau de C/N_0 , qui évoluera de conditions de réception favorables à détériorées. Enfin le dernier de ces facteurs concernera la taille des images, en d’autres termes la résolution de celles-ci, permettant de mesurer l’impact du nombre de sorties de corrélateur utilisées sur les performances d’estimation du CNN.

4.1 Architecture en VGG

La structure du CNN utilisée pour les différentes tâches de régression (et schématisée figure 4.2) s’inspire des réseaux VGG. Ces réseaux se sont illustrés par leurs performances d’analyse de quantité importante d’images lors du challenge *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) en 2014 par Simonyan et Zisserman [20]. Ce sont des réseaux de neurones à convolution caractérisés par des blocs de convolution successifs de noyaux dont les dimensions excèdent rarement les 3×3 . Comme explicité dans le schéma 4.1, le principe est de travailler avec davantage de *features map* au fur et à mesure que l’on progresse en profondeur dans le réseau. Cette augmentation d’images analysées s’accompagne d’une réduction de leur dimension : cela permet de synthétiser les *features* extraites mais également d’accélérer le traitement des images par manipulation de données plus petites [82]. Une fois tous les blocs traversés, les *features map* sont mises sous la forme d’un vecteur par la couche *Flatten* sans aucune autre modification. Les données sous forme 1D sont ensuite traitées par une succession de couches denses. Des expériences préliminaires ont montré une amélioration des performances de l’algorithme en fixant le nombre de neurones des couches denses plutôt qu’en adaptant leur nombre à la taille des images (et donc des *features map*) traitées par le réseau. En comparant

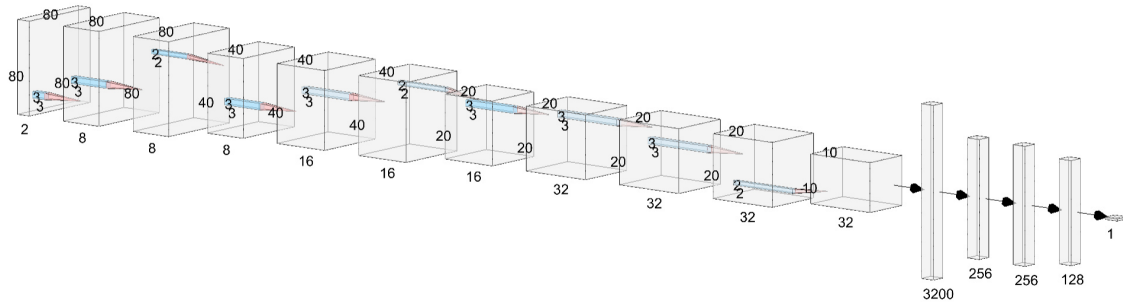


FIGURE 4.2 – Architecture du CNN dans le cadre d’une estimation de paramètres à partir d’images I & Q de 80×80 pixels. Des blocs de convolution de différentes tailles et dimensions composent les couches les plus basses du réseau. Trois couches denses de 256, 256 et 128 neurones forment les couches les plus hautes du réseau et alimentent une couche de sortie linéaire formée par un seul neurone. Les parties convolutive et dense du réseau sont reliées par une couche de *Flatten*, vecteur de longueur 3200 dans ce cas précis. Image générée avec l’outil NN-SVG [104].

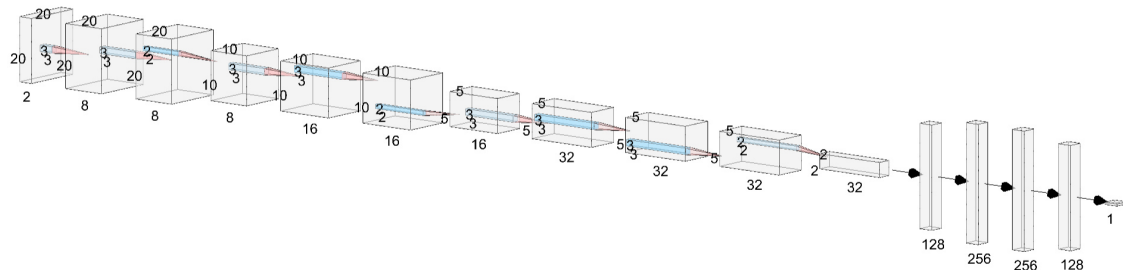


FIGURE 4.3 – Architecture du CNN dans le cadre d’une estimation de paramètres à partir d’images I & Q de 20×20 pixels. Seuls les blocs de convolution et la longueur de la couche de *Flatten* ont été adaptés aux dimensions des images en entrée. La partie dense du réseau reste identique avec trois couches cachées de 256, 256 et 128 neurones. Image générée avec l’outil NN-SVG [104].

le réseau utilisé pour des images 80×80 (figure 4.2) et celui appliqué à des images 20×20 (figure 4.3), on remarque que l’architecture est similaire : le nombre de couches est exactement le même, la seule différence se situe au niveau de la dimension des images et des *feature map* transitant entre les blocs de convolution du réseau. La sortie de ce réseau est restreinte à un unique neurone car le CNN effectue l’estimation d’un seul paramètre à la fois. Le nombre de sorties sera démultiplié dans les architectures du réseau aux chapitres 5, 6 et 7 mais toujours dans l’optique de l’estimation d’un seul paramètre à la fois.

Remarque : Les VGG ne font pas partie des architectures de réseau de neurones les plus récentes. D’architecture relativement simple, ils se sont illustrés par leur performance sur l’analyse d’images [90]. Nous avons choisi d’utiliser une architecture en VGG pour sa simplicité d’implémentation et de par la reconnaissance de ses performances d’analyses d’images, notamment dans des tâches d’estimation par régression [14]. L’architecture VGG utilisée dans les différentes expériences menées lors cette thèse n’a pas été optimisée car l’objectif de la thèse n’était pas une optimisation mais s’inscrivait dans une démarche de preuve de concept : on a cherché à démontrer qu’il était possible d’estimer des paramètres de multitrajets à partir d’images de sorties de corrélateur.

4.2 Puissance de calcul mise en oeuvre

Les expériences mettent en oeuvre un CNN dans le cadre d'un apprentissage *off-line*. La puissance de calcul requise est majoritairement sollicitée lors de la phase d'apprentissage, notamment pour les calculs de rétropropagation du gradient. Le temps de calcul lors de l'inférence, phase pendant laquelle les performances de l'algorithme sont évaluées avec des données du *dataset* de test/généralisation, est négligeable par rapport au temps de calcul correspondant à la phase d'apprentissage. À titre d'exemple, sur des images de résolution de 80×80 pixels, les temps de calcul consacrés à l'apprentissage oscillent entre 13-36 minutes. Sur des images de résolution 10×10, les temps d'apprentissage sont plus bas et compris entre 4 et 11 minutes. Les temps de calcul de la phase d'inférence sont de l'ordre de quelques secondes pour toutes les tailles d'images testées.

La puissance de calcul demandée au cours des expériences est importante en raison de la quantité de données utilisées lors de l'apprentissage, montant à 10 000 images, de résolution pouvant varier de 10×10×2 à 80×80×2 pixels. La taille du réseau, composée de 14 couches, constitue également un facteur non négligeable concernant l'espace mémoire occupé : l'entraînement d'un tel algorithme sollicite un espace mémoire dépassant les 10 GiB. Pour accélérer cette phase, nous utilisons un GPU modèle GeForce RTX 2080 Ti des serveurs du laboratoire ENAC, d'une capacité mémoire maximale de 11 264 MiB. L'utilisation d'un serveur de calcul a permis de générer des résultats avec des temps de calculs non contraignants. Le temps de calcul du réseau n'a pas été optimisé car ce n'était pas l'objectif principal (preuve d'estimation par régression sur des images de sorties de corrélateur) et les temps de calcul atteints sur le serveur étaient déjà satisfaisants.

D'un point de vue utilisateur, la structure du réseau pèse 3.7MiB. Le multitrajet étant défini par les quatre paramètres de retard, fréquence Doppler, amplitude et phase, la mémoire occupée par les quatre CNN dédiés à l'estimation des différents paramètres est de l'ordre de 14.8MiB. Dans [19], Munin et al. embarquent un CNN dédié à la détection de multitrajet sur un VPU, processeur adapté pour les calculs neuronaux. D'une capacité de 500MiB, l'utilisation d'une structure en CNN permet d'obtenir des réseaux de neurones occupant un espace mémoire moindre par rapport à un réseau de neurones uniquement composé de couches denses et autorise l'embarquement de tels algorithmes dans des VPU.

4.3 Protocole expérimental

Les expériences se sont déroulées selon deux phases successives : la première pendant laquelle les différents CNN sont entraînés à l'estimation des paramètres de multitrajet à partir des images de sorties de corrélateur et la deuxième étant l'étape de généralisation des capacités d'estimation des CNN sur des données inédites. L'étape d'apprentissage incluant des aspects stochastiques comme l'initialisation aléatoire des poids du réseau, les expériences vont être répétées à dix reprises afin de calculer les performances moyennes du réseau. Parmi les aspects stochastiques que nous devons prendre en compte nous avons :

- Mélange aléatoire des données : toutes issues du même générateur d'images, elles sont séparées en trois groupes disjoints : l'ensemble d'entraînement, de validation et de test. L'ensemble d'entraînement est créé à partir de 80% de l'effectif total, les ensembles de validation et de généralisation se répartissent équitablement les 20% de données restantes (soit 10% de l'effectif total chacun),
- Optimisation stochastique pour la minimisation du *loss* : calcul de l'erreur moyenne du

réseau sur un *batch* de données. Les données composant le *batch* sont remplacées par d'autres entre chaque *epoch* afin de ne pas effectuer le calcul sur les mêmes données à chaque itération. La taille du *batch* est fixée à 1000 échantillons sachant qu'un ensemble d'apprentissage est composé de 8 000 échantillons,

- Initialisation aléatoire des poids du réseau : le point d'initialisation est un élément clé en tant que problème d'optimisation : il est donc important de le varier afin d'avoir des pertes d'évaluation représentatives. Dans le cadre de la thèse, nous utilisons l'initialisation de He, adaptée pour des fonctions d'activation ReLU [105].

4.4 Calcul MAE moyennée

Afin de comparer les performances entre les différentes séries d'expériences, les résultats de dix *runs* sont résumés en une seule valeur, la avg_{MAE} , qui est la moyenne des *Mean Absolute Error* (MAE) sur chacun des *runs*. Un *run* est une expérience au cours de laquelle un algorithme de *Machine Learning* effectue un apprentissage à partir d'un ensemble de données d'apprentissage avec des poids et biais initiaux. Afin d'observer la stabilité du réseau et pour tenir compte des processus aléatoires présents à la fois dans la construction des ensembles de données, de l'initialisation des paramètres du réseau et des calculs de descente de gradient, plusieurs *runs* d'expériences sont effectués avec des ensembles de données différents et de nouveaux poids et biais initiaux. Ainsi pour un nombre de *runs* $N_r \in \mathbb{N}^*$, le réseau va exécuter N_r phase d'entraînement, chaque entraînement se déroulant sur un ensemble d'apprentissage et des poids et biais différents des autres *runs*. Pour les expériences de régression par CNN, la avg_{MAE} est calculée de la manière suivante :

$$avg_{MAE} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left[\frac{1}{D_i} \sum_{j=1}^{D_i} \left[N_{\theta}(x_j^i) - y_j^i \right] \right] \quad (4.1)$$

avec

- N_{θ} le modèle CNN de paramètre $\theta \in \Theta$, Θ un espace paramétrique,
- $N_r = 10$,
- D_i le nombre d'éléments de l'ensemble de données utilisé à la $i^{\text{ème}}$ expérience ($\forall i \in [1, N_r]$ $D_i = 10000$ car le nombre de données à chaque expérience est le même),
- x_j^i la $j^{\text{ème}}$ donnée lors du $i^{\text{ème}}$ *run*,
- y_j^i le *label* correspondant à la donnée x_j^i .

4.4.1 Résultats selon les différentes amplitudes

Préliminaires

Deux axes d'analyse sont appliqués aux expériences de cette sous-section. Le premier axe d'étude se concentre au niveau de l'influence de la dimension des images d'entrée sur les performances. Les images étant construites à partir des sorties de corrélateur, cela permet d'observer l'évolution des performances avec une diminution du nombre de sorties utilisées (les expériences étant effectuées sur des images 80×80, 40×40, 20×20 et 10×10 pixels), c'est-à-dire avec une donnée d'entrée contenant moins d'informations (on divise par 4 le nombre de pixels par image à chaque fois : une image en 10×10 contient donc 64 fois moins de pixels qu'une

4.4. Calcul MAE moyennée

image de 80×80). Le deuxième axe d'analyse de cette sous-section d'expériences s'intéresse à l'impact de la puissance des multitrajets sur les capacités d'estimation des paramètres. Étant donné que l'amplitude des multitrajets fait partie des paramètres recherchés, ce deuxième axe de recherche peut ainsi donner un ordre d'idée de la précision moyenne du réseau en fonction de l'estimation de la puissance du multitrajet.

Les tables 4.1, 4.2, 4.3 et 4.4 renseignent les résultats sur les estimations de paramètres par l'algorithme de CNN utilisé en régression.

Observations

La table 4.1 expose les performances du CNN en fonction des deux axes d'analyse. La première constatation est que l'erreur d'estimation de $\Delta\tau_{MP}$ est plus élevée pour un multitrajet d'amplitude plus faible : pour une même résolution d'images (par exemple en 80×80), l'erreur d'estimation moyenne est de $2.50 \cdot 10^{-2} T_c$ avec des multitrajets d'amplitude supérieure à $0.6A_c$ (avec A_c l'amplitude du signal LOS) contre $3.03 \cdot 10^{-2} T_c$ pour des multitrajets modérés (d'amplitude entre $0.4A_c$ et $0.6A_c$) et montant à $4.64 \cdot 10^{-2} T_c$ pour des multitrajets d'amplitude inférieure à $0.4A_c$, soit une augmentation de 86% entre des multitrajets forts et des multitrajets faibles. Dans les mêmes conditions, l'écart-type s'accroît et passe de $0.22 \cdot 10^{-2} T_c$ à $1.44 \cdot 10^{-2} T_c$, soit une multiplication de l'écart-type par six : la moyenne est moins représentative avec l'estimation de multitrajets plus faibles. Les erreurs les plus basses sont enregistrées pour les multitrajets les plus forts tandis que les erreurs les plus élevées sont liées aux multitrajets les plus faibles. Lorsque le *dataset* ne contient pas de multitrajets faibles (donc pour les *datasets* Strong MP et Moderated MP), l'erreur d'estimation moyenne ne dépasse pas les 10% T_c .

Le deuxième facteur remarquable de cette augmentation de l'erreur est la dimension des images d'entrée (donc le nombre de sorties de corrélateur utilisées). Une diminution de ce nombre se traduit par une dégradation des performances de l'algorithme. Les performances d'estimation deviennent similaires entre multitrajets forts et modérés sur des images 20×20 (respectivement $avg_{MAE} = 4.30 \cdot 10^{-2} T_c$ et $avg_{MAE} = 4.44 \cdot 10^{-2} T_c$). Les performances observées du CNN liées à l'ensemble de données Random MP se situent entre les résultats obtenus sur les multitrajets modérés et ceux rapportés par les faibles multitrajets. Une augmentation de l'erreur plus prononcée est également constatée lorsque les images sont réduites de 20×20 à 10×10 pixels qu'entre 80×80 - 40×40 , 40×40 - 20×20 ou même entre 80×80 - 20×20 pixels. De manière gé-

nérale, une nette dégradation des performances est observée lorsque l'algorithme est alimenté par des images de dimension 10x10.

TABLE 4.1 – Erreur d'estimation du paramètre $\Delta\tau_{MP}$ par le CNN-Reg selon les différentes amplitudes des multitrajets (unité de table : 10^{-2} Tc).

Ensemble de données	Dimensions des images	$avg_{MAE} \Delta\tau_{MP}$
Strong MP	80×80	2.50 ± 0.22
	40×40	2.91 ± 0.32
	20×20	4.30 ± 0.42
	10×10	6.70 ± 0.40
Moderated MP	80×80	3.03 ± 0.54
	40×40	3.44 ± 0.65
	20×20	4.44 ± 0.45
	10×10	9.90 ± 0.72
Weak MP	80×80	4.64 ± 1.44
	40×40	4.33 ± 0.42
	20×20	7.20 ± 1.55
	10×10	26.13 ± 1.96
Random MP	80×80	3.82 ± 0.68
	40×40	4.92 ± 0.94
	20×20	7.00 ± 0.78
	10×10	16.34 ± 1.19

Dans la table 4.2, on observe un comportement similaire pour l'estimation de Δf_D . Les performances du CNN sont meilleures lorsque le nombre de sorties de corrélateur utilisées pour l'estimation est élevé et lorsque les multitrajets sont de plus grandes intensités. Concernant les multitrajets forts ou modérés, les performances sont quasi équivalentes entre des images en 80×80 et des 40×40 pixels : en prenant les résultats obtenus avec le *dataset* Strong MP, l'erreur moyenne est de 1.22 Hz pour des images 80×80 et égale à 1.33 Hz en 40×40. On constate également un net accroissement de l'erreur lors du passage d'images en 20×20 à 10×10 : par exemple

4.4. Calcul MAE moyennée

sur des multitrajets de toutes amplitudes (Random MP) l'erreur croît de 7.00 Hz à 16.34 Hz, soit une multiplication de l'erreur moyenne de 2.3.

TABLE 4.2 – Erreur d'estimation du paramètre Δf_D par le CNN-Reg selon les différentes amplitudes des multitrajets (unité de la table : Hz).

Ensemble de données	Dimensions des images	$avg_{MAE} \Delta f_D$
Strong MP	80×80	1.22 ± 0.20
	40×40	1.60 ± 0.27
	20×20	2.36 ± 0.41
	10×10	5.16 ± 0.26
Moderated MP	80×80	1.33 ± 0.29
	40×40	1.70 ± 0.23
	20×20	2.61 ± 0.45
	10×10	7.08 ± 0.36
Weak MP	80×80	1.75 ± 0.31
	40×40	2.41 ± 0.38
	20×20	4.30 ± 0.41
	10×10	21.48 ± .68
Random MP	80×80	1.62 ± 0.34
	40×40	2.06 ± 0.36
	20×20	3.40 ± 0.34
	10×10	12.40 ± 0.76

Comme pour les paramètres précédents, les résultats présents dans la table 4.3 montrent une augmentation de l'erreur d'estimation avec la dégradation des images. De bonnes performances d'estimation sont observées sur les faibles multitrajets, avec une erreur ne dépassant pas les 4% sur des images 80×80. Le passage de 20×20 à 10×10 pixels est à nouveau déclencheur d'une hausse conséquente des erreurs d'estimation du réseau. Malgré cette dégradation, l'erreur d'estimation de l'amplitude reste inférieure à 9% de l'amplitude du signal LOS.

TABLE 4.3 – Erreur d'estimation du paramètre $\Delta \alpha_{MP}$ par le CNN-Reg selon les différentes amplitudes des multitrajets (unité de la table : % de l'amplitude de LOS).

Ensemble de données	Dimensions des images	$avg_{MAE} \Delta \alpha_{MP}$
Strong MP	80×80	3.95 ± 0.93
	40×40	4.45 ± 0.61
	20×20	5.60 ± 0.70
	10×10	7.32 ± 0.40
Moderated MP	80×80	3.71 ± 0.64
	40×40	3.76 ± 0.53
	20×20	4.79 ± 0.55
	10×10	5.97 ± 0.16
Weak MP	80×80	3.53 ± 0.88
	40×40	4.06 ± 1.28
	20×20	4.09 ± 1.10
	10×10	7.87 ± 0.33
Random MP	80×80	3.48 ± 0.98
	40×40	3.78 ± 0.60
	20×20	4.97 ± 0.61
	10×10	8.28 ± 0.24

La table 4.4 renseigne les erreurs d'estimation de la phase du multitrajets sur le jeu de données d'amplitudes de multitrajets différentes pour un C/N_0 fixe. Les performances du CNN sont minorées par une erreur de 12.3° , résultat atteint avec des multitrajets de fortes intensités sur des images 80×80 . Comme pour les autres paramètres, l'erreur d'estimation augmente lorsque les multitrajets diminuent en intensité et lorsque le nombre de sorties de corrélateur est plus faible. Une forte variabilité est également observée dans l'estimation de la phase des faibles multitrajets à 80×80 pixels, avec un écart-type de $\pm 6^\circ$. On note une forte dégradation des performances lors de la diminution de la résolution des images de 20×20 à 10×10 pixels pour tous les *datasets* de $C/N_0 = 40$ dBHz, avec une erreur d'estimation dépassant les 50° pour les multitrajets des ensembles Weak MP et Random MP. Sur le *dataset* contenant toutes les amplitudes de multitrajets (Random MP), l'erreur d'estimation moyenne dépasse les 20° en-dessous d'une résolution d'images de 80×80 pixels.

TABLE 4.4 – Erreur d'estimation du paramètre $\Delta\phi_{MP}$ par le CNN-Reg selon les différentes amplitudes des multitrajets (unité de la table : degrés).

Ensemble de données	Dimensions des images	$avg_{MAE} \Delta\phi_{MP}$
Strong MP	80×80	12.30 ± 1.46
	40×40	15.41 ± 3.46
	20×20	15.59 ± 2.44
	10×10	29.68 ± 2.06
Moderated MP	80×80	15.71 ± 3.07
	40×40	15.72 ± 1.74
	20×20	17.90 ± 1.99
	10×10	42.55 ± 2.30
Weak MP	80×80	26.45 ± 6.25
	40×40	30.06 ± 3.35
	20×20	29.57 ± 3.17
	10×10	70.23 ± 1.56
Random MP	80×80	19.90 ± 3.92
	40×40	21.90 ± 3.51
	20×20	22.73 ± 3.15
	10×10	50.63 ± 2.56

Synthèse des observations

Les performances du CNN se dégradent avec la diminution de la résolution des images ainsi qu'avec l'atténuation de l'amplitude des multitrajets. On remarque que la dégradation des performances est beaucoup plus abrupte entre des images 20×20 et 10×10 pour tous les paramètres testés.

4.4.2 Résultats selon les différents niveaux de C/N_0

Préliminaires Dans cette sous-section d'expériences, l'axe d'analyse selon les amplitudes de multitrajets est substitué par l'analyse des performances selon la valeur de C/N_0 . L'objectif est de mesurer l'évolution des performances dans des conditions où le multitrajets et le LOS sont plus difficilement distinguables avec un bruit s'intensifiant, afin d'observer si la tâche d'estimation par régression sur les paramètres de multitrajets devient trop compliquée avec les différents C/N_0 testés. Le niveau de C/N_0 est dégradé selon une grille de quelques valeurs (43 dBHz,

4.4. Calcul MAE moyennée

40 dBHz, 37 dBHz et 34 dBHz) simulant des conditions de réception très bonnes (C/N_0 le plus élevé) à des conditions de réception très mauvaises (C/N_0 le plus faible C/N_0).

Les tables 4.5, 4.6, 4.7 et 4.8 donnent ainsi les erreurs d'estimation des paramètres en fonction des niveaux de C/N_0 et de la résolution des images.

Observations

Dans la table 4.5, on constate que les performances d'estimation du CNN se dégradent avec la diminution du ratio signal/bruit ainsi que du nombre de sorties de corrélateur utilisées. Les performances sont équivalentes pour un $C/N_0 \geq 40$ dBHz avec des images 80×80 ($3.33 \cdot 10^{-2} T_c$ et $3.82 \cdot 10^{-2} T_c$ pour un C/N_0 de 43 dBHz et de 40 dBHz respectivement). On observe le même phénomène sur des images en 40×40 . Pour des images en 20×20 , la similarité entre performances d'estimation s'observe pour des valeurs de C/N_0 de 37 dBHz et 40 dBHz. Les expériences sur les données de $C/N_0 = 34$ dBHz, en plus d'être celles donnant les erreurs les plus importantes, sont aussi celles d'écart-types les plus prononcés. Une dernière remarque peut être effectuée sur les images en 10×10 , où comme pour le jeu de données de la section 4.4.1 l'erreur augmente radicalement et ne partage plus le même ordre de grandeur des images de plus grandes dimensions. Sur ces dimensions, l'erreur d'estimation moyenne est au-dessus de 10% T_c quel que soit le C/N_0 testé tandis que l'erreur d'estimation moyenne ne dépasse pas les 9% T_c sur les images de dimensions supérieures pour tous les C/N_0 expérimentés.

TABLE 4.5 – Erreur d'estimation du paramètre $\Delta\tau_{MP}$ pour différents niveaux de C/N_0 (unité : $10^{-2} T_c$).

Niveau de C/N_0	Dimension image	$avg_{MAE} \Delta\tau_{MP}$
$C/N_0 = 43$ dBHz	80×80	3.33 ± 0.63
	40×40	4.28 ± 0.60
	20×20	5.31 ± 0.70
	10×10	11.63 ± 0.59
$C/N_0 = 40$ dBHz	80×80	3.82 ± 0.68
	40×40	4.92 ± 0.94
	20×20	7.00 ± 0.78
	10×10	16.34 ± 1.19
$C/N_0 = 37$ dBHz	80×80	5.83 ± 1.52
	40×40	5.13 ± 0.57
	20×20	7.88 ± 1.30
	10×10	22.88 ± 0.84
$C/N_0 = 34$ dBHz	80×80	5.75 ± 1.55
	40×40	6.73 ± 1.88
	20×20	8.86 ± 2.25
	10×10	30.11 ± 1.69

Dans la table 4.6, les résultats sont de même ordre de grandeur entre les différents niveaux de C/N_0 pour une même taille d'images. En 80×80 , l'erreur est comprise entre 1.61 Hz et 1.74 Hz. Pour les images en 40×40 , les performances sont comprises entre 2.05 Hz et 2.45 Hz. De même en 20×20 l'erreur reste bornée entre 3.16 Hz et 3.74 Hz. Lorsque les images sont réduites aux dimensions 10×10 , la précision d'estimation de Δf_D se dégrade fortement : l'erreur est multipliée par trois dans de bonnes conditions de réception (erreur de 3.16 Hz en 20×20 contre 9.08 Hz en 10×10) alors qu'elle est presque multipliée par sept dans un cadre de réception fortement

altéré (erreur de 3.81 Hz en 20×20 contre 22.57 Hz en 10×10). Les performances en 80×80 et 40×40 sont presque exactement les mêmes pour un $C/N_0 \geq 40$ dBHz.

TABLE 4.6 – Erreur d'estimation du paramètre Δf_D pour différents niveaux de C/N_0 (unité : Hz).

Niveau de C/N_0	Dimension image	$avg_{MAE} \Delta f_D$
$C/N_0 = 43$ dBHz	80×80	1.61 ± 0.11
	40×40	2.05 ± 0.28
	20×20	3.16 ± 0.18
	10×10	9.08 ± 0.40
$C/N_0 = 40$ dBHz	80×80	1.62 ± 0.34
	40×40	2.06 ± 0.36
	20×20	3.40 ± 0.34
	10×10	12.40 ± 0.76
$C/N_0 = 37$ dBHz	80×80	1.65 ± 0.24
	40×40	2.27 ± 0.30
	20×20	3.74 ± 0.60
	10×10	16.54 ± 1.05
$C/N_0 = 34$ dBHz	80×80	1.74 ± 0.17
	40×40	2.45 ± 0.38
	20×20	3.81 ± 0.56
	10×10	22.57 ± 0.86

Les résultats de la table 4.7 confirment les observations faites dans les deux tables précédentes de cette section. Les performances sont assez proches pour un $C/N_0 \geq 40$ dBHz pour des images avec un minimum de 20×20 pixels. L'augmentation de l'erreur est plus prononcée lorsque le nombre de pixels par image passe en dessous de 20×20 pixels (les performances se dégradant progressivement avec la diminution du nombre de pixels au-dessus de cette limite). Par exemple pour un $C/N_0 = 40$ dBHz l'erreur augmente de 6.07% à 11.96% soit une multiplication

4.4. Calcul MAE moyennée

de l'erreur moyenne d'estimation de l'amplitude du multitrajet par 1.5 en utilisant des images 10×10 au lieu de 20×20.

TABLE 4.7 – Erreur d'estimation du paramètre $\Delta\alpha_{MP}$ pour différents niveaux de C/N_0 (unité : % de l'amplitude de LOS).

Niveau de C/N_0	Dimension image	$avg_{MAE} \Delta\alpha_{MP}$
$C/N_0 = 43$ dBHz	80×80	3.03 ± 0.71
	40×40	4.19 ± 1.50
	20×20	4.42 ± 0.45
	10×10	6.28 ± 0.21
$C/N_0 = 40$ dBHz	80×80	3.48 ± 0.98
	40×40	3.78 ± 0.60
	20×20	4.97 ± 0.61
	10×10	8.28 ± 0.24
$C/N_0 = 37$ dBHz	80×80	4.04 ± 0.96
	40×40	4.13 ± 0.97
	20×20	6.07 ± 1.18
	10×10	11.96 ± 0.64
$C/N_0 = 34$ dBHz	80×80	5.86 ± 1.05
	40×40	5.69 ± 1.76
	20×20	7.00 ± 1.43
	10×10	17.69 ± 1.22

Les erreurs d'estimation de phase affichées table 4.8 sont, dans les meilleures conditions, supérieures à $17,69^\circ$. Pour un C/N_0 fixé à $= 43$ dBHz, les performances du réseau observées sont stables en travaillant sur 40×40 ou 20×20 pixels. Une hausse non négligeable de l'erreur est également remarquable sur des images 10×10, allant de 40.88° en conditions de réception très favorables à 71.71° pour des conditions très dégradées. Le phénomène s'accroît d'autant plus que l'on affaiblit le C/N_0 : à 43 dBHz, l'erreur double allant de 20° à 40.88° , alors qu'à 37 dBHz l'erreur monte de 26° à 62.75° .

TABLE 4.8 – Erreur d'estimation du paramètre $\Delta\phi_{MP}$ pour différents niveaux de C/N_0 (unité : degrés).

Niveau de C/N_0	Dimension image	$avg_{MAE} \Delta\phi_{MP}$
$C/N_0 = 43$ dBHz	80×80	17.96 ± 2.37
	40×40	20.13 ± 2.08
	20×20	20.00 ± 2.05
	10×10	40.88 ± 2.03
$C/N_0 = 40$ dBHz (Random MP)	80×80	19.90 ± 3.92
	40×40	21.90 ± 3.51
	20×20	22.73 ± 3.15
	10×10	50.63 ± 2.56
$C/N_0 = 37$ dBHz	80×80	19.95 ± 1.67
	40×40	25.06 ± 3.15
	20×20	26.00 ± 3.82
	10×10	62.75 ± 1.84
$C/N_0 = 34$ dBHz	80×80	24.34 ± 3.87
	40×40	26.42 ± 3.30
	20×20	26.59 ± 2.16
	10×10	71.71 ± 1.91

4.4.3 Discussion

Les expériences sur les multitrajets triés par amplitude montrent la possibilité de régression sur les paramètres de multitrajet avec un niveau de précision satisfaisant pour le retard, la fréquence Doppler et l'amplitude du signal. Le retard, uniformément généré sur un intervalle $[0, 1.5 T_c]$, est estimé avec un maximum de $0.26 \pm 0.02 \cdot 10^{-2} T_c$ d'erreur dans le cas d'un multitrajet de faible intensité avec 200 sorties de corrélateur ($10 \times 10 = 100$ pour la voie I, 100 pour la voie Q). Cette erreur est majorée à $7.2 \cdot 10^{-2} T_c$ lorsque l'on fixe un minimum de 20×20 pixels par images. Cette première série d'expériences montre que la perte d'information sur le signal + multitrajet aux sorties de corrélateur conduit toujours vers une dégradation des performances lorsque l'on descend en dessous d'une résolution de 20×20 pour une image. Des performances d'estimation presque équivalentes sont également observées pour des multitrajets d'amplitude $\alpha_{MP} \geq 0.4$ sur des images 80×80 et 40×40 , notamment pour le décalage en fréquence Doppler, laissant penser que des sorties en 40×40 seraient suffisantes pour cette tâche. En revanche le CNN montre des difficultés quant à l'estimation de la phase, avec une erreur dépassant toujours 12° de moyenne et grim pant jusqu'à 70° dans le pire des cas. L'ensemble de données Random MP étant le plus réaliste des quatre *datasets*, on conclut que la phase est un paramètre montrant des difficultés dans son estimation, avec près de 20° d'erreur en moyenne, y compris en utilisant le maximum de sorties allouées par le générateur, c'est-à-dire avec des images de résolution 80×80 .

Avec le deuxième jeu de données, l'algorithme montre une certaine robustesse vis-à-vis de l'estimation de paramètres sur des conditions de réception très différentes. Pour le décalage en fréquence Doppler Δf_D , les performances sur des images 80×80 sont similaires pour tous les C/N_0 testés. Cette deuxième série d'expériences apporte des résultats confirmant les observations effectuées sur les premières expériences : l'utilisation d'images en 80×80 et 40×40 donnent des résultats d'estimation assez proches pour $\Delta \alpha_{MP}$ par exemple, tandis que la phase reste un paramètre compliqué à estimer pour le réseau, avec une erreur ne descendant pas sous le seuil des 17.9° . À propos de ce paramètre, un fait intéressant est à remarquer table 4.8 : pour tous les niveaux de C/N_0 , il y a une augmentation systématique de l'erreur d'estimation d'au moins 2° lorsque que l'on travaille avec moins de 80×80 sorties par images. De plus, les performances d'estimation sont très proches entre des images 40×40 et 20×20 . Cela semble indiquer que l'utilisation de 40×40 pixels ne semble pas améliorer de façon nette la tâche de régression du réseau vis-à-vis de la phase par rapport à 20×20 sorties (c'est-à-dire avec quatre fois moins de pixels). Le phénomène se répète dans la table 4.4 où pour les multitrajets appartenant aux données Strong MP, Weak MP et Random MP, les performances sont quasiment équivalentes pour des images 40×40 et 20×20 (cette observation se faisant également pour les Moderated MP mais entre images 80×80 et 40×40). En revanche, la restriction à l'utilisation de 100 sorties (images 10×10) par voie du signal conduit à une hausse conséquente de l'erreur d'estimation.

Plus généralement, que ce soit pour des niveaux de C/N_0 différents ou des amplitudes de multitrajets distinctes, le réseau semble avoir besoin d'un minimum de 20×20 pixels par image pour éviter une dégradation significative de ses performances d'estimation de paramètres du multitrajets.

4.5 Conclusion

De précédents travaux [21] et [18] ont montré qu'il était possible de détecter des multitrajets sur des images produites par le générateur de sorties de corrélateur (chapitre 2) avec l'utilisation d'un CNN. Notre objectif étant l'estimation des paramètres le caractérisant, une structure plus

4.5. Conclusion

complexe a été élaboré en s'appuyant sur des architectures ayant démontré de bonnes performances d'analyse de plusieurs images en entrée [20]. Les séries d'expériences effectuées sur différents ensembles de données ont démontré la possibilité d'effectuer une estimation par régression des paramètres d'un multitrajet à partir des deux images I & Q. Le réseau a affiché des performances satisfaisantes sur des images de résolution variant de 80×80 à 20×20 . Quelques difficultés se sont manifestées lorsque les images étaient réduites à 10×10 pixels, pour des multitrajets de faibles intensités ainsi qu'avec des niveaux de C/N_0 approchant les 34 dBHz. Cependant, l'estimation de la phase a montré des difficultés bien plus grandes comparées aux autres paramètres. La multiplication des sorties de corrélateur ne montre pas d'améliorations significatives (ex : résultats similaires pour la régression de la phase sur images 40×40 et 20×20) et les performances semblent minorées par un seuil de précision de 10° .

Le réseau étant relativement complexe et performant de manière satisfaisante pour les trois autres paramètres, nous avons cherché à perfectionner les performances en régression du réseau afin d'améliorer l'estimation de la phase. Nous avons également cherché à diminuer la chute de performances lors du passage d'une résolution en 20×20 à 10×10 . La phase étant le seul des paramètres à être défini sur un intervalle modulo 2π , nous avons étudié une solution prenant en compte cette particularité. Ces trois objectifs ont conduit au recours d'une technique appelée le *Distributional Loss* que nous proposons d'appliquer pour la première fois sur un réseau de type CNN et faisant l'objet du chapitre suivant.

Chapitre 5

Méthode du *Distributional Loss* appliquée aux réseaux de neurones convolutionnels

Sommaire

5.1	Lien entre performances et <i>soft labelling</i>	76
5.1.1	Robustesse et généralisation	76
5.1.2	<i>Soft-Labelling</i>	77
5.2	<i>Distributional Loss</i> pour la régression	78
5.2.1	Rappel général régression	78
5.2.2	Principe du <i>Distributional Loss</i>	79
5.2.3	Divergence de Kullback-Leibler	79
5.2.4	Histogramme	79
5.2.5	Fonction perte : <i>Histogram Loss</i> (HL)	80
5.2.6	Choix de la distribution à priori	81
5.2.7	Loi de Von Mises	82
5.3	Application du <i>Distributional Loss</i> aux CNN	84
5.3.1	Calcul MAE moyennée et adaptée à l'architecture en <i>Distributional Loss</i>	84
5.3.2	Expériences	85
5.3.3	Analyse des performances de régression avec <i>Distributional Loss</i>	94
5.4	Conclusion	96

La robustesse est un des enjeux fondamentaux des algorithmes de *Machine Learning* renfermant plusieurs définitions comme la stabilité des performances pour un apprentissage sur des *labels* erronés [79], sur des données de généralisation différentes de celles rencontrées en phase d'entraînement ou face à des attaques adversariales [106][107]. Lors de l'entraînement d'un algorithme de *Machine Learning*, il y a une recherche de performances similaires sur les données d'entraînement à l'étape d'apprentissage et sur les données inédites à l'étape de généralisation. Plusieurs facteurs peuvent dégrader ces performances comme la présence de bruit au sein des entrées [108]. Des techniques ont été développées afin de rendre l'algorithme « robuste » face à ces altérations parmi lesquelles par l'ajout de bruit sur les *labels* [109], la régularisation lors de l'apprentissage par pénalisation de la norme du gradient de la fonction perte [110] ou par rapport à la valeur des poids du réseau [79]. Dans ce chapitre nous présentons une technique modifiant les *labels* et appartenant à la catégorie des méthodes de *soft labelling*. En s'inspirant

5.1. Lien entre performances et *soft labelling*

des travaux effectués par Imani et al. dans [111], le problème de régression sur valeurs continues est transformé en un problème d'apprentissage de densités de distributions discrètes avec pour objectif l'amélioration des performances d'estimation de paramètres du CNN.

Ce chapitre est découpé de la manière suivante : après une introduction portant sur la robustesse des algorithmes de *Machine Learning* et un rappel du concept de *soft labelling*, nous enchaînerons sur la présentation générale de la technique de *Distributional Loss* avec la définition de la fonction perte découlant de la divergence de Kullback-Leibler (KL) et son utilisation couplée aux CNN. Nous analyserons ensuite une série d'expériences utilisant cette technique à des fins de régression de paramètres de multitrajets, notamment en comparant ces performances avec celles obtenues au chapitre 4. Nous mettrons également en avant l'utilisation d'une distribution adaptée à des données de *labels* périodiques.

5.1 Lien entre performances et *soft labelling*

5.1.1 Robustesse et généralisation

La robustesse d'un algorithme de *Machine Learning* peut évoquer plusieurs concepts :

- la robustesse envers les attaques adversariales [106][107] où le principe consiste à duper l'algorithme entraîné avec une faible modification de la donnée d'entrée provoquant une erreur de *labellisation* (par exemple avec la détection par CNN d'un panneau de limitation à 45 miles par heure par simple pose d'un sticker sur un panneau stop [108]),
- la robustesse vis-à-vis des « *out of distribution data* » [112] étant un cas de généralisation sur des données qui n'ont pas de rapport avec celles rencontrées en apprentissage et présentant un fort risque d'erreur dans leur prédiction par l'algorithme [113] (ex : entraînement sur des données contenant exclusivement des images de chat et généralisation sur des images de poissons),
- la robustesse face au *distribution shift* [114] qui est une situation où les données sont sensiblement différentes des données d'entraînement comme dans le cas d'un apprentissage sur un *dataset* insuffisamment exhaustif (ex : dans [71], l'apprentissage de détection de tanks s'est effectué sur des images où les tanks étaient uniquement photographiés par temps couvert et les images sans tanks présentaient un ciel dégagé : la généralisation sur des photos quelconques a montré des erreurs de détection sur des tanks par temps ensoleillé),
- la généralisation sur des données inédites. Il s'agit de la capacité de l'algorithme à obtenir une erreur de généralisation (différence entre les *labels* et les prédictions sur le *dataset* de test) aussi basse ou du moins dans le même ordre de grandeur que l'erreur d'entraînement, obtenue en mesurant la différence entre les *labels* et les prédictions sur le *dataset* d'entraînement [79]. Ce type de robustesse a été étudié dans de nombreux travaux de recherche comme dans la reconstruction d'images [115][116], la classification d'images [117][118][119], la reconnaissance d'objet [120] ainsi qu'en imagerie médicale [121],
- la robustesse envers un apprentissage sur *label* ambiguë/bruité [122, 123] avec l'exemple du test Captcha (explicité en détail en annexe E).

Parmi les phénomènes pouvant affecter l'étape de généralisation d'un algorithme, il y a le sur-apprentissage. Signe d'une trop grande optimisation de l'algorithme vis-à-vis des données d'apprentissage, il se profile par une différence non négligeable entre l'erreur d'entraînement et l'erreur de généralisation : le modèle se focalise sur du bruit plutôt que sur des *features* per-

tinentes décrivant les données [124, 73]. Plusieurs stratégies ont été développées afin de diminuer l'erreur de généralisation (parfois au détriment de l'optimisation de l'erreur d'entraînement) et limiter le phénomène de sur-apprentissage : c'est le principe de régularisation [79]. Parmi ces stratégies, certaines s'appliquent directement sur le réseau : l'ajout d'un terme de pénalisation dans la fonction de perte (ex : pénalisation proportionnelle à la valeur des poids du réseau [125]), la technique de *Dropout* (neutralisation aléatoire d'un pourcentage de neurones en sortie de couche [79]) ou l'*early-stopping* (fin prématurée de l'apprentissage en cas de différence trop importante entre l'erreur d'entraînement et de validation). D'autres stratégies, différentes de la régularisation, agissent sur le *dataset* des données avec l'augmentation du nombre de données en apprentissage par la collecte de davantage d'échantillons d'entraînement ou par *data augmentation* (technique de génération de nouvelles données par l'application de transformations sur les données existantes [126, 127]). Il y a aussi les méthodes de *soft labelling*, appliquant une modification sur les *labels* afin de dégrader la précision des *labels* pour un apprentissage plus général [109]. C'est sur ce type précis de régularisation que nous allons nous focaliser.

La régularisation vise à pénaliser la complexité du modèle [128] afin d'équilibrer le compromis entre optimisation des performances sur les données d'apprentissage et généralisation sur des données inédites (compromis biais-variance).

5.1.2 *Soft-Labelling*

5.1.2.1 Introduction

La mécanique développée par le *soft labelling* consiste à diminuer la confiance de l'algorithme envers les *labels* fournis. Le *soft labelling* est par exemple utilisé pour rendre l'algorithme plus robuste à une généralisation sur un *dataset* de *labels* bruités : le bruit présent dans les *labels* augmente le temps de convergence de la courbe de perte pendant l'entraînement et dégrade les performances de généralisation [129, 130]. Ce phénomène est d'autant plus problématique que ses facteurs sont multiples : variabilité dans la labellisation par utilisateur amateur ou expert [131], erreurs d'annotation ou erreurs de capteur [132]. Les *soft labels* permettent de contrecarrer les effets des *labels* bruités et apportent une amélioration des performances de généralisation par rapport à un apprentissage sans *soft labels* [129]. La méthode de *soft labelling* apporte d'autres bénéfices comme le renforcement l'algorithme par rapport à de potentielles données de *labels* erronés [133].

La représentation fournie avec le *soft labelling* enrichit les informations contenues dans les *labels* : dans [134], les *soft labels* permettent d'affiner les informations liées à une donnée comme dans le cas d'exams médicaux en reflétant le degré de certitude d'un expert médical envers un diagnostic. La méthode peut s'appliquer sur des problèmes de classification comme dans le cadre de reconnaissance faciale [133] ou dans le cadre de la détection du stade d'avancement de cancers [135]. Elle s'applique également sur des problèmes de régression : par exemple dans [136], Xiao et al. présentent une construction de *soft labels* pour l'estimation du triplet d'angles (tangage, roulis, lacet) dans le cadre d'une estimation de la position de la tête dans un environnement 3D à partir d'images.

Le principe du *soft labelling* peut être appliqué par différentes techniques. Le *label smoothing* est l'une d'entre-elles et consiste en l'ajout d'un bruit spécifique sur les *labels*. Dans [137] Peryra et al. appliquent cette technique dans le cadre de reconnaissance vocale par *Deep Learning* avec une pénalisation de la fonction perte du réseau via l'utilisation d'une distribution uniforme pour diminuer la confiance du réseau envers les *labels* fournis. Une application simi-

5.2. *Distributional Loss* pour la régression

laire du *label smoothing* a été utilisée dans [109] avec une architecture *Inception* appliquée sur des images du *dataset* ILSVRC 2012. Une autre technique consiste à remplacer les *labels* sous forme de densités de distribution : sous la forme d'une Gaussienne [138], d'une somme d'un Dirac avec une distribution uniforme [23] ou d'autres densités [139, 123].

5.1.2.2 Performances liées au *soft labelling*

L'utilisation d'une *labellisation* par *soft labelling* a montré une amélioration des performances des algorithmes. Dans [140], Hilton et al. ont montré que l'utilisation de *soft labels* permettait d'améliorer la précision de détection sur le *dataset* MNIST ainsi que pour de la reconnaissance vocale par algorithme de *Deep Learning*. L'application des *soft labels* a permis d'obtenir des performances similaires avec un nombre de données d'entraînement inférieur : un modèle entraîné avec des *soft labels* sur 3 % des données d'entraînement obtenait des performances en généralisation similaires à un modèle sans *soft labels* et entraîné sur la totalité du *dataset* d'entraînement. L'utilisation des *soft labels* a aussi montré le renforcement de la robustesse des CNN face à des attaques adversariales [141]. Dans les expériences d'estimation d'âge ou de position d'un visage sur images conduits par Gao et al. dans [138], l'utilisation de *soft labels* a permis d'accélérer la vitesse de convergence du réseau de neurones à convolution utilisé.

Les *soft labels* peuvent prendre différentes formes en fonction des méthodes utilisées : le *label smoothing* construit les *soft labels* à partir du vecteur *one-hot* (vecteur nul en tout point sauf 1, souvent utilisé dans le cadre de la classification : par exemple pour des données à classer en trois classes, une donnée se verra attribuer un vecteur *one-hot* $v_{hot} = (0, 1, 0)$, soulignant l'appartenance de la donnée à la classe 2) en ajoutant du bruit sur toutes ces composantes afin d'augmenter l'incertitude sur l'appartenance de la classe (ex : le vecteur v_{hot} devient alors $v'_{hot} = (\epsilon, 1 - 2\epsilon, \epsilon)$ avec $\epsilon > 0$). D'autres travaux de recherche définissent des *soft labels* sous la forme de distributions de probabilité [138, 23]. C'est cette dernière méthode qui sera appliquée aux *labels* des images de sorties de corrélateur, c'est-à-dire aux paramètres des multitrajets.

5.2 *Distributional Loss* pour la régression

5.2.1 Rappel général régression

La tâche de régression consiste à construire une fonction capable de calculer l'espace image des données à partir d'un ensemble d'observations. Soit E l'espace des données du problème : on possède un sous-ensemble de données $X \subset E$ (ensemble d'observations) où les observations des données sont connues. En d'autres termes, pour tout $x \in X$ correspond un *label* y connu. Alors l'objectif de la régression est de construire une fonction f telle que $f(x)$ soit connue $\forall x \in E$ avec $f(x) \approx y \forall x \in X$. Lorsque f est représenté par un réseau de neurones N_θ avec θ ses paramètres, la tâche de régression consiste à trouver un $\theta_{opt} \in \Theta$ tel que :

$$\theta_{opt} = \underset{\theta \in \Theta}{\operatorname{argmin}} \left(\sum_{x \in X} \mathcal{L}(N_\theta(x), y) \right)$$

- $N_\theta(x)$ la prédiction du réseau de neurones pour une donnée d'entrée x ,
- Θ l'espace des paramètres θ ,
- $\mathcal{L}(y_1, y_2)$ une distance permettant de quantifier la différence entre deux éléments y_1 et y_2 .

La distance entre les *labels* et les prédictions permet de quantifier l'erreur d'approximation de l'algorithme. Parmi des métriques existantes et fréquemment utilisées, on peut citer la

Mean Absolute Error (MAE), Mean Square Error (MSE) et la Mean Absolute Percentage Error (MAPE) [142].

5.2.2 Principe du *Distributional Loss*

L'idée du *Distributional Loss* est de considérer les *labels* y des données $x \in X$ comme étant issues de distributions de probabilités. Les *labels* sont des réalisations de ces distributions (en l'occurrence leur moyenne) et la tâche d'estimation consiste à approcher la moyenne des distributions. Le *soft Labelling* par *Distributional Loss* propose de s'intéresser directement à la distribution et chercher à l'estimer. La tâche de prédiction de valeurs est ainsi remplacée par une prédiction de distributions de probabilité. Explicitée dans [23], la technique attribue à chaque donnée une distribution construite sur un support contenant toutes les valeurs de *labels* possibles. Soit $S_Y \subseteq \mathbb{R}$ le support de tous les *labels* possibles, on note $Q_x : S_Y \rightarrow [0, 1]$ la distribution à priori attribuée à une donnée x . Le principe du *Distributional Loss* vise à reproduire cette distribution en sortie du réseau et la comparer avec la distribution à priori via une fonction de perte adaptée. On notera $P_x : S_Y \rightarrow [0, 1]$ la prédiction par l'algorithme d'une donnée x . Dans le cadre de la thèse, on suppose que le support S_Y est un compact de \mathbb{R} de la forme $[a, b]$ avec $(a, b) \in \mathbb{R}^2$, $a < b$.

5.2.3 Divergence de Kullback-Leibler

Les *labels* et prédictions étant sous la forme de densités de distributions, il est nécessaire d'avoir un outil capable d'évaluer la différence entre les distributions. Parmi les outils mathématiques capables d'accomplir cette tâche, notre choix s'est porté sur la divergence de Kullback-Leibler (KL) pour sa rapidité de temps de calcul [23, 139], sa simplicité d'implémentation et d'exécution. De plus, Imani et al. [23] ont démontré l'apport de propriétés intéressantes avec l'utilisation de la divergence de KL sur la fonction perte (introduite section 5.2.5) pour des distributions Gaussienne tronquée. Ces propriétés seront explicitées dans la sous-section 5.2.5. Soient deux distributions de probabilité Q et P , la divergence de Kullback-Leibler est définie comme étant :

$$\mathcal{D}_{KL}(Q||P) : \int_{-\infty}^{+\infty} Q(u) \log\left(\frac{Q(u)}{P(u)}\right) du \quad (5.1)$$

5.2.4 Histogramme

La sortie du réseau de neurones à convolution est une couche de *softmax*, composée d'un nombre de neurones arbitraire. Pour représenter les différentes distributions de probabilité, nous allons passer par l'intermédiaire d'histogrammes reproduisant une discrétisation des distributions. Le support des *labels* possibles $[a, b]$ va être divisé en $k \in \mathbb{N}^*$ subdivisions de longueurs identiques $w > 0$, appelée bin :

$$[a, b] = \bigcup_{i=1}^k [a_i, a_i + w]$$

avec $a_i \in [a, b]$.

Chaque bin $[a_i, a_i + w]$ est chargé par la probabilité d'appartenance du *label* d'une donnée x à ce même intervalle. La distribution de probabilité Q_x d'une donnée x devient le vecteur histogramme $q_x = (q_x^1, \dots, q_x^k) \in [0, 1]^k$ où q_x^i est la masse de probabilité de la donnée x calculée sur le bin $[a_i, a_i + w]$. Une illustration de la densité de distribution et de son histogramme

5.2. Distributional Loss pour la régression

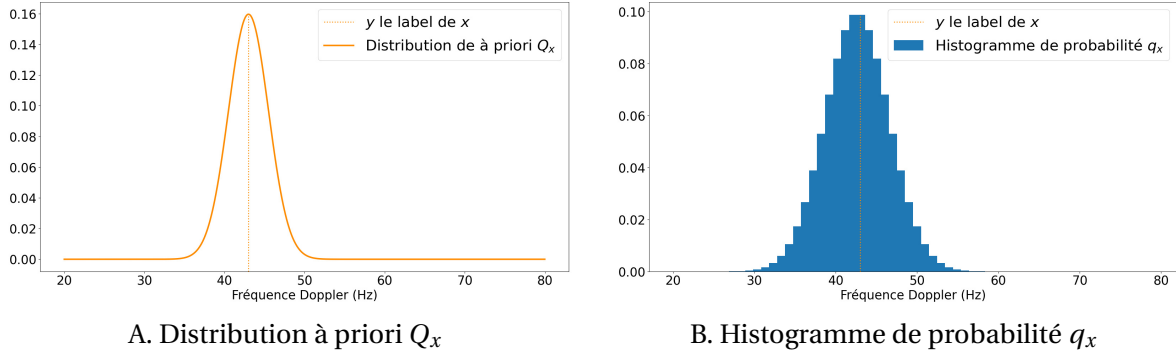


FIGURE 5.1 – Différentes représentations des *soft labels*. Dans la figure A, le *soft label* prend la forme d'une densité de probabilité d'espérance Q_x d'espérance y . Dans la figure B, le *soft label* est l'histogramme q_x construit afin de reproduire la densité de probabilité Q_x . Lors des expériences de *soft labelling* par *Distributional Loss*, le label y est remplacé par l'histogramme q_x . C'est ce nouveau label que l'on cherchera à prédire avec le CNN.

correspondant est donnée à la figure 5.1. De même la prédiction sera un vecteur de k éléments calculés par les sorties de la couche *softmax* du réseau. P_x devient alors le vecteur histogramme $p_x = (p_x^1, \dots, p_x^k) \in [0, 1]^k$. L'utilisation d'une couche de *softmax* en sortie du CNN permet d'assurer deux conditions :

- tous les termes de l'histogramme de prédiction sont positifs : $p_x^i \geq 0 \forall i \in \llbracket 1, k \rrbracket$,
- la somme des composantes de l'histogramme est égale à 1 : $\sum_{i=1}^k p_x^i = 1$.

5.2.5 Fonction perte : *Histogram Loss* (HL)

D'après l'équation 5.1, la divergence de KL entre les distributions et prédictions Q_x et P_x d'une donnée x se décompose en deux termes :

$$\mathcal{D}_{KL}(Q_x || P_x) = \underbrace{\int_{-\infty}^{+\infty} Q_x(u) \log(Q_x(u)) du}_{\text{Entropie différentielle}} - \int_{-\infty}^{+\infty} Q_x(v) \log(P_x(v)) dv$$

Le terme d'entropie différentielle ne dépendant que de la distribution de probabilité à priori Q_x , minimiser la divergence de KL revient simplement à minimiser le terme d'entropie croisée :

$$- \int_{-\infty}^{+\infty} Q_x(v) \log(P_x(v)) dv \quad (5.2)$$

Dans l'algorithme nous considérons des histogrammes de probabilité. La fonction perte découlant du terme d'entropie croisée (équation 5.2) et appelée *Histogram Loss* (HL) [23] est définie comme étant :

$$HL(q_x, p_x) : - \sum_{i=1}^k q_x^i \log(p_x^i)$$

Pour retrouver la valeur de l'estimation représentée par l'histogramme prédit p_x , on définit c_i le milieu du bin $i \in \llbracket 1, k \rrbracket$. La valeur prédite par le réseau est calculée comme suit :

$$\tilde{y} = \sum_{i=1}^k c_i p_x^i \quad (5.3)$$

Avec la connaissance du *label* y et de la prédiction calculée \tilde{y} , on est en mesure de calculer l'erreur d'estimation d'une donnée x par le CNN. Dans la suite, le terme *Convolutional Neural Network - Histogram Loss* (CNN-HL) désignera le réseau de neurones à convolution utilisant la fonction de perte HL sur des distributions de probabilité.

5.2.6 Choix de la distribution à priori

Cette sous-section introduit les deux distributions de probabilités utilisées pour les différents paramètres du multitrajet à estimer : la Gaussienne tronquée pour le retard $\Delta\tau_{MP}$, le décalage en fréquence Doppler Δf_D et l'atténuation $\Delta\alpha_{MP}$ et la loi de Von Mises pour le paramètre de phase $\Delta\phi_{MP}$. Les deux distributions choisies sont définies sur des compacts de \mathbb{R} :

- pour le paramètre $\Delta\tau_{MP}$, le support de définition de la Gaussienne tronquée est $[-0.5 T_c, 2 T_c]$,
- pour le paramètre Δf_D , le support est $[-150 \text{ Hz}, 150 \text{ Hz}]$,
- pour le paramètre $\Delta\alpha_{MP}$, le support est $[-0.4, 1.4]$,
- pour le paramètre $\Delta\phi_{MP}$, le support est $[0, 2\pi]$. Afin de comparer les performances du modèle avec la distribution Gaussienne tronquée et la distribution de Von Mises, l'évaluation a été réalisée sur le même support fermé $[0, 2\pi]$.

5.2.6.1 Loi Gaussienne tronquée

La distribution Gaussienne tronquée a été mise en avant dans l'article [23] comme étant celle donnant les meilleurs résultats parmi les distributions proposées dans l'étude (Dirac, somme d'une loi uniforme et d'un Dirac, somme pondérée de distributions Gaussiennes). Dans un premier temps, nous considérons cette distribution à priori pour tous les paramètres du multitrajet, y compris la phase. Pour une donnée $x \in X$ de *label* $y \in [a, b]$, on suppose par hypothèse que la distribution à priori $Q_x \sim \mathcal{N}_{[a,b]}(\mu, \sigma^2)$ de moyenne $\mu = y$ et d'écart-type $\sigma > 0$. Plus précisément, la densité de probabilité de la Gaussienne tronquée prend la forme :

$$f_{\mu, \sigma, [a,b]}(u) = \frac{1}{\sigma} \frac{g\left(\frac{u-\mu}{\sigma}\right)}{h\left(\frac{b-\mu}{\sigma}\right) - h\left(\frac{a-\mu}{\sigma}\right)} \mathbb{1}_{[a,b]}(u)$$

où g et h sont respectivement les densités et fonction de répartition de la loi normale centrée réduite $\mathcal{N}(0, 1)$.

La discrétisation sous forme d'histogramme est représentée par un vecteur q_x dont les composantes q_x^i sont données par :

$$\forall i \in \llbracket 1, k \rrbracket, q_x^i = \frac{1}{2Z} \left(\text{erf}\left(\frac{a_i + w - \mu}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{a_i - \mu}{\sqrt{2}\sigma}\right) \right)$$

avec $Z = \frac{1}{2} \left(\text{erf}\left(\frac{b-\mu}{\sqrt{2}\sigma}\right) - \text{erf}\left(\frac{a-\mu}{\sqrt{2}\sigma}\right) \right)$ où erf est la fonction d'erreur de Gauss définie comme :

$$\text{erf} : \begin{cases} \mathbb{C} & \rightarrow [-1, 1] \\ x & \mapsto \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \end{cases}$$

Propriétés données par l'utilisation de la fonction HL avec la Gaussienne tronquée

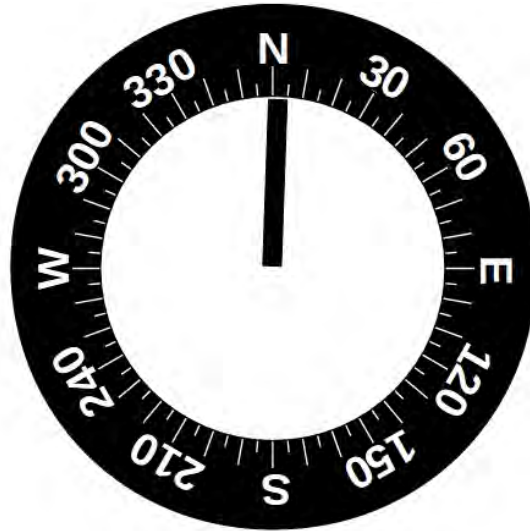


FIGURE 5.2 – Image de compas magnétique avec l’aiguille proche de la valeur de cap ambiguë $0^\circ - 360^\circ$.

Imani et al. ont montré que l’utilisation de la divergence de KL permettait d’obtenir des propriétés intéressantes au niveau de la fonction de perte [23]. La fonction HL est localement Lipschitzienne avec l’utilisation d’une distribution Gaussienne tronquée : le gradient de la fonction perte est donc borné et participe à la stabilité des gradients lors de l’apprentissage. Lors du processus d’apprentissage, des prédictions du réseau « proches » des *labels* induiront une norme du gradient HL plus faible. Une autre propriété découle de la divergence de KL : la minimisation de la divergence entre les deux distributions q_x et p_x revient à borner l’erreur de prédiction du réseau. Ainsi pour une donnée $x \in X$ de *label* $y \in [a, b]$:

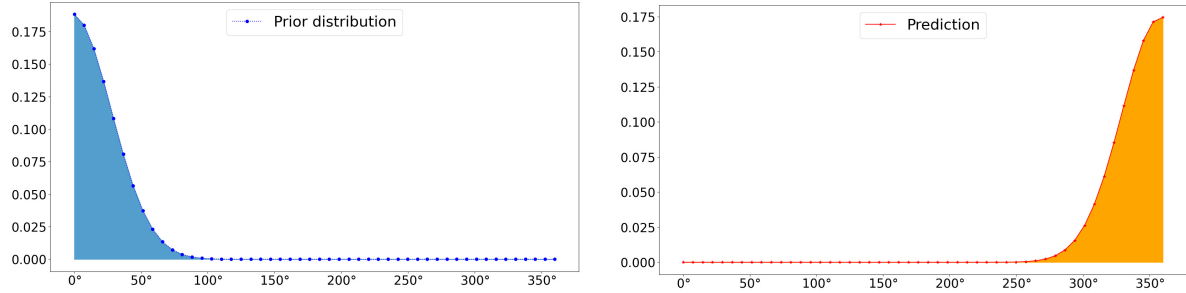
$$(y - \tilde{y})^2 \leq \frac{(b - a)^4}{32} \mathcal{D}_{KL}(q_x || p_x) \quad (5.4)$$

Pour une divergence de valeur faible, l’erreur de prédiction sera aussi faible. La proximité entre deux distributions « proches » au sens de la divergence de KL impliquera une erreur faible entre le *label* et la prédiction. Enfin par rapport aux autres fonctions pertes testées par Imani et al., la distribution Gaussienne tronquée est la moins sensible à des perturbations en entrée.

5.2.7 Loi de Von Mises

Le *soft labelling* construit selon une distribution Gaussienne tronquée ne contient pas d’informations sur le comportement périodique des paramètres. Or pour l’estimation d’un angle compris entre 0° et 360° , il est important de prendre en compte qu’un angle proche de 0° et un angle proche de 360° sont peu éloignés. Pour illustrer le problème, nous allons nous appuyer sur un exemple simple de prédictions d’une variable continue à partir d’une image.

Exemple introductif Nous avons à disposition des images en 80×80 pixels de compas magnétique dont nous donnons un exemple figure 5.2, l’objectif étant de fournir une estimation précise du cap magnétique à partir d’une image de compas.



A. Distribution en Gaussienne tronquée cap=2° Distribution en Gaussienne tronquée cap=355°

FIGURE 5.3 – Comparaison des distributions Gaussiennes tronquées pour des caps proches des extrémités d’un intervalle modulo 2π . La totalité du poids est chargé sur les valeurs proches de 0° sur la figure A : les valeurs significatives sont localisées pour des angles inférieurs à 100° . Pour la distribution gaussienne tronquée construite autour de 355° , les valeurs significatives de la distribution sont localisées pour des angles supérieurs à 250° . La d_{KL} entre ces deux distributions est élevée.

Le cap indiqué est compris dans l’intervalle $[0^\circ, 360^\circ]$, sachant que 360° et 0° représentent exactement le même *label* d’un point de vue angulaire. Pour une donnée $x \in X_c$ (X_c l’ensemble des images de compas) de *label* $y = 2^\circ$ proche des extrémités de l’intervalle des caps possibles (voir figure 5.2), la distribution Gaussienne tronquée construite prend la forme illustrée figure 5.3.A. Si le réseau de neurones renvoie une prédiction centrée autour de 355° , l’erreur entre le *label* $y = 2^\circ$ et sa prédiction $\tilde{y} = 355^\circ$ est faible et égale à 7° . En posant q_x la distribution discrète à priori selon une Gaussienne tronquée de la donnée de *label* $y = 2^\circ$ et p_x la prédiction discrète en Gaussienne tronquée effectuée par le réseau, alors $d_{KL}(q_x||p_x) \approx 12.33$. Ainsi le calcul de la divergence de KL résulte en une erreur importante compte tenu de la dissimilitude flagrante entre la distribution à priori (figure 5.3.A) et la distribution de prédiction (figure 5.3.B) du réseau. L’utilisation du HL ne peut plus assurer une valeur faible de la majoration de l’erreur comme d’après l’inégalité de l’équation 5.4. Dans le cas pris en exemple, l’erreur entre y et \tilde{y} se retrouve majorée par $\sqrt{\frac{(b-a)^4}{32} \mathcal{D}_{KL}(q_x||p_x)} \approx 24.5^\circ$.

Afin d’intégrer l’information périodique de la phase, nous avons choisi une distribution de probabilité tenant compte du caractère cyclique de la phase : la loi de Von Mises. La densité de Von Mises, également appelée distribution normale circulaire, est un cercle centré aux coordonnées $(0,0)$ en 2D et proportionnel à $\exp(\kappa(\theta - \mu))$ où $\kappa \geq 0$ est la concentration des observations de la loi, $\mu \geq 0$ leur localisation et $\theta \in [0, 2\pi]$ un angle [143]. Dans le cas de la loi de Von Mises, la construction des distributions aux extrémités du support est modifiée et intègre le caractère périodique des *labels*. La loi de Von Mises permet une construction plus fidèle des distributions de *labels* proches des extrémités sur un support modulo 2π . Pour un angle $\theta \in [0, 2\pi]$, la distribution de probabilité de Von Mises est définie par la fonction densité suivante :

$$f_{\kappa, \mu}(\theta) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(\theta - \mu)}$$

avec I_0 la fonction de Bessel modifiée à l’ordre zéro.

En reprenant le problème du compas, les distributions Gaussiennes tronquées sont remplacées par des distributions selon une loi de Von Mises. En posant q_x la distribution discrète à priori selon une loi de Von Mises pour la donnée de *label* $y = 2^\circ$ et p_x la prédiction discrète de cette

5.3. Application du *Distributional Loss* aux CNN

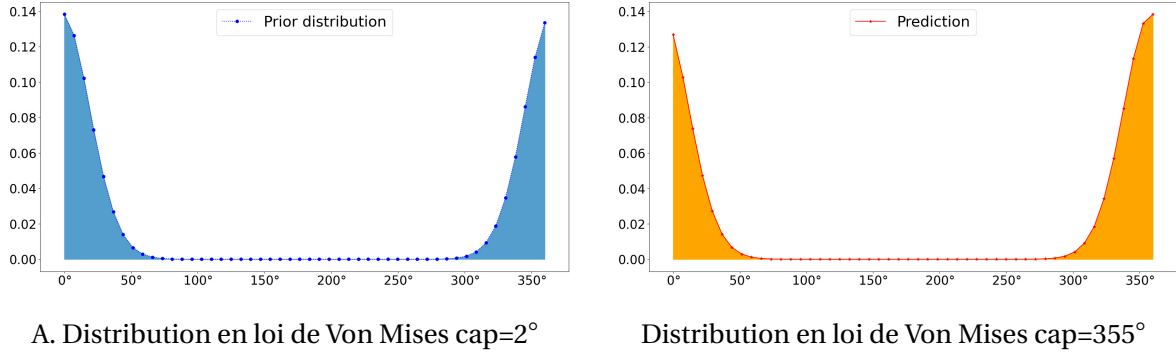


FIGURE 5.4 – Comparaison de distributions selon une loi de Von Mises de caps proches des extrémités d’un intervalle modulo 2π . Pour la distribution en loi de Von Mises construite autour de l’angle égal à 2° (figure 5.4.A), on remarque qu’une partie des poids est attribuée aux bins proches de 360° , plus précisément dans l’intervalle $[300^\circ, 360^\circ]$. De même pour la distribution centrée autour de 355° , les bins concernant les angles $[0^\circ, 100^\circ]$ sont chargés de poids significatifs. Les deux distributions sont similaires et de divergence de KL faible.

donnée selon une loi de Von Mises avec $\tilde{y} = 355^\circ$, alors la divergence de KL sous forme discrète vaut $d_{KL}(q_x||p_x) \approx 0.82$.

Les divergences de KL obtenues pour de mêmes *labels* mais de distributions construites à partir de lois distinctes (Gaussienne tronquée et en loi de Von Mises) sont d’un ordre de grandeur différent, avec une divergence 15 fois plus élevée dans le cas des distributions de probabilité en Gaussienne tronquée. Ainsi pour des données $x \in X_c$ de *labels* proches des extrémités du support, les distributions à priori (figure 5.4.A) et leurs prédictions (figure 5.4.B) montrent une divergence de KL beaucoup plus faible et plus représentative de la différence réelle entre les caps et leurs prédictions.

5.3 Application du *Distributional Loss* aux CNN

La structure du réseau est reprise de l’architecture CNN présentée au chapitre 4. Seule la couche de sortie est modifiée avec une sortie en *softmax* afin de construire un histogramme. L’architecture du réseau utilisant la technique de *soft labelling* est illustrée figure 5.5.

5.3.1 Calcul MAE moyennée et adaptée à l’architecture en *Distributional Loss*

Dans le cas de la régression avec une sortie unique, le calcul de MAE sur $N_r \in \mathbb{N}^*$ expériences s’effectue selon la formule donnée par l’équation 4.1. Dans le cas de sorties en densités de distributions, le calcul de MAE nécessite le calcul intermédiaire de la moyenne de la prédiction (sous forme d’une distribution) donnée par le CNN-HL. Pour un seul *run* le calcul de MAE prend la forme suivante :

$$\frac{1}{N_D} \sum_{l=1}^{N_D} |\tilde{y}_l - y_l|$$

- $\tilde{y}_l = \sum_{i=1}^k c_i p_{x_l}^i$ l’estimation de la valeur prédite à partir de l’histogramme en sortie du réseau,
- $k \in \mathbb{N}^*$ la dimension de la couche de sortie du réseau,

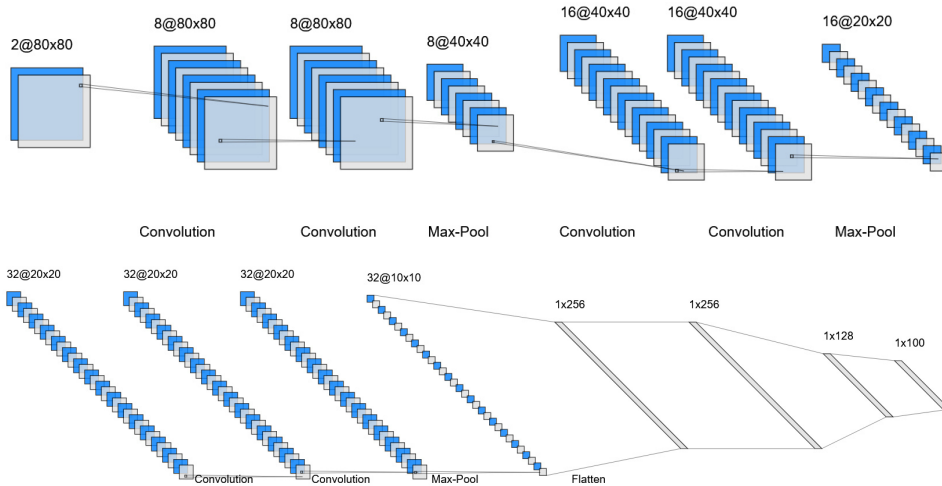


FIGURE 5.5 – Architecture du CNN-HL pour des images 80×80. La notation $X@YxZ$ pour une couche donnée du réseau indique un nombre X d’images de dimensions YxZ . L’image du haut représente les différentes couches de convolution et de sous-échantillonnage composant la première partie du réseau. L’image du bas représente les différentes couches de convolution manipulant les *features map* les plus informatives, la couche de *Flatten* et les couches denses reconstruisant la courbe de probabilité au niveau du *softmax* en sortie. Image générée avec l’outil NN-SVG [104].

- $N_D \in \mathbb{N}^*$ le cardinal de l’ensemble de données D (ensemble des paires d’images de multi-trajet),
- $p_{x_l}^i$ la pondération associée à la $i^{\text{ème}}$ sortie du réseau pour une donnée $x_l \in D$.

En généralisant le calcul sur N_r runs, la MAE moyenne est donnée par :

$$\text{avg}_{MAE} = \frac{1}{N_r} \sum_{j=1}^{N_r} \left[\frac{1}{N_{Dj}} \sum_{l=1}^{N_{Dj}} |y_{l,j} - \tilde{y}_{l,j}| \right]$$

où

- N_{Dj} le cardinal de l’ensemble de données Dj (ensemble des paires d’images I & Q),
- $\tilde{y}_{l,j}$ l’estimation de la valeur prédite pour le run j ,
- $x_{l,j}$ une donnée x_l au run j ,
- $y_{l,j}$ le label du $l^{\text{ème}}$ exemple d’apprentissage au run j .

5.3.2 Expériences

Différents types d’expériences ont été menées afin de mesurer les performances d’estimation entre le CNN-Reg (estimation sans la méthode de *soft labelling*) et le CNN-HL sur différents points de comparaison : en fonction du niveau de C/N_0 des images, du nombre de sorties utilisées ainsi que de l’amplitude des multitrajets à estimer. D’autres expériences ont également été conduites afin de montrer l’influence des hyper-paramètres σ et κ (cas du *Distributional Loss* avec une loi de Von Mises) des lois à priori sur les performances du CNN-HL.

Les tables 5.1, 5.2, 5.3 et 5.4 comparent les performances du CNN-Reg et du CNN-HL sur les ensembles de données Strong MP, Moderated MP, Weak MP et Random MP (voir section 2.7).

5.3. Application du *Distributional Loss* aux CNN

Les figures 5.6.A et B représentent des distributions prédites en sortie du réseau, correspondant à un cas avec faible erreur et un cas d'erreur d'estimation élevée. Les figures 5.7, 5.8, 5.9 et 5.10 illustrent les courbes de performances du CNN-Reg du CNN-HL pour plusieurs résolutions d'images (variation du nombre de sorties de corrélateur) par rapport au C/N_0 lors de la génération des images. Enfin les tables en annexe C contiennent les performances d'estimation CNN-HL pour différentes valeurs des hyper-paramètres σ et κ .

5.3.2.1 Comparaison des performances des CNN-Reg et CNN-HL à niveau de bruit constant

Les expériences menées dans cette sous-section simulent une perturbation des sorties de corrélateur par la présence d'un multitrajet dans des conditions de réception favorables. Les images I & Q ont toutes été générées à un niveau de $C/N_0=40$ dBHz. On souhaite ainsi mesurer la différence de précision sur l'estimation des différents paramètres de multitrajet avec l'utilisation des *soft labels* avec des distributions Gaussiennes tronquées et selon une loi de Von Mises (pour le paramètre de phase). La dichotomie des expériences est organisée en fonction de l'amplitude des multitrajets et des tailles d'images.

Dans la table 5.1 sont comparées les erreurs d'estimation du *delay* pour le CNN-Reg et le CNN-HL. Une meilleure performance d'estimation est observée pour le CNN-HL sur toutes les amplitudes de multitrajets et pour toutes tailles d'images testées. On remarque également une augmentation de la précision apportée par l'application du *soft labelling* par *Distributional loss* : les erreurs d'estimation sont moins élevées par CNN-HL sur des images de dimensions 20×20 que par le CNN-Reg sur des images de tailles inférieures, similaires et supérieures. À titre de comparaison, sur des images 80×80 le CNN-Reg montre une MAE moyenne quatre fois plus importante ($4.64 \cdot 10^{-2} Tc$) que proposée par le CNN-HL sur des images 20×20 ($1.01 \cdot 10^{-2} Tc$) pour des multitrajets de faibles intensités. Moins les images contiennent de pixels et plus l'écart de performances entre les deux algorithmes diminue. Pour des multitrajets appartenant à l'ensemble Moderated MP, de 80×80 à 10×10 pixels, la différence de performances entre les deux techniques d'estimation passe d'un rapport quatre à un rapport proche de 1. Comme dans les sections 4.4.1 et 4.4.1, une hausse de l'erreur d'estimation est observée lors du passage d'images 20×20 à 10×10 . Les performances en CNN-HL subissent une multiplication de leurs erreurs d'estimation plus prononcée que celles du CNN-Reg : l'erreur de 20×20 à 10×10 est au

minimum multipliée par trois sur *soft label*, tandis que l'erreur moyenne de 20×20 à 10×10 est multipliée par trois dans le cas le plus défavorable pour de la régression sur *hard label*.

TABLE 5.1 – Erreur moyenne d'estimation du paramètre $\Delta\tau_{MP}$ par le CNN-Reg et le CNN-HL selon les différentes amplitudes des multitrajets (unité de la table 10^{-2} Tc).

Ensemble de données	Dimensions des images	Algorithme	
		CNN-Reg	CNN-HL
Strong MP	80×80	2.50 ± 0.22	0.64 ± 0.18
	40×40	2.91 ± 0.32	0.76 ± 0.11
	20×20	4.30 ± 0.42	1.28 ± 0.20
	10×10	6.70 ± 0.40	4.85 ± 0.17
Moderated MP	80×80	3.03 ± 0.54	0.66 ± 0.04
	40×40	3.44 ± 0.65	0.96 ± 0.26
	20×20	4.44 ± 0.45	1.63 ± 0.15
	10×10	9.90 ± 0.72	7.35 ± 0.38
Weak MP	80×80	4.64 ± 1.44	1.01 ± 0.13
	40×40	4.33 ± 0.42	1.55 ± 0.20
	20×20	7.20 ± 1.55	2.86 ± 0.40
	10×10	26.13 ± 1.96	21.17 ± 0.58
Random MP	80×80	3.82 ± 0.68	0.81 ± 0.11
	40×40	4.92 ± 0.94	1.20 ± 0.21
	20×20	7.00 ± 0.78	2.21 ± 0.29
	10×10	16.34 ± 1.19	12.1 ± 0.59

Les performances d'estimation du décalage en fréquence Doppler dans la table 5.2 montrent des observations similaires à la table 5.1. Le CNN-HL affiche une erreur d'estimation du paramètre plus faible que le CNN-Reg dans les mêmes conditions. Sur un même ensemble de multitrajet, le CNN-HL est capable de fournir sur des images en 40×40 pixels une moyenne erreur d'estimation inférieure à celle observée par le CNN-Reg sur toutes les tailles d'images générées, y compris celles en 80×80 . Comme dans la table précédente, la dégradation des per-

5.3. Application du *Distributional Loss* aux CNN

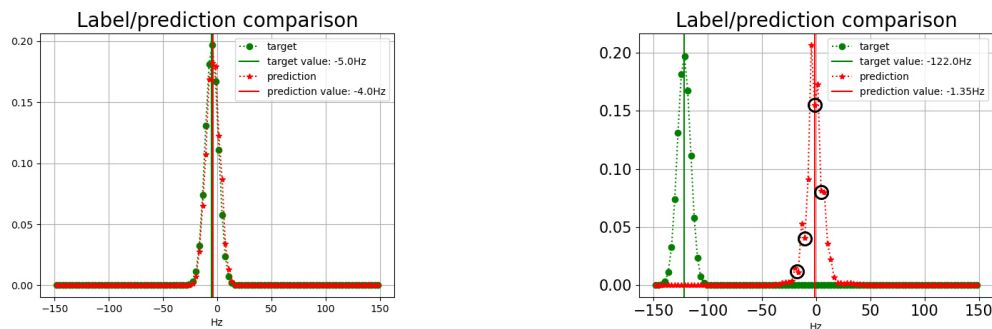
performances est plus prononcée lors du passage d'images de 20×20 à 10×10 que de 80×80 à 40×40 et de 40×40 à 20×20 pixels.

TABLE 5.2 – Erreur moyenne d'estimation du paramètre Δf_D par le CNN-Reg et le CNN-HL selon les différentes amplitudes des multitrajets (unité de la table : Hz).

Ensemble de données	Dimensions des images	Algorithme	
		CNN-Reg	CNN-HL
Strong MP	80×80	1.22 ± 0.20	0.60 ± 0.04
	40×40	1.60 ± 0.27	0.75 ± 0.07
	20×20	2.36 ± 0.41	1.17 ± 0.12
	10×10	5.16 ± 0.26	3.60 ± 0.18
Moderated MP	80×80	1.33 ± 0.29	0.72 ± 0.06
	40×40	1.70 ± 0.23	0.87 ± 0.06
	20×20	2.61 ± 0.45	1.52 ± 0.11
	10×10	7.08 ± 0.36	5.52 ± 0.27
Weak MP	80×80	1.75 ± 0.31	1.26 ± 0.15
	40×40	2.41 ± 0.38	1.42 ± 0.13
	20×20	4.30 ± 0.41	2.84 ± 0.45
	10×10	21.48 ± .68	19.04 ± 1.07
Random MP	80×80	1.62 ± 0.34	0.95 ± 0.08
	40×40	2.06 ± 0.36	1.10 ± 0.13
	20×20	3.40 ± 0.34	2.10 ± 0.28
	10×10	12.40 ± 0.76	9.71 ± 0.22

Les figures 5.6.A et 5.6.B illustrent deux cas de prédiction de décalage en fréquence Doppler par le CNN-HL. Les distributions à priori et de prédiction sont illustrées par interpolation des histogrammes à priori et de prédiction. Dans les deux figures, la courbe verte « *target* » est une représentation de la distribution à priori. Dans le cas de la figure A la distribution à priori est une Gaussienne tronquée d'espérance $\mu_A = -5$ Hz, dans la figure B la distribution correspond à une autre donnée, d'espérance $\mu_B = -122$ Hz. Les courbes rouges « *prediction* » représentent la distribution de prédiction en sortie du réseau, chaque figure correspondant à la prédiction d'une donnée différente. L'espérance des distributions de prédiction est calculée selon l'équation 5.3 : ainsi pour la figure 5.6.A $\tilde{\mu}_A = -4$ Hz et pour la figure 5.6.B $\tilde{\mu}_B = -1.35$ Hz. La figure 5.6.A expose un cas de prédiction satisfaisante, avec une erreur d'estimation $|\mu_A - \tilde{\mu}_A| = 1$ Hz. On remarque que les distributions à priori et de prédiction se superposent : le CNN-HL a réussi à reconstruire une loi Gaussienne tronquée d'un écart-type similaire et d'une espérance différente à un hertz près de la valeur μ_A . La figure 5.6.B expose un cas de prédiction erronée. La prédiction prend également une forme de Gaussienne tronquée. Cependant on remarque la présence de creux au niveau de certains *bins* à proximité du pic de la distribution de prédiction (entourés en noir). Contrairement à la figure 5.6.A, la sortie du réseau n'est pas une reconstruction correcte d'une Gaussienne tronquée. De plus, l'espérance $\tilde{\mu}_B$ calculée à partir de la distribution de prédiction est différente de l'espérance de la distribution à priori μ_B : ici l'erreur $|\mu_B - \tilde{\mu}_B|$ est supérieure à 100 Hz.

Dans la table 5.3, les performances du CNN-HL sur l'estimation de l'amplitude du multitrajet sont meilleures qu'avec l'utilisation d'une architecture en CNN-Reg. Comme pour le *delay*, l'utilisation de *soft labels* permet d'obtenir de meilleurs résultats sur des images 20×20 qu'en régression sur des *labels* sous forme de valeurs continues avec des images de meilleures qua-



A. Comparaison des distributions à priori et de prédiction sur une donnée de *label* -5 Hz

B. Comparaison des distributions à priori et de prédiction sur une donnée de *label* -122 Hz

FIGURE 5.6 – Comparaison des distributions à priori et de prédiction dans un cas de prédiction proche du *label* et dans un cas de prédiction éloignée du *label*. Dans la figure A, la distribution à priori et la prédiction en sortie du réseau de neurones sont similaires au niveau de la forme de leur distribution ainsi que de leur écart-type. Les deux distributions se superposent et sont presque confondues. La prédiction donnée par le réseau de neurones est ici considérée comme correcte, l'espérance de la prédiction (-4 Hz) étant proche de l'espérance de la distribution à priori (-5 Hz). Dans la figure B en revanche, la distribution à priori et la prédiction sont distinctes. La distribution reconstruite en prédiction s'approche d'une distribution en Gaussienne tronquée, avec quelques bins montrant une difficulté de reconstruction d'une distribution en Gaussienne tronquée (entourés en noire sur la figure, quelques bins posent problème comme par exemple la présence de 2 pics autour de -1.35 Hz pour la prédiction alors que la distribution en Gaussienne tronquée est une distribution possédant un seul pic). La différence entre l'espérance de la prédiction et l'espérance de la distribution à priori est notable : plus de 120 Hz d'écart. La figure B est un exemple de mauvaise prédiction.

5.3. Application du *Distributional Loss* aux CNN

lités. L'erreur augmente et se rapproche des valeurs obtenues en CNN-Reg lors de l'utilisation d'images en 10x10.

TABLE 5.3 – Erreur moyenne d'estimation du paramètre $\Delta\alpha_{MP}$ par le CNN-Reg et le CNN-HL selon les différentes amplitudes des multitrajets (unité de la table : % de l'amplitude de LOS).

Ensemble de données	Dimensions des images	Algorithme	
		CNN-Reg	CNN-HL
Strong MP	80×80	3.95 ± 0.93	1.30 ± 0.26
	40×40	4.45 ± 0.61	1.48 ± 0.14
	20×20	5.60 ± 0.70	2.22 ± 0.20
	10×10	7.32 ± 0.40	5.99 ± 0.25
Moderated MP	80×80	3.71 ± 0.64	1.08 ± 0.05
	40×40	3.76 ± 0.53	1.22 ± 0.15
	20×20	4.79 ± 0.55	1.73 ± 0.10
	10×10	5.97 ± 0.16	5.00 ± 0.07
Weak MP	80×80	3.53 ± 0.88	1.91 ± 1.16
	40×40	4.06 ± 1.28	1.38 ± 0.57
	20×20	4.09 ± 1.10	1.66 ± 0.22
	10×10	7.87 ± 0.33	6.08 ± 0.12
Random MP	80×80	3.48 ± 0.98	1.45 ± 0.19
	40×40	3.78 ± 0.60	1.60 ± 0.14
	20×20	4.97 ± 0.61	2.21 ± 0.28
	10×10	8.28 ± 0.24	6.84 ± 0.12

Pour le cas particulier de la phase les performances du CNN-Reg sont comparées à deux CNN-HL, chacun ayant été entraîné suivant différentes distributions qui sont la Gaussienne tronquée et la loi de Von Mises. Le premier constat que l'on fait table 5.4 est la supériorité des techniques de *soft labelling* comparées à la régression sans *soft labels*. Le choix d'une loi de Von Mises montre un meilleur niveau de précision du réseau par rapport à la distribution en Gaussienne tronquée. Par exemple sur les ensembles Strong MP et Moderated MP, les performances du CNN-HL-VM (avec une distribution en loi de Von Mises) sont meilleures sur des images en 20×20 comparées aux performances atteintes par le CNN-HL-GT (avec une distribution en Gaussienne tronquée), y compris pour des images de meilleures résolutions, en 40×40 et 80×80. Les différences d'erreurs entre ces deux types de *Distributional Loss* s'atténuent pour des multitrajets d'amplitudes plus faibles et/ou lors de la diminution des dimensions d'images. De manière générale, l'erreur d'estimation est en moyenne trois fois plus faible pour les CNN-HL comparée au CNN-Reg. Avec la loi de Von Mises cette différence peut parfois atteindre un facteur 5. L'utilisation de la technique d'apprentissage HL et plus particulièrement avec la loi de Von Mises permet de conserver des performances d'estimation satisfaisantes. L'erreur sur des images 80×80, 40×40 et 20×20 ne dépasse jamais les 10° pour le CNN-HL-VM alors que celle-ci

est supérieure à 12° même sur des images de multitrajets puissants en 80×80 (c'est-à-dire dans les conditions expérimentales les plus favorables).

TABLE 5.4 – Erreur moyenne d'estimation du paramètre $\Delta\phi_{MP}$ par le CNN-Reg, le CNN-HL-GT (distribution en Gaussienne tronquée) et le CNN-HL-VM (distribution avec une loi de Von Mises) selon les différentes amplitudes des multitrajets (unité de la table : degrés).

Ensemble de données	Dimensions des images	Algorithme		
		CNN-Reg	CNN-HL GT	CNN-HL VM
Strong MP	80×80	12.30 ± 1.46	4.26 ± 0.46	2.21 ± 0.16
	40×40	15.41 ± 3.46	5.06 ± 0.61	2.62 ± 0.18
	20×20	15.59 ± 2.44	5.68 ± 0.62	3.84 ± 0.23
	10×10	29.68 ± 2.06	19.88 ± 0.88	13.16 ± 0.57
Moderated MP	80×80	15.71 ± 3.07	5.13 ± 0.72	3.08 ± 0.35
	40×40	15.72 ± 1.74	5.61 ± 0.32	3.70 ± 0.41
	20×20	17.90 ± 1.99	7.28 ± 0.64	4.48 ± 0.48
	10×10	42.55 ± 2.30	29.78 ± 1.44	19.90 ± 0.85
Weak MP	80×80	26.45 ± 6.25	9.29 ± 1.40	6.46 ± 1.24
	40×40	30.06 ± 3.35	12.26 ± 1.20	7.26 ± 0.54
	20×20	29.57 ± 3.17	16.27 ± 1.86	9.58 ± 1.08
	10×10	70.23 ± 1.56	60.53 ± 2.31	51.53 ± 2.41
Random MP	80×80	19.90 ± 3.92	7.02 ± 0.67	4.47 ± 0.66
	40×40	21.90 ± 3.51	8.26 ± 0.57	5.39 ± 0.78
	20×20	22.73 ± 3.15	10.00 ± 1.06	7.21 ± 0.82
	10×10	50.63 ± 2.56	37.85 ± 1.26	28.48 ± 1.66

5.3.2.2 Comparaison des performances de CNN-Reg et CNN-HL à niveau de bruit variable

Dans cette sous-section d'expériences, nous utilisons des images I & Q générées à différents niveaux de C/N_0 afin de simuler des conditions de réception bonnes (43 dBHz) à dégradées (34 dBHz). Les *datasets* ne sont plus triés selon l'amplitude des multitrajets sinon par le niveau de bruit présent sur les images. L'objectif est la mesure de la robustesse des performances d'estimation du CNN-HL vis-à-vis du bruit présent aux sorties de corrélateur et de la résolution des images en entrée.

Les figures 5.7, 5.8, 5.9 et 5.10 démontrent la supériorité en performances du CNN-HL. Sur une même taille d'image et pour des niveaux de C/N_0 équivalents, l'erreur d'estimation sur CNN-Reg est toujours supérieure à celle observée sur CNN-HL. On constate par exemple sur la figure 5.7 pour des images de dimensions de 20×20 , 40×40 et 80×80 pixels, l'erreur d'estimation du CNN-HL dans des conditions de réception très défavorables ($C/N_0 = 34$ dBHz) est plus faible que pour le CNN-Reg dans des conditions de réception bonnes ($C/N_0 = 43$ dBHz). Sur des images en 10×10 , on remarque que les deux architectures sont affectées de la même manière par la dégradation des conditions de réception, avec toujours un avantage pour le CNN-HL.

Pour la figure 5.8, les résultats en 80×80 sont similaires : les deux méthodes donnent une erreur d'estimation toujours inférieure à 2 Hz. Ici aussi, pour une même taille d'image, le CNN-HL obtient une erreur d'estimation plus faible que le CNN-Reg. Hors images 10×10 , les performances d'estimation du CNN-HL dans des conditions de réception dégradées sont meilleures comparées aux erreurs d'estimation obtenues par le CNN-Reg dans n'importe quelles conditions testées. Les écarts-types des erreurs d'estimation du CNN-Reg sont plus élevés que ceux obtenus par le CNN-HL à taille d'image et C/N_0 équivalents. On observe comme pour la figure 5.7

5.3. Application du *Distributional Loss* aux CNN

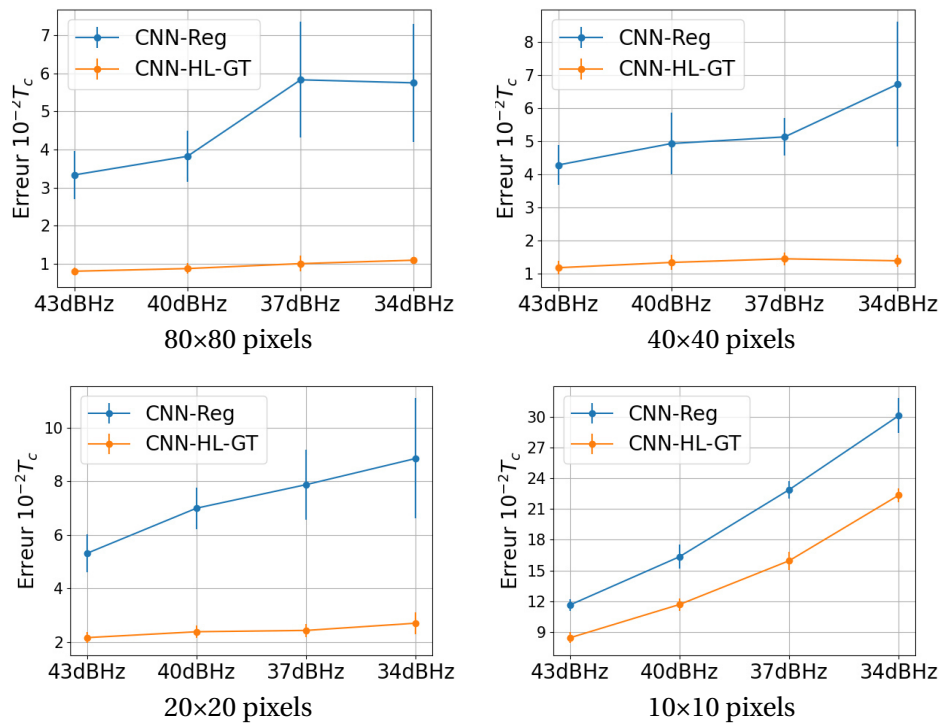


FIGURE 5.7 – Erreur moyenne (avg_{MAE}) d'estimation de $\Delta\tau_{MP}$ en fonction du C/N_0 pour différentes résolutions d'images.

une évolution de l'erreur d'estimation similaire en fonction du C/N_0 pour des images de 10×10 pixels.

Concernant l'estimation de l'amplitude du multitrajet en figure 5.9, le CNN-HL montre une supériorité dans la tâche de régression sur toutes les dimensions d'images testées ainsi que pour toutes les conditions de réception créées. Une nette difficulté d'estimation se profile toujours pour les deux méthodes concernant le travail sur des images en 10×10 pixels.

Pour l'estimation de la phase du multitrajet dans la figure 5.10, la différence entre les performances des deux architectures HL est quasi-constante pour des images en 80×80, 40×40 et 20×20 pixels. L'écart de performances est plus important entre le CNN-Reg et les CNN-HL qu'entre les CNN-HL-GT et CNN-HL-VM. Pour des résolutions 80×80, 40×40 et 20×20, les performances du CNN-HL-VM en conditions dégradées ($C/N_0 = 34$ dBHz) sont similaires aux performances du CNN-HL-GT en conditions de réception bonnes ($C/N_0 = 43$ dBHz). Comme pour les autres paramètres, une difficulté croissante de la tâche d'estimation appara pour tous les algorithmes sur des images 10×10 au fur et à mesure que les multitrajets à estimer sont dans des conditions se détériorant.

5.3.2.3 Influence des hyper-paramètres σ & κ sur les performances d'estimation

L'écart-type σ pour la Gaussienne tronquée et son homologue κ pour la loi de Von Mises appartiennent aux hyper-paramètres liés à l'utilisation du *Distributional Loss*. Leurs valeurs dilatent plus ou moins la distribution et règlent donc leur « capacité » de *soft labelling*, c'est-à-dire la répartition de l'information du *label* de départ dans la distribution à priori. Le schéma 5.11 illustre les distributions à priori pour des σ différents et d'espérance $\mu = 45$ Hz (*label initial*).

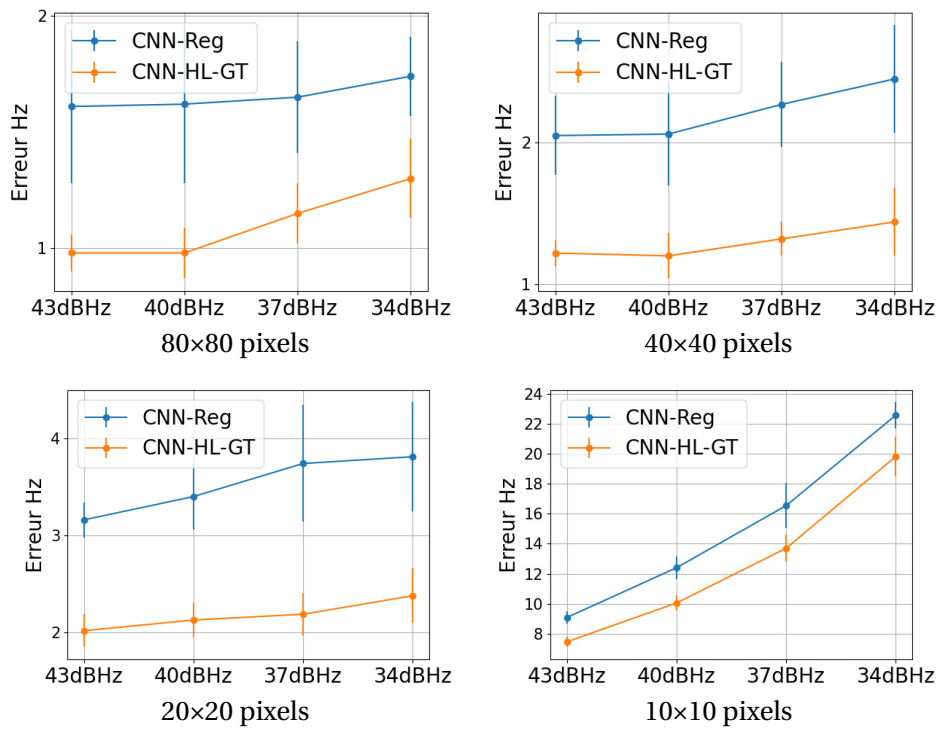


FIGURE 5.8 – Erreur moyenne (avg_{MAE}) d'estimation de Δf_D en fonction du C/N_0 pour différentes résolutions d'images.

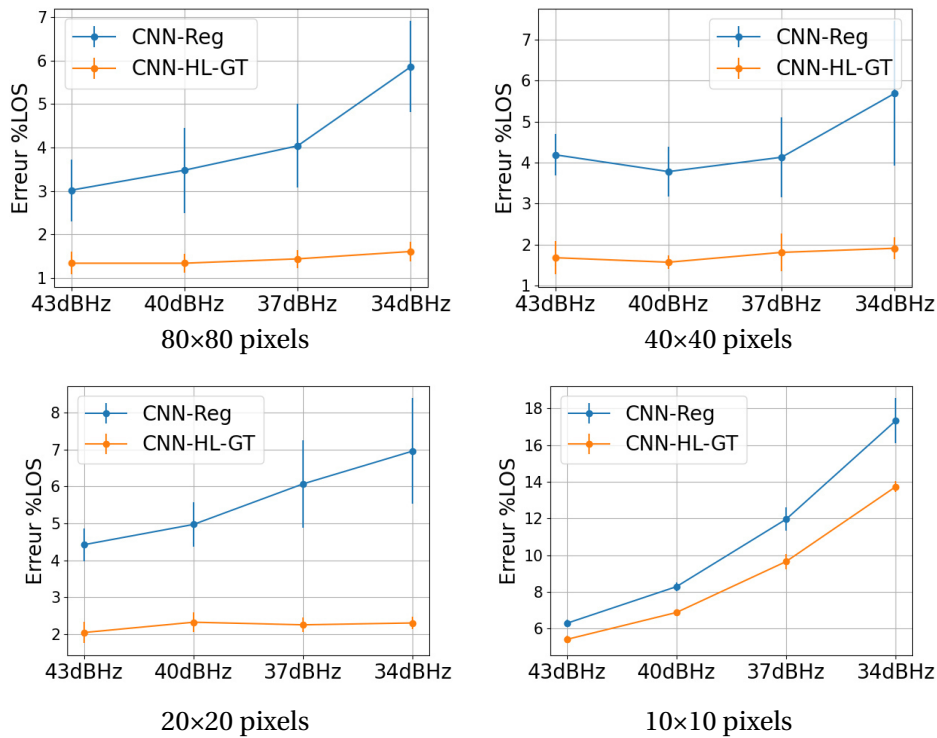


FIGURE 5.9 – Erreur moyenne (avg_{MAE}) d'estimation de $\Delta \alpha_{MP}$ en fonction du C/N_0 pour différentes résolutions d'images.

5.3. Application du *Distributional Loss* aux CNN

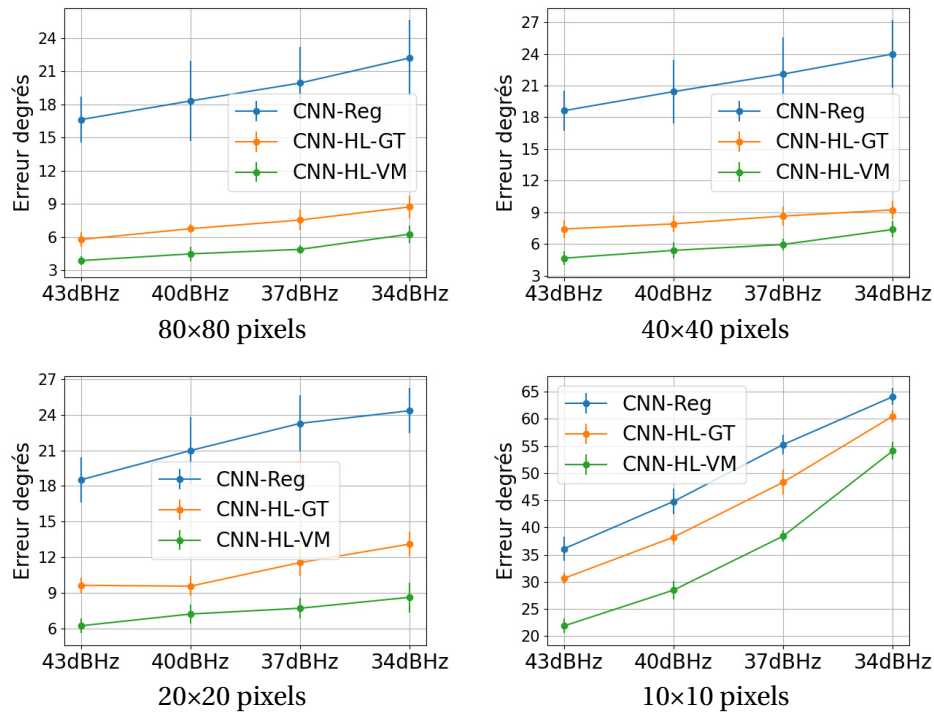


FIGURE 5.10 – Erreur moyenne (avg_{MAE}) d'estimation de $\Delta\phi_{MP}$ en fonction du C/N_0 pour différentes résolutions d'images.

On remarque que plus la valeur de σ est élevée, plus la courbe est large : l'effet *soft labelling* est plus important pour un σ élevé. Au contraire, plus cet hyper-paramètre a une valeur proche de zéro, plus la distribution se resserre. Le cas extrême étant atteint pour un $\sigma \rightarrow 0$, la distribution devient un Dirac et tout le poids est chargé sur l'espérance μ : cela revient à faire de la régression « pure » comme le CNN-Reg. De même en faisant tendre $\sigma \rightarrow +\infty$, la courbe se rapproche d'une distribution uniforme, avec cette fois-ci une perte totale de l'information concernant la *label* initial. Le même phénomène s'observe pour une distribution en loi de Von Mises de manière inversée. κ étant l'analogie à $\frac{1}{\sigma^2}$, la distribution de la loi de Von Mises se retrouvera plus resserrée pour de grandes valeurs de κ et inversement (cf figure 5.12). Les tables compilant des expériences comparant les performances de régression en fonction du réglage de ces hyper-paramètres sont disponibles en annexe C. Les algorithmes ont été entraînés sur l'ensemble de données Random MP ainsi que sur des ensembles de données contenant des multitrajets plus réalistes (et introduits au chapitre 6). Les tables permettent d'observer des valeurs optimales de σ et κ donnant de meilleures performances en régression.

5.3.3 Analyse des performances de régression avec *Distributional Loss*

La section d'expérimentation 5.3.2 a permis de mettre en lumière une amélioration de la précision d'estimation du CNN avec l'utilisation de la méthode du *Distributional Loss*. Les performances d'estimation sur tous les paramètres sont supérieures pour des images de moindres qualités et dans des conditions plus dégradées. Il y a un gain de performances notable entre la régression par CNN avec *soft labels* et la régression par CNN utilisant directement les *labels* sous forme de variables continues. La substitution de la distribution en Gaussienne tronquée par une loi de Von Mises afin de prendre en compte le caractère périodique de la phase a permis

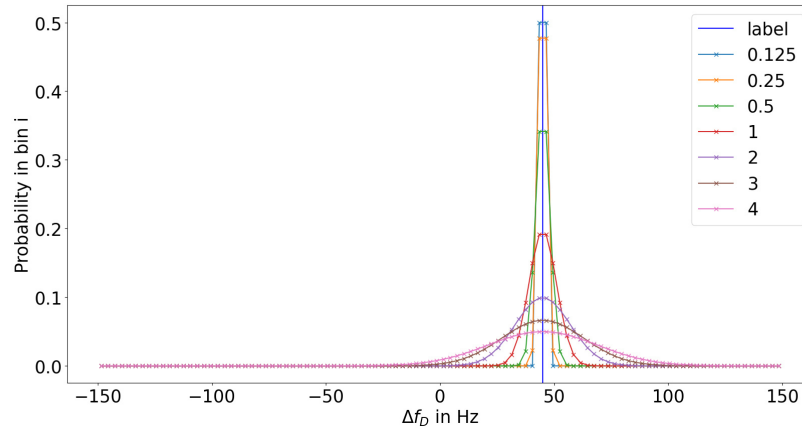


FIGURE 5.11 – Distributions à priori construites par interpolation à partir des histogrammes à priori en Gaussienne tronquée pour différents écarts-types σ et d’espérance $\mu = \Delta f_D = 45$ Hz. σ impacte à la fois la largeur des courbes de distributions ainsi que leur amplitude (notamment le pic localisé au niveau du *label* $\Delta f_D = 45$ Hz).

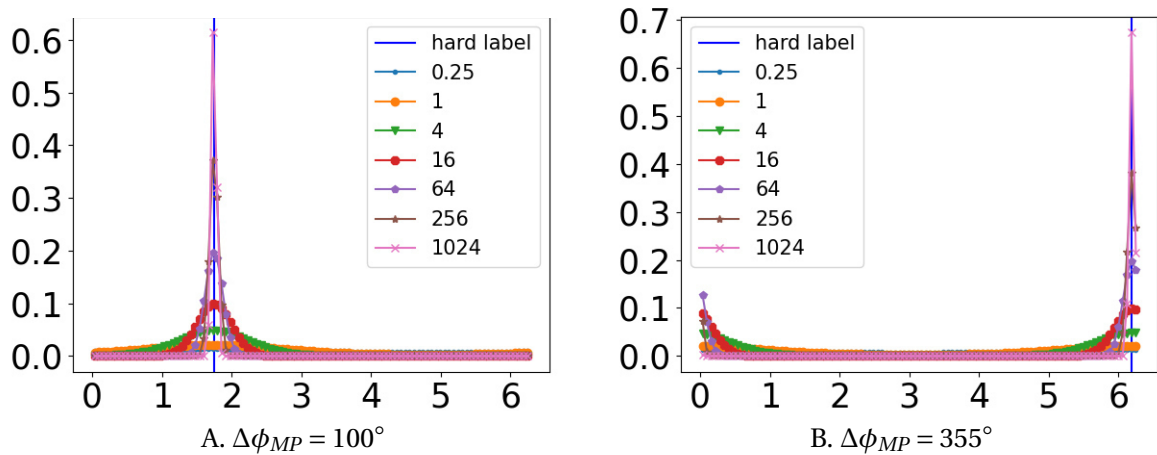


FIGURE 5.12 – Histogrammes en Von Mises construits pour deux *labels* différents dont un à proximité des limites de l’intervalle $[0,360]$. Comme pour la figure 5.11, la valeur de κ affecte la largeur des courbes de distribution ainsi que leur amplitude. L’unité des abscisses est le radian. Le *hard label* correspond à la valeur ϕ_{MP} à partir de laquelle les distributions ont été construites.

5.4. Conclusion

de repousser l'amélioration des performances, avec parfois une division de l'erreur d'estimation par 5 par rapport aux performances d'estimation du chapitre 4. Malgré de meilleures performances pour tous types de multitrajets sur des résolutions d'images différentes et pour des conditions de réception variées, comme dans le chapitre 4, une hausse prononcée de l'erreur est systématiquement observée lors du passage à des images en 10×10 . Il semblerait que l'utilisation du *soft labelling* par *Distributional Loss* ne suffise pas à maintenir des performances satisfaisantes lorsque le nombre de sorties de corrélateur est si réduit. De même, la meilleure adaptation de la loi de Von Mises par rapport à la Gaussienne tronquée pour l'estimation de phase ne suffit pas non plus à conserver des erreurs d'estimation raisonnables dans certains cas de figure : par exemple sur des données Weak MP en 10×10 , l'erreur d'estimation par CNN-HL-VM dépasse les 50° (table 5.4). De manière générale, le gain apporté par la méthode du HL est intéressant pour les paramètres de retard $\Delta\tau_{MP}$ et de phase $\Delta\phi_{MP}$. Pour le décalage en fréquence Doppler Δf_D et l'amplitude du multitrajet $\Delta\alpha_{MP}$, les performances sont quasiment équivalentes entre CNN-Reg et CNN-HL malgré un avantage permanent du CNN-HL.

5.4 Conclusion

Dans ce chapitre, nous avons testé un apprentissage s'appuyant sur la méthode du *Distributional Loss* d'après l'amélioration des performances en régression démontrée par Imani et al. dans [23]. Cette technique, se résumant à la substitution des *labels* sous formes de variables continues par des densités de probabilité, permet d'améliorer les performances de généralisation du CNN pour l'estimation des paramètres des multitrajets à partir d'images de sorties de corrélateur. La tâche de régression ne devient pas une tâche de classification, cependant plutôt que de directement estimer une valeur y d'une donnée x en sortie, le réseau apprend à prédire une distribution de probabilité Q_x sous la forme d'un histogramme. La prédiction de la donnée x par le réseau est calculée à partir de la distribution de prédiction avec la formule de l'équation 5.3. Afin d'appliquer cette technique de *soft labelling*, la structure du CNN a été adaptée avec la modification de la sortie en couche de *softmax* et avec l'implémentation de la fonction perte *Histogram Loss* s'appuyant sur un calcul de divergence de KL. La divergence de KL est un outil permettant de calculer les dissimilarités entre deux distributions de probabilités. Une nouvelle distribution basée sur une loi de Von Mises et plus adaptée pour des paramètres périodiques a été testée sur le paramètre de phase du multitrajet. Dans les expériences effectuées section 5.3.2, une amélioration de la précision d'estimation a été observée pour toutes les résolutions d'images, les conditions de réception et les amplitudes des multitrajets testées. Un double effet d'amélioration a été constaté au niveau de la phase avec une distribution suivant une loi de Von Mises : un premier palier d'amélioration est apporté par l'utilisation de *soft labels* à la place des *labels* initiaux et un deuxième palier d'amélioration est observée par l'utilisation d'une loi de distribution plus adaptée à un paramètre au comportement périodique. Bien qu'améliorant l'erreur d'estimation, les deux méthodes rencontrent des difficultés dans l'estimation d'images 10×10 .

Le prochain chapitre se propose d'appliquer des multitrajets aux paramètres plus réalistes au CNN avec la méthode du *Distributional Loss* afin de mesurer l'évolution des performances avec des images I & Q contenant des multitrajets de paramètres collectés à partir d'un simulateur de propagation de signaux GNSS en environnement 3D.

Chapitre 6

Estimation des paramètres de multitrajets générés par simulation

Sommaire

6.1	Logiciel de propagation de signaux GNSS : SE-Nav	97
6.2	Environnements simulés	98
6.2.1	Toulouse centre - place du Capitole	100
6.2.2	Quartier de Manhattan, New York City, États-Unis d'Amérique	101
6.2.3	Aéroport Roissy Charles-De-Gaulle, Paris Gonesse, France	103
6.3	Régression par CNN-HL	106
6.3.1	Production des images de sorties de corrélateur des multitrajets simulés	106
6.3.2	Résultats	108
6.3.3	Discussion	111
6.4	Conclusion	111

Dans les chapitres 4 et 5 les données utilisées contenaient exclusivement des multitrajets de paramètres aléatoirement générés. Afin de confronter le CNN à des données plus réalistes, des images vont être construites à partir de multitrajets simulés dans un logiciel de propagation de rayons déterministes. Cette modification de la chaîne de génération permettra la création de multitrajets plus réalistes pour un environnement géotypique donné (centre-ville, zone semi-urbaine...) tout en générant une quantité de données suffisante pour l'entraînement du CNN.

Dans ce chapitre, après une courte introduction sur les caractéristiques du logiciel de simulation permettant la collecte de données, nous présenterons les environnements 3D qui ont permis la reproduction des conditions physiques favorables au phénomène de multitrajet. Nous concluons ce chapitre par les expériences de régression du CNN sur ces données simulées.

6.1 Logiciel de propagation de signaux GNSS : SE-Nav

SE-Nav (SE-Nav) est un logiciel de simulation de propagation de signaux GNSS et RF développé par l'entreprise Oktal-SE. Il a pour but de recréer les conditions de propagation de signaux dans des environnements 3D. S'appuyant sur le principe du lancer de rayons, il effectue le calcul de trajectoires de signaux à partir d'équations de l'optique géométrique. La reconstitution de l'environnement 3D local du récepteur permet de reproduire les interactions physiques entre

6.2. Environnements simulés

les signaux émis et la zone environnante du récepteur. Un calcul de bilan de puissance sur le logiciel SE-Nav repose sur trois éléments :

1. Un récepteur. Son évolution dans l'environnement simulé impacte la création des diverses interférences résultantes au cours du temps. Un récepteur sera entièrement caractérisé par sa trajectoire (incluant notamment sa dynamique). Un exemple de trajectoire est illustré figure A.1 dans l'annexe A,
2. Un ou des émetteurs. Dans notre cas d'étude, nous tirons profit de la capacité du logiciel à modéliser une constellation de satellites dans son intégralité. Par le biais de fichiers almanachs, l'orbite exacte des divers satellites de la constellation est calculée ainsi que leur puissance d'émission et leur disponibilité. Cela permet de reproduire la couverture des signaux délivrés par la constellation,
3. Un environnement 3D. Les paramètres du multitrajet sont tributaires des caractéristiques de l'objet réflecteur à partir duquel le multitrajet a été généré. Il est donc nécessaire de reproduire un environnement simulé où les différents matériaux composant celui-ci ont des caractéristiques (coefficients de réflexion de Fresnel) réalistes. L'atténuation atmosphérique et les effets engendrés par la traversée de l'ionosphère sont aussi pris en compte dans SE-Nav au travers des modèles empiriques de l'*International Telecommunication Union* (ITU) [144].

La simulation peut être réalisée sur une durée allant de quelques secondes à quelques minutes et permet la collecte pour chaque pas de temps (défini par l'utilisateur) des caractéristiques des signaux suivants :

- le retard par rapport au LOS $\Delta\tau_{MP}$ (m),
- le décalage de fréquence Doppler par rapport au LOS Δf_D (Hz),
- la puissance de réception P_r (dBW) dont la formule de calcul est détaillée en annexe B,
- la différence de phase par rapport au LOS $\Delta\phi_{MP}$ (rad).

C'est donc à partir de la collecte de ces quatre paramètres que les sorties de corrélateur correspondant aux multitrajets plus réalistes sont construites. Comme illustrés dans la figure 6.1 dans le chapitre 4, les paramètres des multitrajets générés sur SE-Nav alimenteront le générateur de sorties de corrélateur.

6.2 Environnements simulés

Cette section présente les caractéristiques des différents environnements simulés dans lesquels le phénomène de multitrajets a été reproduit. Afin de diversifier les multitrajets collectés, chaque base de données a été construite selon le procédé suivant :

1. Quatre trajectoires totalement différentes ont été empruntées par le récepteur (la figure 6.2 répertorie les 4 différentes trajectoires utilisées dans le scénario aéroportuaire),
2. Pour chacune de ces trajectoires, deux phases de réception ont été réalisées à différents horaires afin d'avoir une configuration de la constellation de satellites différente.

Ainsi 24 différentes simulations ont été nécessaires pour produire les trois *datasets* distincts (8 par par *dataset*). À l'instar de ce qui a été fait en section 2.7 au chapitre 2, ces trois *datasets* ont eux-mêmes été déclinés en trois sous-ensembles de données, où seule la taille des images varie. Les sous-ensembles de données contiennent des images de 80×80, 40×40, 20×20 ou 10×10 sorties de corrélateur.

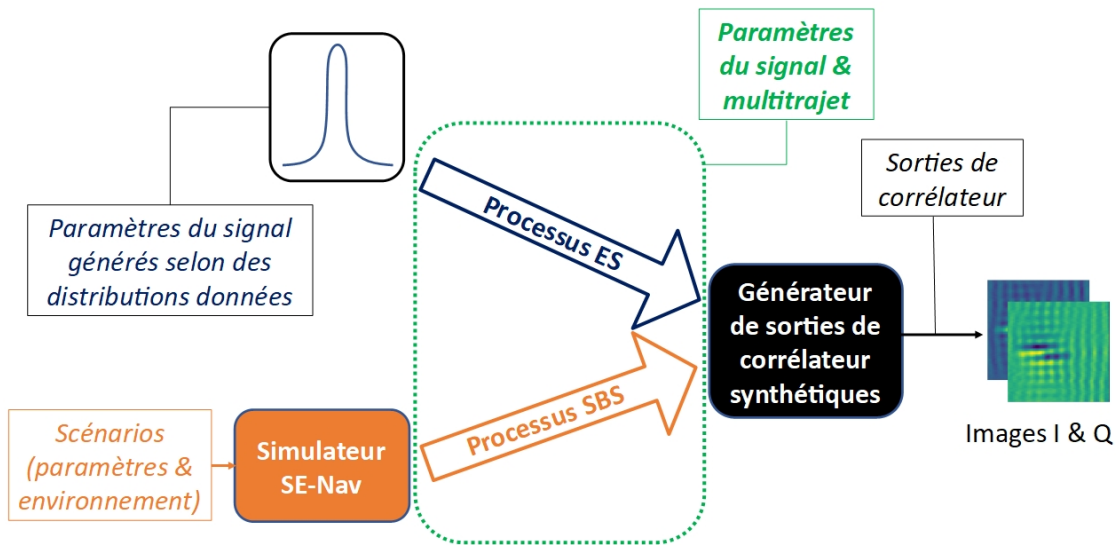


FIGURE 6.1 – Chaîne de génération des données. La branche Entièrement Synthétique (ES) correspond à la génération aléatoire des paramètres de multitrajet des chapitres 4, 5 et 7. La branche Synthétique et Basée sur Scénario (SBS) correspond à l’approche plus réaliste abordée dans ce chapitre (multitrajets générés par le simulateur de propagation SE-Nav).



FIGURE 6.2 – Les quatre différentes trajectoires empruntées par le récepteur pour construire le *dataset* Aéroport Charles-De-Gaulle.



A. Vue aérienne centrée sur le Capitole B. Phénomène de multitrajet dans une ruelle

FIGURE 6.3 – Scénario urbain Toulouse centre - place du Capitole, France. Les rayons blancs correspondent aux signaux LOS en provenance des satellites, les rayons de couleur représentent différents types de multitrajets (rouge pour les réfractions, bleu pour les diffractions).

6.2.1 Toulouse centre - place du Capitole

6.2.1.1 Principales caractéristiques

Parmi les environnements disponibles dans le logiciel SE-Nav, le premier choix s’est porté sur le centre-ville de Toulouse en France. La justification de ce choix s’appuie sur l’architecture dominante du centre urbain. La particularité du centre-ville réside dans la présence d’un grand nombre de ruelles avec des bâtiments d’un maximum de quatre étages, dont le revêtement des façades est principalement composé de briques. Les étroites ruelles sont propices au phénomène de multitrajets comme on peut le constater dans la figure 6.3. Les simulations n’ont pas pris en compte de piétons ou véhicules en mouvement, les multitrajets ont uniquement été générés à partir des interactions avec la structure des bâtiments.

6.2.1.2 Analyse des multitrajets

La collecte des multitrajets selon quatre différentes trajectoires à deux horaires distincts a permis de dresser les histogrammes illustrés Figure 6.4. Il y a une majorité de faibles retards dans les multitrajets collectés ainsi qu’un grand nombre de multitrajets à faibles amplitudes. Ces observations diffèrent des hypothèses qui ont servi à la génération des multitrajets aléatoires dans la section 2.3 où le paramètre $\Delta\tau_{MP}$ suivait une loi de génération uniforme sur l’intervalle $[0, 1.5 T_c]$. La distribution des fréquences Doppler ressemble à une loi normale, avec cependant une différence au niveau de l’écart-type constaté par rapport à celui choisi en hypothèse section 2.3. On remarque un effet de bord en figure 6.4.C où toutes les valeurs de $\alpha_{MP} \geq 1$ ont été ramenées à 1.

Les figures 6.5 exposent différents nuages de points construits à partir de la combinaison des paramètres du multitrajet. Par exemple d’après les figures 6.5.B et D les multitrajets de fortes amplitudes collectés sont généralement liés à un retard et une fréquence Doppler faibles. De même les amplitudes maximales des fréquences Doppler observées sont atteintes pour des retards importants d’après la figure 6.5.A. La figure 6.5.E montre une forte concentration de multitrajets de faibles décalages en fréquence Doppler sans avoir d’influence évidente sur la répartition des phases du multitrajet. De même on observe une forte concentration de multitrajets atténués (faibles α_{MP}) mais à nouveau sans conséquence évidente sur la répartition des phases dans la figure 6.5.F).

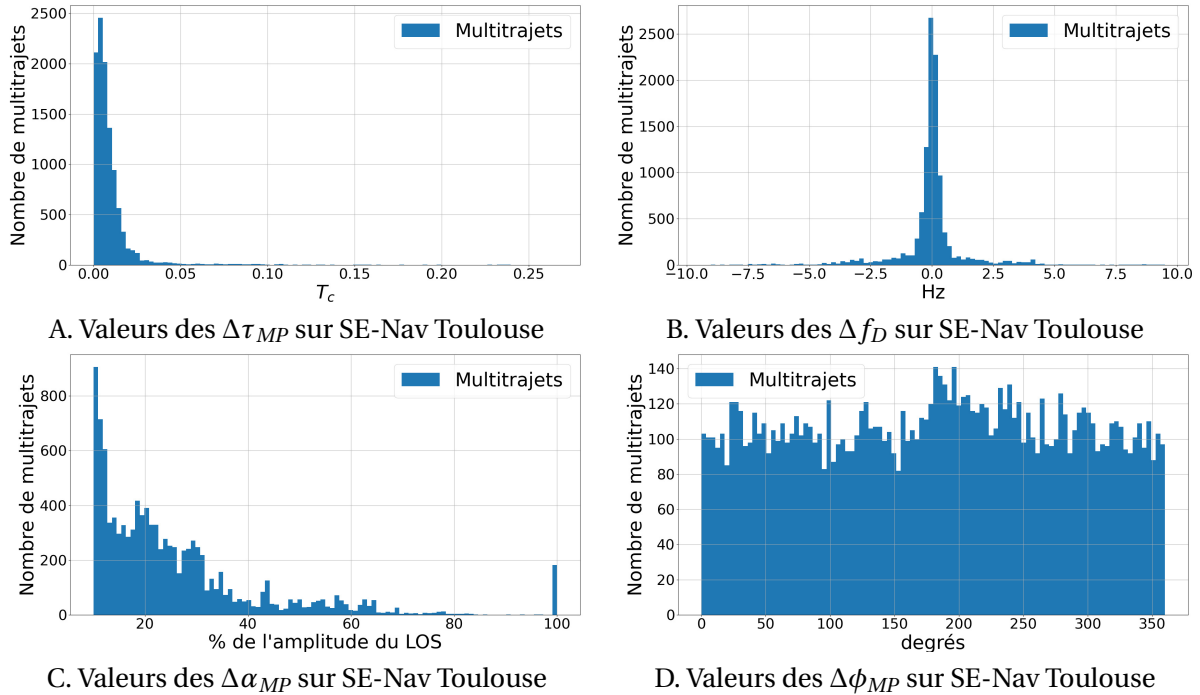


FIGURE 6.4 – Distribution des paramètres de multitrajets collectés au sein de l’environnement 3D simulé du centre-ville de Toulouse utilisés dans le générateur d’images (10 722 multitrajets).

6.2.2 Quartier de Manhattan, New York City, États-Unis d’Amérique

6.2.2.1 Principales caractéristiques

Contrairement à Toulouse, le quartier de Manhattan à *New York City* (NYC) est caractérisé par la présence de gratte-ciels (figure 6.6) avec un revêtement alternant verre et métal. Ces derniers attirent particulièrement l’attention car le métal offre une réflexion considérée comme parfaite dans le simulateur et permet de renvoyer des multitrajets de forte amplitude vers le récepteur. En opposition avec le cas du centre urbain de Toulouse, la ville de Manhattan possède une géométrie de rues suivant un plan hippodaméen avec un maillage régulier des rues se croisant presque exclusivement en angles droits. Comme pour le centre-ville toulousain, les multitrajets ont uniquement été générés à partir des interactions physiques entre les signaux satellites et les bâtiments environnants, sans l’implantation de piétons ou véhicules en mouvement lors des différentes simulations.

6.2.2.2 Analyse des multitrajets

La figure 6.7 permet d’observer la collecte de multitrajets de forte intensité (plus de 6000 multitrajets ont un $\alpha_{MP} \approx 1$ sur 30 676, soit environ 20% des multitrajets ; dans le scénario du centre-ville de Toulouse, seulement 2% des multitrajets présentent une amplitude aussi élevée) quand ils sont générés dans les canyons urbains de Manhattan. On remarque également le même phénomène d’effet de bord déjà présent en figure 6.4.C dans la figure 6.7.C. Autre fait observable, il y a une non-uniformité de la distribution de la phase au cours des différentes simulations.

Les figures 6.8.A et B permettent de constater une majorité de multitrajets de faibles retards ainsi qu’une minorité de fréquences Doppler de forte amplitude. On observe également une

6.2. Environnements simulés

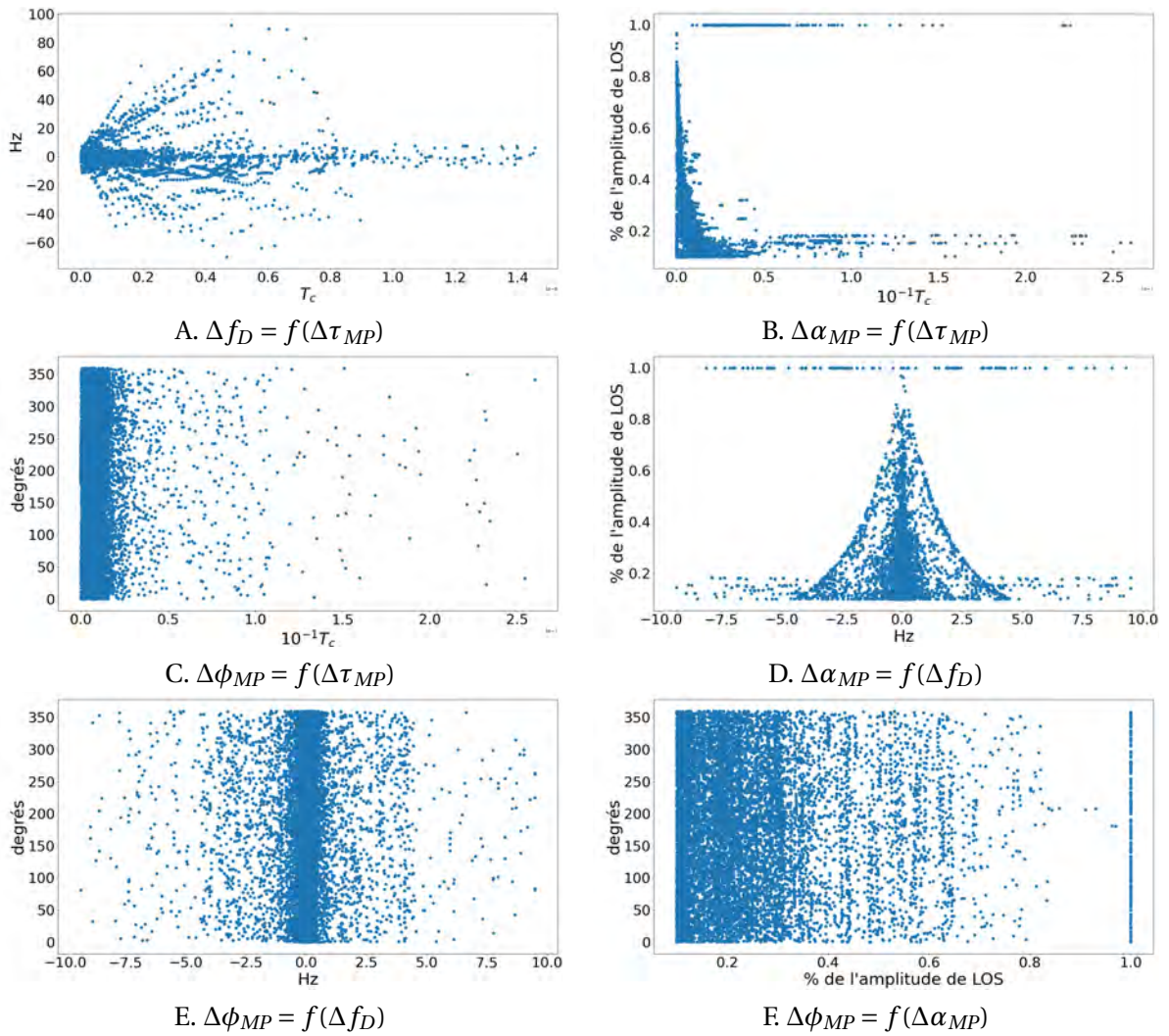
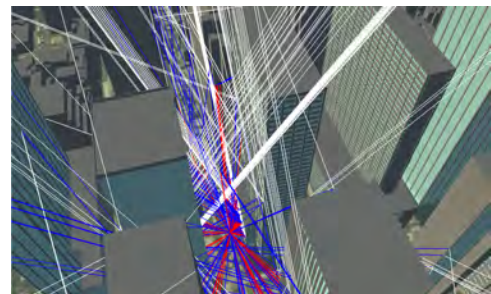
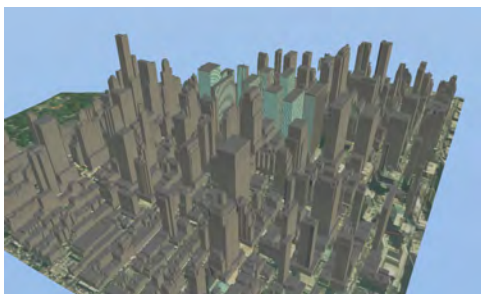


FIGURE 6.5 – Nuages de points des différents paramètres de multitrajets générés à partir des simulations au sein de l’environnement 3D simulé du centre-ville de Toulouse.



A. Vue aérienne de la maquette de Manhattan B. Phénomène de multitrajet dans un canyon urbain

FIGURE 6.6 – Scénario urbain Manhattan à NYC, États-Unis d’Amérique.

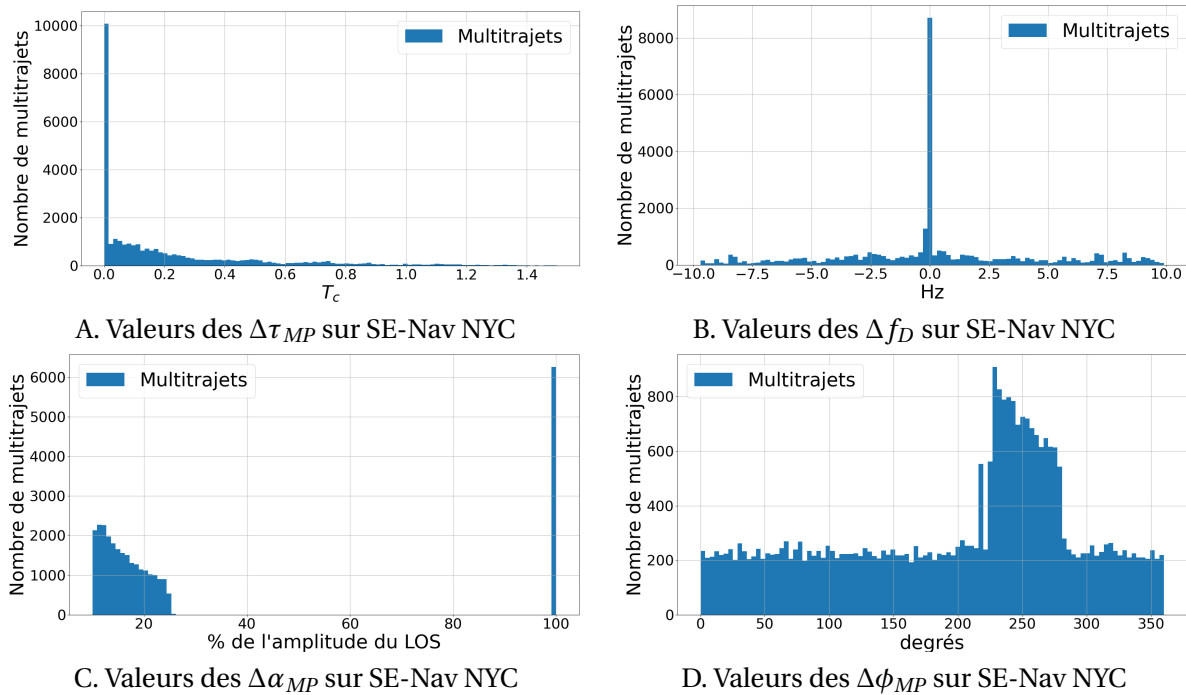


FIGURE 6.7 – Distribution des paramètres de multitrajets collectés au sein de l’environnement 3D simulé du quartier de Manhattan utilisés dans le générateur d’images (30 676 multitrajets).

dichotomie au niveau des amplitudes de multitrajets figures 6.8.B, D et F : un groupe majeur avec des multitrajets de faibles amplitudes et un groupe mineur composé de multitrajets de très fortes amplitudes (allant jusqu’à atteindre une amplitude similaire à celle du LOS). Comme pour la figure 6.5.B dans la section précédente, la figure 6.8.B montre un lien entre faible retard et atténuation du multitrajet. De même que pour les nuages de points du scénario SE-Nav Toulouse, les fréquences Doppler de fortes amplitudes concernent dans la globalité les multitrajets de retards conséquents.

6.2.3 Aéroport Roissy Charles-De-Gaulle, Paris Gonesse, France

6.2.3.1 Principales caractéristiques

Les deux premiers scénarios étaient caractérisés par leur potentiel de génération de multitrajets avec la présence de nombreux canyons urbains. Cependant ces scénarios ne possédaient pas d’objets mobiles de grande cinétique et les fréquences Doppler récoltées avaient un écart-type de quelques Hertz. Or la fréquence Doppler d’un multitrajet dépend aussi de la vitesse de l’objet réfléchissant. Le scénario aéroportuaire a été construit de façon à favoriser des décalages plus importants au niveau des fréquences Doppler avec la réflexion sur des objets à forte cinétique (avion en mouvement comme illustré figure 6.9.B) et offrant aussi une réflexion quasi-parfaite (avion considéré en métal). Pour ce scénario en particulier, une des trajectoires empruntées par le récepteur correspondra à celle d’un avion civil avec une phase de roulage, de décollage et de survol à basse altitude, incluant ainsi des phases de vol à faible et haute vitesses, de forte accélération et décélération.

L’aéroport international de Roissy-Charles-De-Gaulle (CDG), situé à Gonesse en France, est caractérisé dans SE-Nav par des bâtiments environnants peu nombreux et d’altitude basse (dans

6.2. Environnements simulés

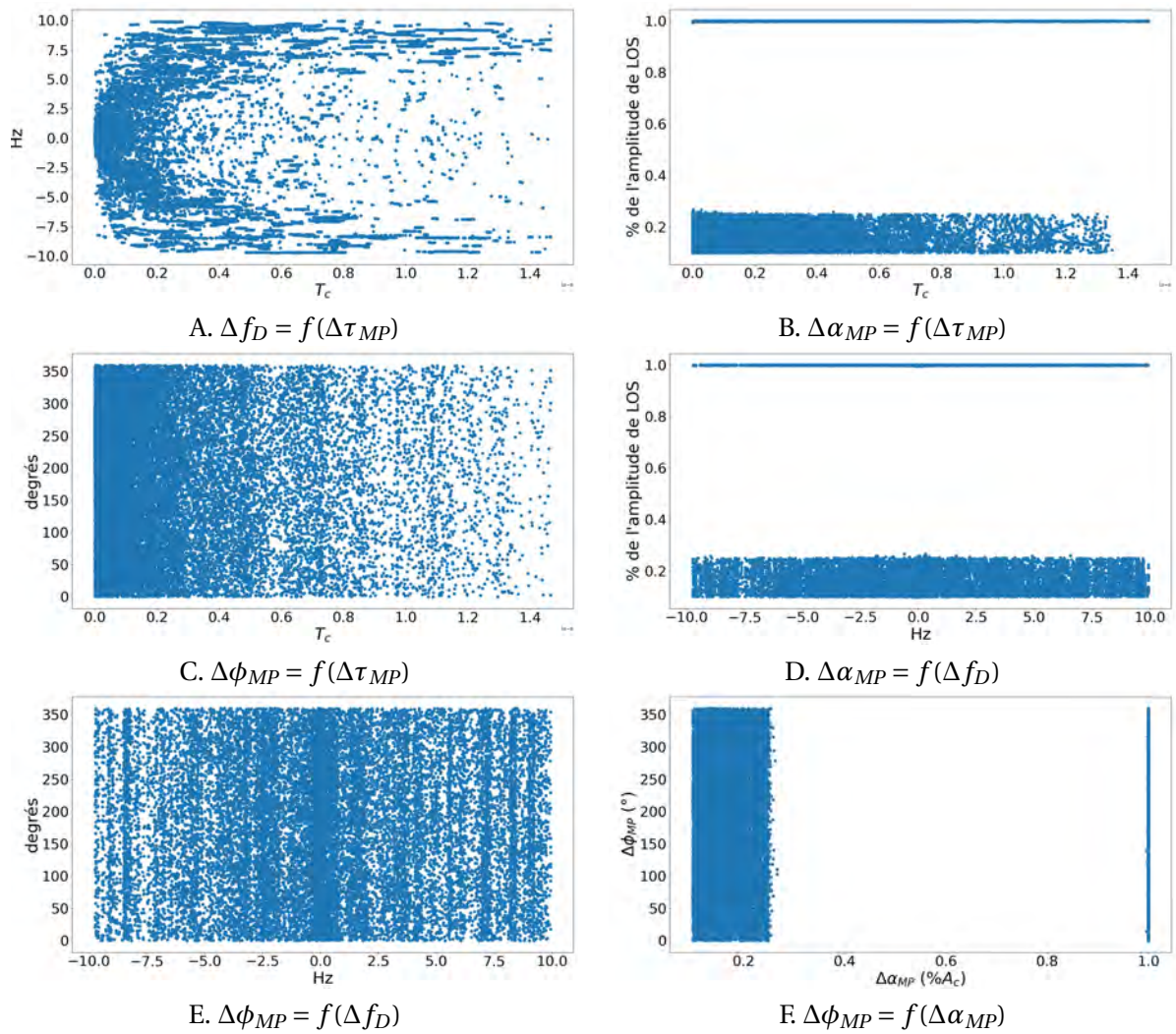


FIGURE 6.8 – Nuages de points des différents paramètres de multitrajets récoltés à partir des simulations au sein de l’environnement 3D simulé des quartiers de Manhattan.



A. Vue aérienne de l’aéroport CDG



B. Multitrajet en provenance d’un avion

FIGURE 6.9 – Scénario rural avec mobiles centré sur l’aéroport Roissy-CDG, Gonesse, France.

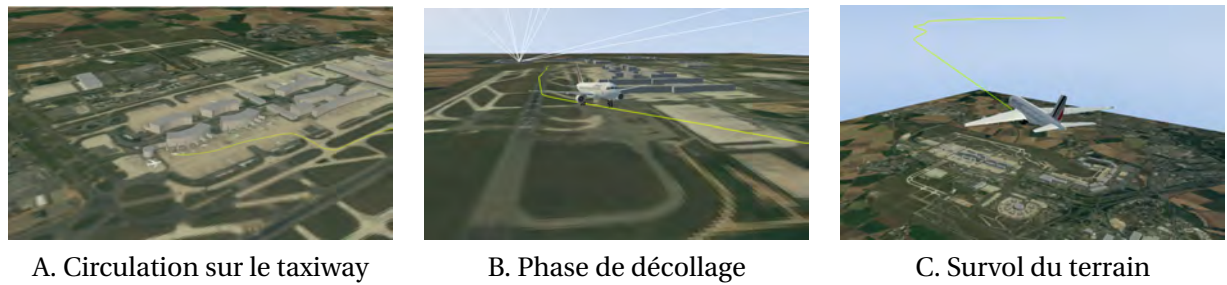


FIGURE 6.10 – Différentes trajectoires empruntées.

ce scénario, les deux bâtiments les plus hauts étant des tours de contrôle). Au contraire des deux premiers scénarios, les simulations au sein de l'environnement aéroportuaire se sont déroulées dans un cadre rural. L'autre principale différence réside dans la présence d'avions autres que le porteur du récepteur et animés par trois types de trajectoires illustrées figure 6.10 :

- Trajectoire au sol (A) : le premier type de trajectoires allouées aux avions est un déplacement au sol sur les taxiways. Les taxiways sont des routes empruntées par les avions et leur permettant de transiter d'une piste d'atterrissage à un terminal d'embarquement. La vitesse des avions suivant ces trajectoires n'excède pas 5 m.s^{-1} et ces mobiles restent au niveau du sol,
- Trajectoire de décollage/atterrissage (B) : les avions vont simuler des phases de décollage et d'atterrissage, accompagnées de variations de vitesse (de 1 à 100 m.s^{-1}) et d'altitudes (0 à 300 m),
- Trajectoire de survol (C) : dans ce cas précis, les avions survolent les pistes/terminaux de l'aéroport. C'est dans cette configuration qu'ils seront animés des plus fortes cinétiques (jusqu'à 200 m.s^{-1}) et évolueront à des altitudes variant de 150 m à 4 500 m).

6.2.3.2 Analyse des multitrajets

Comme pour les deux environnements simulés précédents, la majorité des multitrajets collectés ont une valeur de retard faible et un décalage en fréquence Doppler proche de zéro (bien que ce scénario soit celui contenant les décalages en fréquences Doppler les plus extrêmes). On remarque également dans la figure 6.11.C trois groupes de multitrajets. Deux groupes majeurs sont composés de multitrajets de faibles et moyennes amplitudes tandis qu'un dernier groupe est constitué de quelques multitrajets intenses (d'amplitudes α_{MP} s'approchant des forts multitrajets collectés sur le scénario SE-Nav NYC). Comme pour les multitrajets du scénario Manhattan, la phase des différents multitrajets collectés n'est pas uniformément répartie sur l'intervalle $[0, 2\pi]$ (figre 6.11.D. Cette subdivision est à nouveau observée dans la figure 6.12.F où il y a une répartition uniforme de la phase quelle que soit l'amplitude des multitrajets. Sur ces mêmes figures, on constate une distribution ressemblant à une répartition gaussienne des fréquences Doppler (figure 6.12.D) ainsi qu'une atténuation des multitrajets inversement proportionnelle au retard de multitrajet.

6.3. Régression par CNN-HL

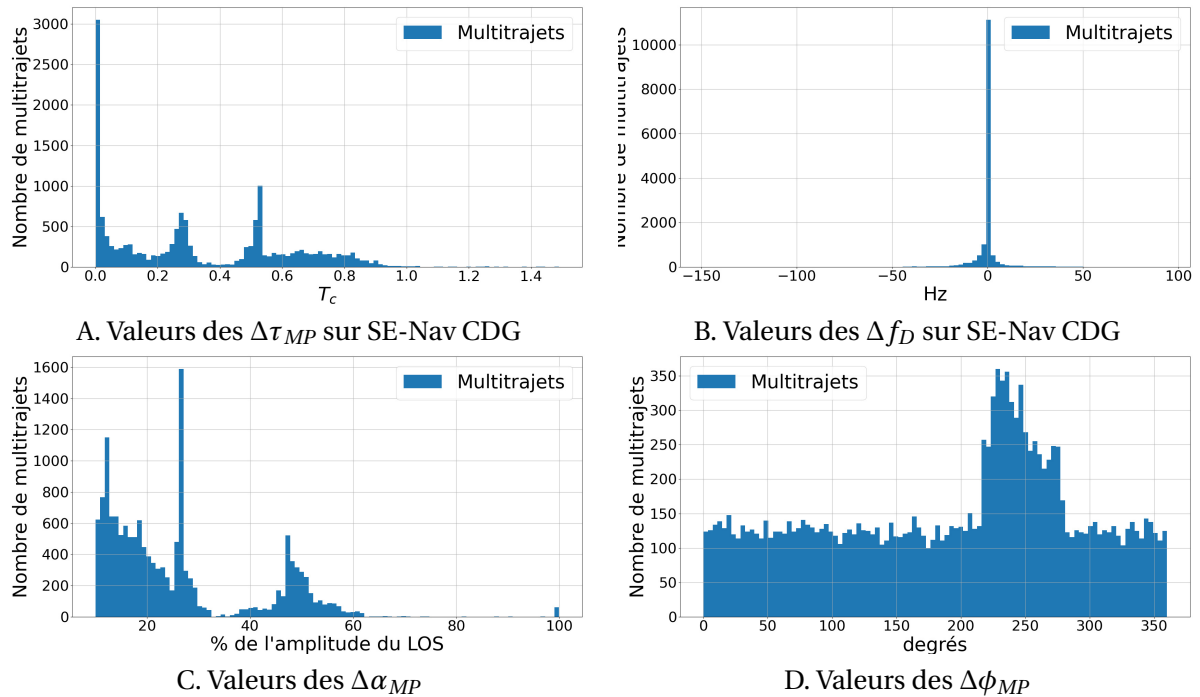


FIGURE 6.11 – Distribution des paramètres de multitrajets collectés au sein de l'environnement 3D simulé de Roissy-CDG (15 215 multitrajets).

6.3 Régression par CNN-HL

6.3.1 Production des images de sorties de corrélateur des multitrajets simulés

À partir des environnements simulés, nous avons produit des couples d'images I & Q représentant les sorties de corrélateur correspondant aux multitrajets simulés. Ces données plus réalistes ont été utilisées pour l'entraînement d'un CNN-HL, utilisant la technique du *Distributional Loss* introduit au chapitre précédent. Dans cette section, les résultats d'estimation sont comparés selon quatre ensembles de données différents. L'ensemble Random MP est le *dataset* contenant des multitrajets directement générés à partir de variables aléatoires pour un C/N_0 fixé à 40 dBHz. Les autres ensembles de données correspondent aux environnements simulés introduits en section 6.2. Le but de ces expériences étant l'observation des différences de performances entre données aléatoires et simulées ainsi qu'entre des environnements de simulation distincts.

Pour réunir les données nécessaires à la construction des ensembles de données simulées, les multitrajets sont sélectionnés selon leur retard, afin que celui-ci n'excède pas 1.5 T_c . Comme rappelé en section 2.3, un multitrajet d'un tel retard n'impacte plus les sorties E-P-L du corrélateur : les signaux correspondants sont ainsi écartés. 10 000 multitrajets sont aléatoirement sélectionnés parmi ceux restant afin de former l'ensemble de données utilisé par le CNN.

Remarque : Les images I & Q construites à partir des paramètres de multitrajets simulés ont toutes été générées pour un niveau de C/N_0 fixé à 40 dBHz.

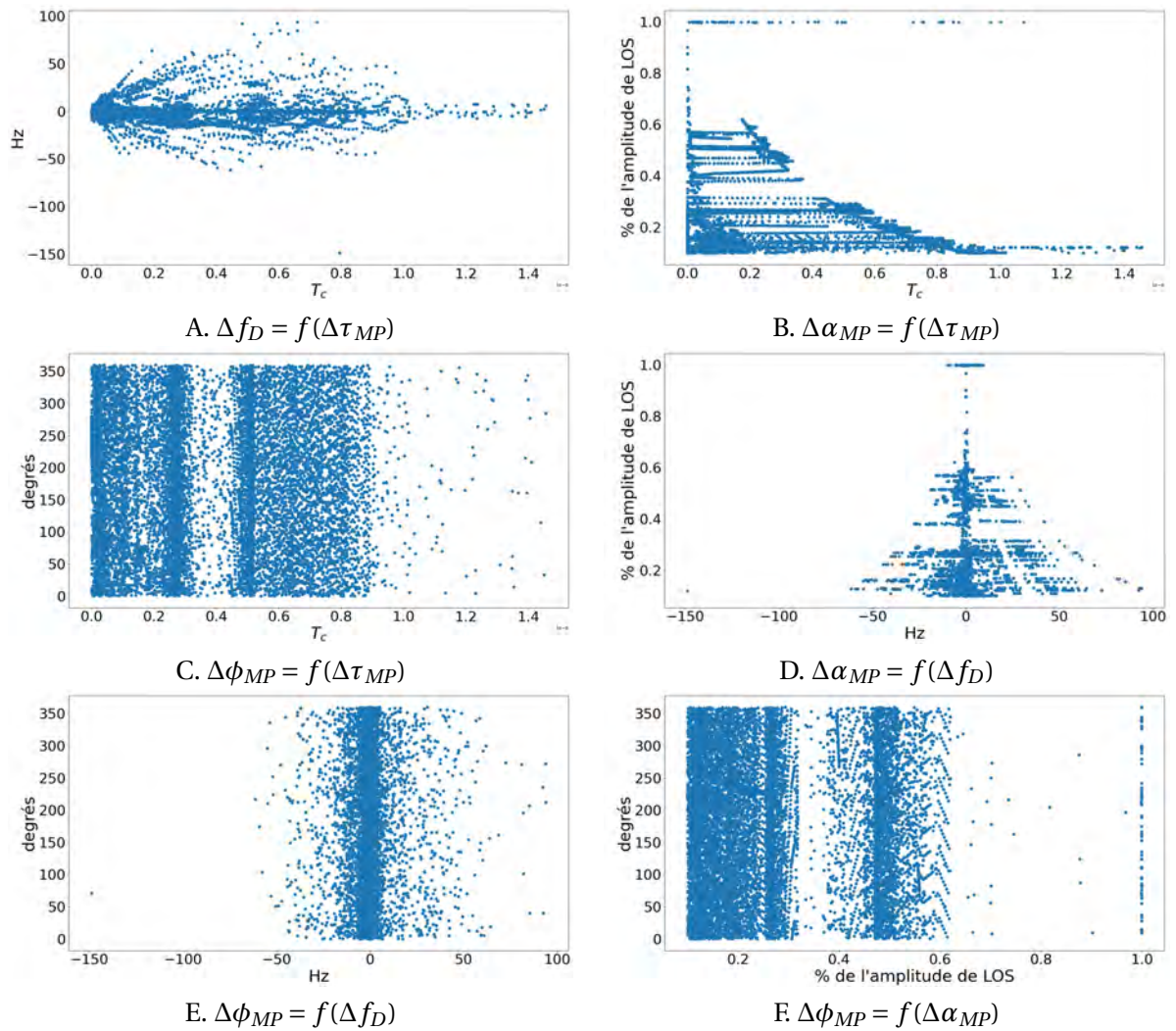


FIGURE 6.12 – Nuages de points des différents paramètres de multitrajets récoltés à partir des simulations au sein de l’environnement 3D simulé de Roissy-CDG.

6.3.2 Résultats

Le processus expérimental est le même qu’aux sections 4.4.1, 4.4.2 et 5.3.2 : par scénario de simulation, les *datasets* utilisés par le CNN-HL sont composés de 10 000 échantillons. Dix phases d’apprentissage et de généralisation différentes sont effectuées pour chaque scénario. Les performances d’estimation effectuée dans les différents environnements simulés s’appuient sur la moyenne des MAE sur dix expériences (calculs détaillés à la section 5.3.1). Les tables 6.1, 6.2, 6.3 et 6.4 renseignent les performances du CNN-HL sur l’estimation des quatre paramètres de multitrajets. Les dimensions des images prendront les valeurs suivantes : 80×80, 40×40, 20×20 et 10×10 pixels afin de mesurer l’impact de la résolution des images sur l’estimation à partir de données issues du simulateur.

Dans la table 6.1, on constate pour la majorité des multitrajets qu’une diminution du nombre de pixels des images I & Q provoque une hausse de l’erreur d’estimation du CNN-HL. L’augmentation de cette erreur explose lorsque les images passent de 20×20 à 10×10 pixels : l’erreur d’estimation est multipliée par un facteur trois concernant les multitrajets récoltés des environnements de NYC (de $2.69 \cdot 10^{-2} T_c$ à $8.17 \cdot 10^{-2} T_c$) et CDG (de $2.08 \cdot 10^{-2} T_c$ à $7.94 \cdot 10^{-2} T_c$). L’erreur est multipliée par un facteur supérieur à cinq pour les multitrajets non issus du simulateur ($2.21 \cdot 10^{-2} T_c$ à $1.21 \cdot 10^{-1} T_c$). Quant aux multitrajets générés à partir du scénario Toulouse, les performances restent constantes et la taille des images ne semble pas avoir de conséquences évidentes sur ces données en particulier.

TABLE 6.1 – Erreur moyenne d’estimation du paramètre $\Delta\tau_{MP}$ sur les différents ensembles de données (unité de la table : $10^{-2} T_c$).

Ensemble de données	Taille des images	CNN-HL
Multitrajets aléatoirement générés	80×80	0.81 ± 0.11
	40×40	1.20 ± 0.21
	20×20	2.21 ± 0.29
	10×10	12.1 ± 0.59
SE-Nav Toulouse	80×80	0.64 ± 0.05
	40×40	0.71 ± 0.07
	20×20	0.62 ± 0.06
	10×10	0.73 ± 0.12
SE-Nav New York City	80×80	1.14 ± 0.19
	40×40	1.53 ± 0.31
	20×20	2.69 ± 0.36
	10×10	8.17 ± 0.81
SE-Nav Aéroport Charles-De-Gaulle	80×80	1.14 ± 0.14
	40×40	1.35 ± 0.17
	20×20	2.08 ± 0.31
	10×10	7.94 ± 0.40

Dans la table 6.2, comme pour la table 6.1 les ordres de grandeur observés sont similaires entre les erreurs d’estimation sur multitrajets non issus du simulateur et ceux générés par celui-ci. Pour les ensembles de données SE-Nav NYC et CDG, les performances sont quasiment équivalentes sur des images en 80×80 et 40×40 pixels. Là où l’erreur d’estimation s’accroît fortement pour les multitrajets aléatoirement générés (de 0.95 Hz à 9.71 Hz, soit une erreur d’estimation décuplée entre des images 80×80 et 10×10), les estimations effectuées sur les données issues du simulateur montrent une augmentation progressive de l’erreur avec une diminution du

nombre de sorties de corrélateur utilisées. Par exemple, l'erreur est multipliée par deux entre des images 80×80 et des images 10×10 pour le scénario SE-Nav CDG (la moyenne de l'erreur est comprise entre 1.03 ± 0.14 Hz et 2.75 ± 0.18 Hz). Les résultats issus des données simulateur sont moins impactés par un passage d'images en 20×20 à 10×10 pixels. À titre d'exemple, pour les scénarios NYC et CDG, l'erreur augmente entre 70% et 80% lors du passage d'images de taille 20×20 à 10×10 alors que l'erreur augmente d'un facteur quatre pour les multitrajets aléatoirement générés.

TABLE 6.2 – Erreur moyenne d'estimation du paramètre Δf_D sur les différents ensembles de données (unité de la table : Hz).

Ensemble de données	Taille des images	CNN-HL
Multitrajets aléatoirement générés	80×80	0.95 ± 0.08
	40×40	1.10 ± 0.13
	20×20	2.10 ± 0.28
	10×10	9.71 ± 0.47
SE-Nav Toulouse	80×80	0.79 ± 0.04
	40×40	0.74 ± 0.06
	20×20	0.72 ± 0.05
	10×10	0.72 ± 0.03
SE-Nav New York City	80×80	0.82 ± 0.11
	40×40	0.85 ± 0.13
	20×20	1.29 ± 0.09
	10×10	2.39 ± 0.10
SE-Nav Aéroport Charles-De-Gaulle	80×80	1.03 ± 0.14
	40×40	1.06 ± 0.12
	20×20	1.59 ± 0.20
	10×10	2.75 ± 0.18

Dans la table 6.3, les erreurs d'estimation sont inférieures à 2.21% de l'amplitude du LOS pour des images de taille 80×80 à 20×20. Le passage à des images en 10×10 impacte de manière plus significative l'estimation des multitrajets aléatoires par rapport à ceux extraits des simulations (erreur triplée). On relève également, pour les images I & Q en 10×10, une erreur deux fois moins élevée pour l'estimation de l'amplitude de multitrajets en provenance de SE-Nav par rapport à ceux aléatoirement générés.

L'erreur d'estimation de $\Delta\phi_{MP}$ à la table 6.4 reste comprise entre 4° et 7°, avec une augmentation manifeste de celle-ci lorsque le nombre de pixels sur image est réduit à du 10×10 (l'erreur va du double au quadruple). Comme dans la table 6.3, le passage à des images en 10×10 dé-

6.3. Régression par CNN-HL

térieure plus significativement les performances sur les données synthétiques par rapport aux données simulées.

TABLE 6.3 – Erreur moyenne d'estimation du paramètre $\Delta\alpha_{MP}$ sur les différents ensembles de données (unité de la table : % de l'amplitude de LOS).

Ensemble de données	Taille des images	CNN-HL
Multitrajets aléatoirement générés	80×80	1.45 ± 0.19
	40×40	1.60 ± 0.14
	20×20	2.21 ± 0.28
	10×10	6.84 ± 0.12
SE-Nav Toulouse	80×80	1.02 ± 0.14
	40×40	1.14 ± 0.11
	20×20	1.59 ± 0.26
	10×10	3.68 ± 0.17
SE-Nav New York City	80×80	1.12 ± 0.34
	40×40	0.86 ± 0.22
	20×20	1.08 ± 0.14
	10×10	2.62 ± 0.25
SE-Nav Aéroport Charles-De-Gaulle	80×80	1.43 ± 0.44
	40×40	1.26 ± 0.19
	20×20	1.50 ± 0.17
	10×10	3.58 ± 0.13

TABLE 6.4 – Erreur moyenne d'estimation du paramètre $\Delta\phi_{MP}$ sur les différents ensembles de données (unité de la table : degrés).

Ensemble de données	Taille des images	CNN-HL
Multitrajets aléatoirement générés	80×80	5.07 ± 0.69
	40×40	5.43 ± 0.49
	20×20	6.63 ± 0.40
	10×10	28.05 ± 1.41
SE-Nav Toulouse	80×80	5.13 ± 1.96
	40×40	4.24 ± 0.59
	20×20	4.36 ± 0.57
	10×10	9.89 ± 0.58
SE-Nav New York City	80×80	5.54 ± 0.85
	40×40	5.37 ± 0.82
	20×20	6.30 ± 0.59
	10×10	15.57 ± 0.44
SE-Nav Aéroport Charles-De-Gaulle	80×80	6.57 ± 1.27
	40×40	6.69 ± 1.35
	20×20	6.35 ± 0.74
	10×10	16.70 ± 1.07

6.3.3 Discussion

Les tables 6.1, 6.2 et 6.3 montrent une augmentation progressive de l'erreur d'estimation avec la diminution de la résolution des images. L'accroissement de l'erreur est plus prononcée lors du passage d'images 20×20 à 10×10 pixels qu'à une quelconque autre diminution de la taille d'image. Par exemple de 20×20 à 10×10 pixels, on peut observer un triplement de l'erreur sur l'estimation de $\Delta\tau_{MP}$ sur les données NYC et CDG. Ce phénomène est d'autant plus flagrant pour l'estimation de la phase dans la table 6.4 où les performances sont quasi-constantes sur des images 80×80 à 20×20 avant de s'écrouler en passant à du 10×10 . Les performances sur les multitrajets issus du scénario Toulouse ont un comportement différent des autres scénarios où les variations d'erreur d'estimation du retard et de la fréquence Doppler sont négligeables avec la détérioration de la qualité des images (tables 6.1 et 6.2). Dans la table 6.1 l'erreur d'estimation observée sur les multitrajets collectés dans le scénario Toulouse est comprise dans l'intervalle $[0.62 \cdot 10^{-2} T_c, 0.73 \cdot 10^{-2} T_c]$. La figure 6.4.A. indique une moyenne des retards de multitrajet issus de ce scénario à $0.01 T_c$. Ce qui signifie que sur les expériences menées table 6.1, il y a une différence en moyenne de 62% entre la valeur vraie (*label*) du retard du multitrajet et celle estimée par le CNN-HL. Ainsi pour ce scénario en particulier, on peut voir de ces résultats que l'estimation du retard a rencontré des difficultés. Au contraire la différence entre la valeur vraie et la prédiction des multitrajets issus du scénario NYC (en moyenne de retard $0.221 T_c$ d'après la figure 6.7.A.) débute à 5% avec des images en 80×80 et plafonne à 36% avec des images en 10×10 .

6.4 Conclusion

L'objectif de ce chapitre était double avec la génération de multitrajets plus réalistes et l'estimation de leurs paramètres par CNN-HL. L'utilisation du logiciel de simulation de propagation en environnements simulés a permis de compléter le premier objectif. On a ainsi pu observer les différentes distributions de paramètres au travers de trois scénarios distincts et les confronter à nos hypothèses en section 2.3. Cela a permis de mettre en lumière les différences suivantes :

- Les retards ne sont pas uniformément répartis mais semblent suivre une loi exponentielle,
- Les fréquences Doppler semblent suivre une loi gaussienne centrée avec un écart-type plus ou moins important en fonction de l'environnement de génération,
- L'atténuation n'est pas uniformément distribuée selon l'intervalle $[0.1, 0.9]$,
 - Des valeurs en dehors de l'intervalle $[0.1, 0.9]$ ont été détectées, avec une majorité de multitrajets en dehors de cet intervalle d'une amplitude inférieure à 0.1,
 - La répartition de ce paramètre est corrélée à son environnement simulé d'origine. Pour les données SE-Nav Toulouse, on remarque sur la figure 6.4.C une distribution analogue à une loi exponentielle avec des multitrajets de fortes intensités (se rapprochant de l'amplitude du LOS). Pour les données de SE-Nav NYC, les multitrajets collectés forment deux groupes distincts : les multitrajets d'amplitudes faibles et modérés et les multitrajets de fortes amplitudes. Enfin pour le scénario SE-Nav CDG, les interférences collectées permettent de définir trois groupes différents d'amplitude,
- La phase n'est pas toujours répartie de manière uniforme (ex : figure 6.7.D),
- Nos générations aléatoires de paramètres (section 2.3) étaient indépendantes et n'influaient pas sur la valeur des autres paramètres. Les nuages de points construits à partir des simulations ont montré que des liens semblent exister entre différents paramètres. Les faibles retards sont souvent associés à des multitrajets de hautes intensités. De même des décalages

6.4. Conclusion

en fréquence Doppler élevés étaient synonymes de forts retards et d'amplitudes faibles (les fréquences Doppler sont élevées pour des retards élevés car généralement les mobiles se déplaçant à une grande vitesse ne rasant pas le récepteur).

Les différentes expériences ont mis en avant la robustesse de l'algorithme d'estimation CNN-HL pour des images de multitrajets physiquement réalistes pour des dimensions comprises entre 80×80 et 20×20 pixels. Les expériences ont également montré un accroissement important de l'erreur d'estimation lors du travail avec des images en 10×10 pixels. Les ordres de grandeur des erreurs d'estimation entre données purement aléatoires et simulées étaient similaires pour des images 80×80 , 40×40 et 20×20 pixels. Cependant une différence apparaissait lors du passage à 10×10 pixels, montrant une difficulté plus prononcée dans l'estimation de données non réalistes avec moins de sorties de corrélateur. Il est également à noter que toutes les tentatives d'estimation ne se sont pas soldées en succès, avec une erreur non négligeable (de l'ordre de 60% de la valeur vraie) quant à l'estimation du retard sur des données SE-Nav Toulouse, quel que soit le nombre de sorties de corrélateur utilisées (table 6.1). On en conclut que dans la majorité des cas, les performances d'estimation par CNN-HL sont satisfaisantes avec des images de tailles comprises entre 80×80 et 20×20 pixels, avec des performances similaires entre données aléatoires et simulées.

Chapitre 7

Perspectives

Sommaire

7.1 Élimination du multitrajet	113
7.2 Estimation de paramètres d'un seul multitrajet dans un environnement multi-interféré	114
7.3 Estimation des paramètres de plusieurs multitrajets en environnement multi-interféré	119
7.3.1 Régression de multi multitrajets : approche itérative	119
7.3.2 Régression de multi multitrajets : approche en simultané	119
7.4 Améliorations	121
7.5 Perspectives à plus long terme	122

Ce chapitre a pour but d'exposer des travaux préliminaires sur la régression multi-interférences ainsi qu'étudier les perspectives dans les domaines du GNSS et du *Deep Learning* suite aux travaux de la thèse. Ce chapitre est divisé en plusieurs sections. La première section propose un moyen de suppression d'un unique multitrajet sur les images de sorties de corrélateur avec plusieurs idées de validation du résultat. La deuxième section présente les travaux préliminaires de l'estimation d'un multitrajet dans un environnement avec plusieurs multitrajets, testant ainsi la robustesse du CNN et notamment de la méthode du *Distributional Loss* face à l'ajout d'interférences. La troisième section explore des perspectives concernant l'estimation de $M \in \mathbb{N}^*$ multitrajets (les chapitres 2 à 6 ayant considéré un seul multitrajet par image). La quatrième et dernière section apportera des critiques et points d'amélioration sur différents choix expérimentaux de la thèse : la pertinence de l'utilisation de la divergence de *Kullback-Leibler* pour le calcul de dissimilarité entre deux distributions, le choix d'une autre architecture réseau, un calcul différent des sorties de corrélateur. . .

7.1 Élimination du multitrajet

Suite aux résultats d'estimation par régression des paramètres de multitrajets obtenus dans les chapitres 4, 5 et 6, les prochains travaux devront porter sur l'élimination de la composante du multitrajet dans les images de sorties de corrélateur. Pour rappel, dans les expériences précédentes, nous avons considéré que les images I & Q étaient construites comme suit :

$$I = I_{LOS} + I_{MP}(\Delta\tau_{MP}, \Delta f_D, \Delta\alpha_{MP}, \Delta\phi_{MP}) + b_I$$

7.2. Estimation de paramètres d'un seul multitrajet dans un environnem

$$Q = Q_{LOS} + Q_{MP}(\Delta\tau_{MP}, \Delta f_D, \Delta\alpha_{MP}, \Delta\phi_{MP}) + b_Q$$

avec

- I_{LOS} & Q_{LOS} étant respectivement les images de sortie du corrélateur des canaux I & Q,
- $I_{MP}(\Delta\tau_{MP}, \Delta f_D, \Delta\alpha_{MP}, \Delta\phi_{MP})$ & $Q_{MP}(\Delta\tau_{MP}, \Delta f_D, \Delta\alpha_{MP}, \Delta\phi_{MP})$ correspondant aux composantes du multitrajet avec ses paramètres caractéristiques,
- b_I et b_Q les composantes de bruit gaussien sur I & Q respectivement.

La suppression du multitrajet s'appuie sur la linéarité de la chaîne de réception par rapport aux composantes du signal. Dans le cas d'une réception d'un multitrajet en présence du signal LOS, la sortie du corrélateur peut être considérée comme la somme entre la sortie de corrélateur correspondant au multitrajet et celle correspondant au signal LOS [21]. Avec les estimations du quadruplet de paramètres $(\Delta\tilde{\tau}_{MP}, \Delta\tilde{f}_D, \Delta\tilde{\alpha}_{MP}, \Delta\tilde{\phi}_{MP})$, nous sommes en mesure de reconstruire les images de sorties de corrélateur correspondant au multitrajet estimé seul (c'est-à-dire sans signal LOS ni bruit) et notées \tilde{I}_{MP} pour le canal I et \tilde{Q}_{MP} pour le canal Q. La suppression du multitrajet se fait simplement par soustraction des images I & Q par les images du multitrajet estimé :

$$I_c = \underbrace{I_{LOS} + I_{MP} + b_I}_{\text{image I}} - \tilde{I}_{MP}$$

$$Q_c = \underbrace{Q_{LOS} + Q_{MP} + b_Q}_{\text{image Q}} - \tilde{Q}_{MP}$$

avec I_c et Q_c les images corrigées des voies I et Q. Pour mesurer la qualité de suppression, plusieurs choix sont possibles. Ici nous évoquons les trois possibilités suivantes :

- Mesure de similarité entre images : étant capables de générer des sorties de corrélateur d'un signal LOS non perturbé, une première idée serait de comparer les images I_{LOS} et Q_{LOS} avec les images corrigées I_c et Q_c avec une métrique de distance entre images. Un exemple de mesure de similarités entre images est la distance de Hausdorff. Dans [145] Zhao et al. ont démontré la bonne robustesse à l'encontre du bruit d'une distance de Hausdorff modifiée pour mesurer la similarité d'images binaires de lettres de l'alphabet anglais,
- *Signal Quality Monitoring* (SQM) : une alternative est l'utilisation d'un SQM [146]. Opérateur servant à détecter les distorsions au niveau du récepteur à partir d'une valeur seuil fixée, il pourrait être couplé à la correction apportée par le CNN-HL. En appliquant le SQM sur les images I & Q puis sur les images I_c & Q_c , les différentes valeurs du SQM permettraient de discriminer le cas d'une correction améliorant le calcul de positionnement par CNN-HL ou au contraire d'une dégradation de la qualité de réception,
- Détection par *Deep Learning* : par l'utilisation d'algorithmes de détection de multitrajets aux images corrigées I_c et Q_c , on serait en capacité de classer la correction par le CNN-HL comme satisfaisante ou non. L'avantage de cette méthode concerne l'existence d'algorithmes de détection, notamment appliqués à des images de sorties de corrélateur [21, 18, 19].

7.2 Estimation de paramètres d'un seul multitrajet dans un environnement multi-interféré

Afin de prouver la possibilité d'estimation par régression des paramètres de multitrajets par CNN, les expériences de la thèse se sont déroulées sous une hypothèse simplificatrice d'un

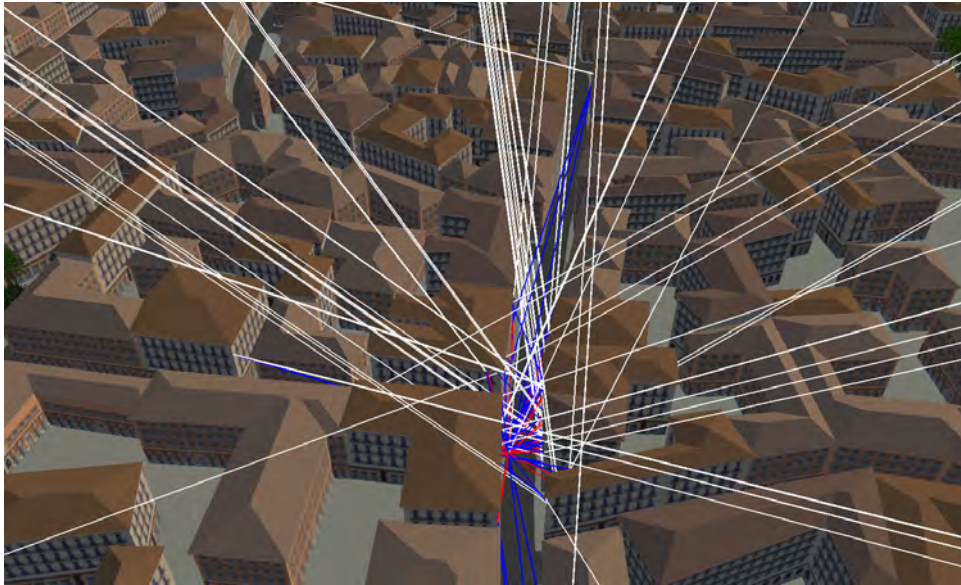


FIGURE 7.1 – Lors d'une simulation de réception dans une ruelle du scénario Toulouse centre, plusieurs multitrajets impactent le récepteur. Certaines de ces interférences sont issues du même signal LOS mais sont provoquées par une intéraction avec différentes façades de bâtiment. Par exemple trois multitrajets (rayons bleus) sont les conséquences d'une diffraction avec une arête d'un bâtiment bien plus loin au nord de la ruelle.

unique multitrajet présent à l'entrée du récepteur. Cependant dans la réalité plusieurs multitrajets de paramètres différents peuvent interférer en même temps au niveau de l'antenne réceptrice. Ce phénomène est illustré à la figure 7.1 dans le cadre d'une des simulations se déroulant au sein de l'environnement 3D du Capitole. Ce cas de figure plus réaliste fait l'objet de cette section : plusieurs expérimentations ont été conduites afin de tester les capacités du CNN à estimer les paramètres du multitrajet dans un environnement contenant d'autres multitrajets de plus faibles amplitudes.

Les images ont été générées à bruit constant pour deux résolutions : 80×80 et 20×20 pixels. Les multitrajets ont été aléatoirement générés selon les lois de distributions introduites en section 2.3. Nous avons remarqué dans les chapitres précédents que les performances de régression entre images 80×80 et 20×20 étaient similaires. Afin d'observer si c'est toujours le cas dans un environnement avec M multitrajets, les *datasets* avec plusieurs multitrajets ont été générés avec 80×80 sorties, afin d'avoir les meilleures performances atteignables par le réseau, et en 20×20 pour observer la robustesse de l'algorithme vis-à-vis d'une résolution 16 fois plus faible. Les figures 7.2, 7.3, 7.4 et 7.5 comparent les performances d'estimation du CNN avec et sans technique de *soft labelling* (Gaussienne tronquée pour les paramètres de retard, de fréquence Doppler et d'amplitude ; loi de Von Mises pour la phase). Le nombre de multitrajets considérés en abscisse prend en compte le multitrajet que l'on cherche à estimer. Le cas où le nombre de multitrajet est égal à 1 est la référence prise à partir des résultats des chapitres 4 et 5 : dans ce cas particulier le multitrajet à estimer est également le seul multitrajet présent en sortie de corrélateur. Un nombre de multitrajet égal à 2 signifie que le multitrajet à estimer est en présence d'un autre multitrajet de paramètres différents dont une amplitude plus faible. Pour les expériences exposées ci-dessous le nombre de multitrajets monte jusqu'à 5.

Dans la figure 7.2, l'erreur d'estimation augmente de manière constante en fonction du nombre de multitrajets présents. Le CNN avec *soft labelling* donne de meilleurs résultats dans tous les

7.2. Estimation de paramètres d'un seul multitrajet dans un environnem

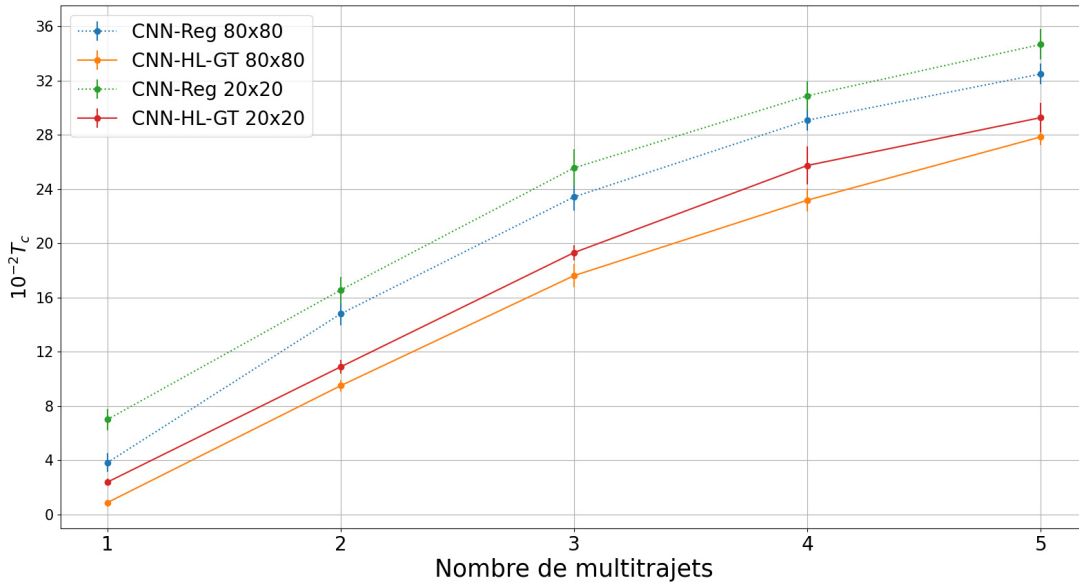


FIGURE 7.2 – Erreur moyenne d'estimation du retard $\Delta\tau_{MP}$ du multitrajet d'amplitude la plus importante en fonction du nombre de multitrajets présents, de la technique de régression choisie et de la résolution des images.

cas considérés par rapport au CNN-Reg. Avec l'hypothèse d'un retard $\Delta\tau_{MP} \in [0, 1.5 T_c]$, les courbes permettent d'observer un taux d'erreur dépassant les 25% T_c concernant l'estimation du retard de la part des deux modèles sur les deux tailles d'images pour une présence de 5 multitrajets.

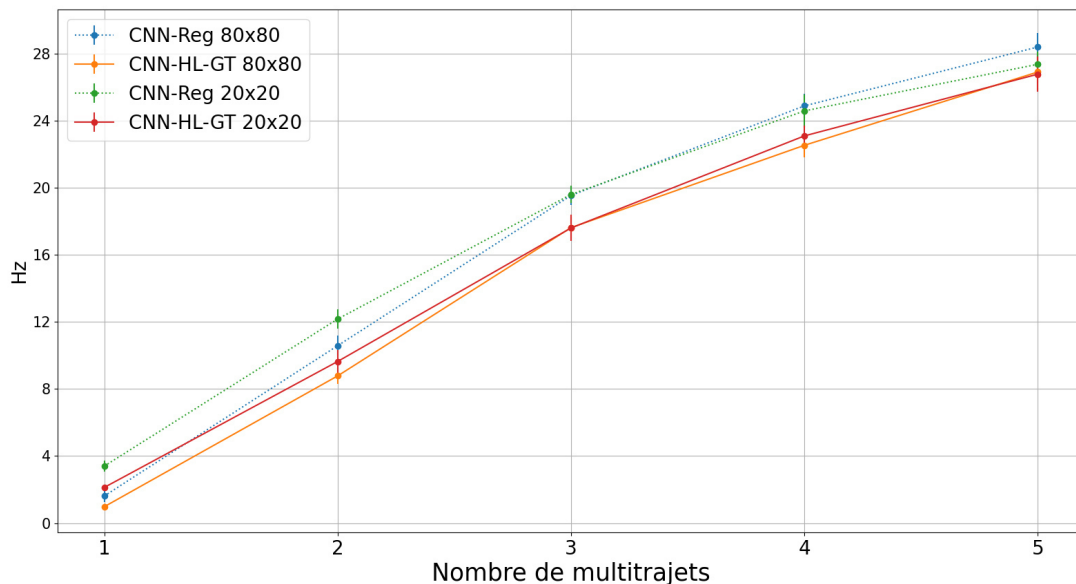


FIGURE 7.3 – Erreur moyenne d'estimation de la fréquence Doppler Δf_D du multitrajet d'amplitude la plus importante en fonction du nombre de multitrajets présents, de la technique de régression choisie et de la résolution des images.

Pour le paramètre de fréquence Doppler, une augmentation de l'erreur d'estimation est observée en fonction du nombre de multitrajets avec un comportement quasi-linéaire. L'augmentation d'erreur est conséquente entre les cas d'estimation passant de la présence d'un multitrajet à 2 multitrajets ainsi qu'en passant de 2 multitrajets à 3. Les performances d'estimation entre des images de sorties de corrélateur 80×80 et 20×20 sont semblables : les sorties de corrélateur supplémentaires des images 80×80 par rapport aux images 20×20 ne semblent pas impacter les capacités de régression du CNN avec ou sans *soft labelling*. Ceci est d'autant plus remarquable pour un nombre de multitrajets supérieur à 3 dans le cas de la régression par *Distributional Loss* où les différentes courbes de performances sur les images 80×80 et 20×20 sont quasiment confondues.

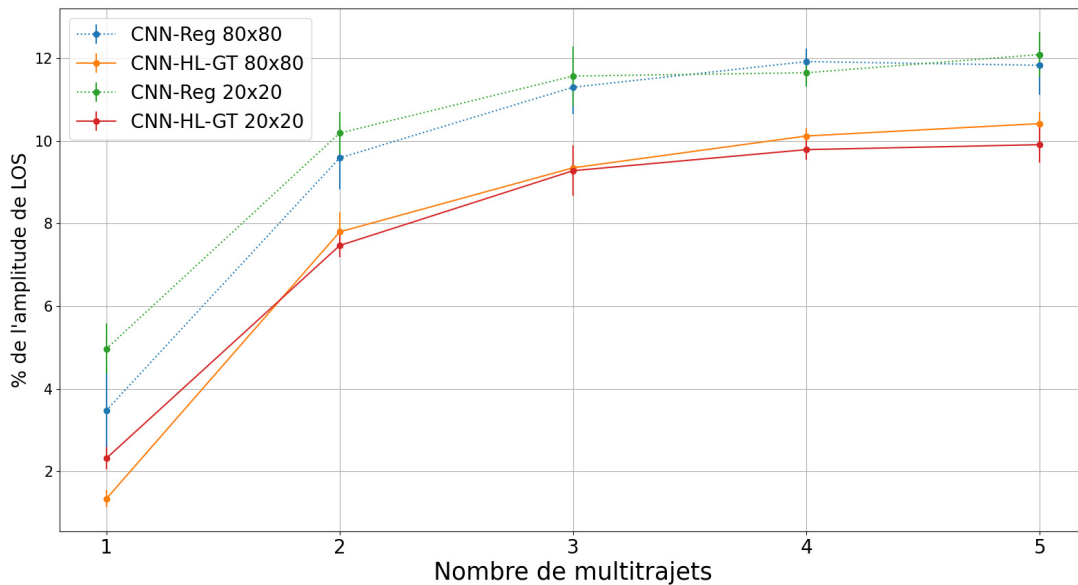


FIGURE 7.4 – Erreur moyenne d'estimation de l'amplitude $\Delta\alpha_{MP}$ du multitrajet le plus puissant en fonction du nombre de multitrajets présents, de la technique de régression choisie et de la résolution des images.

Dans le cas du paramètre d'amplitude du multitrajet, les courbes d'évolution des erreurs d'estimation montrent plutôt un comportement logarithmique en fonction du nombre de multitrajets. L'estimation de $\Delta\alpha_{MP}$ subit une grosse augmentation d'erreur une fois que le multitrajet à estimer n'est plus « seul ». Le gain de performances apporté par le CNN-HL-GT est ici bien visible avec une erreur d'estimation dépassant 10% de l'amplitude du signal LOS lorsque le nombre de multitrajets est supérieur à quatre. Pour le CNN-Reg, l'erreur d'estimation dépasse les 10% de l'amplitude du signal LOS en présence de seulement 2 multitrajets : en d'autres termes l'erreur minimale effectuée par le CNN-Reg lors de l'estimation d'un multitrajet en présence d'au moins un autre multitrajet est d'au moins 10% de l'amplitude du signal LOS.

Pour le paramètre de phase du multitrajet, le CNN-Reg n'est plus capable d'effectuer l'estimation par régression de la phase lorsque le nombre de multitrajet $M \geq 2$: l'erreur d'estimation atteint une moyenne de 90° . Avec le CNN-HL-VM, la régression de la phase reste possible : l'erreur est bien plus faible qu'avec le CNN-Reg (30° d'erreur en moyenne pour le CNN-HL-VM lorsque $M = 2$). Néanmoins l'utilisation du *Distributional Loss* avec une distribution selon une loi de Von Mises n'empêche pas l'erreur d'estimation du CNN d'augmenter de manière consi-

7.2. Estimation de paramètres d'un seul multitrajet dans un environnem

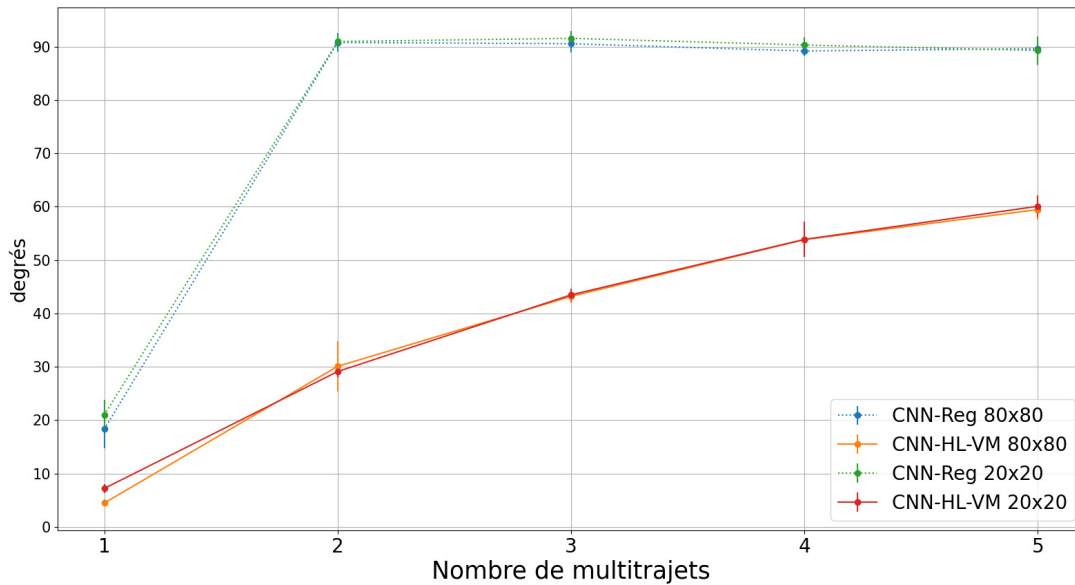


FIGURE 7.5 – Erreur moyenne d'estimation de la phase $\Delta\phi_{MP}$ du multitrajet d'amplitude la plus importante en fonction du nombre de multitrajets présents, de la technique de régression choisie et de la résolution des images.

dérable en fonction du nombre de multitrajets M par image. Pour un nombre de multitrajets $M > 3$, l'erreur d'estimation dépasse les 50° .

On observe de ces expériences effectuées sur les ensembles de données contenant plusieurs multitrajets par image une complexification de la tâche d'estimation par régression des paramètres du multitrajet de plus grande amplitude par rapport au cas d'un unique multitrajet. Les plus faibles multitrajets semblent parasiter la tâche d'estimation du multitrajet le plus fort. Le CNN-HL-GT (et sa version CNN-HL-VM pour la phase) permettent d'améliorer les performances de régression mais n'empêchent pas l'augmentation d'erreur constante avec le nombre de multitrajets. Pour la phase notamment, l'erreur dépasse les 25° dès la présence de 2 multitrajets. Les résultats permettent également d'observer des performances de régression similaires dans chaque expérience sur des images 80×80 et 20×20 , semblant indiquer que l'information nécessaire à l'estimation des paramètres est suffisante sur une résolution de 20×20 pixels.

De ces résultats préliminaires plusieurs pistes d'amélioration sont à étudier parmi lesquelles les paramètres du réseau et notamment sa structure, pouvant nécessiter des modifications avec l'augmentation du nombre de multitrajets. Les résultats font état d'une tâche de régression plus complexe à mesure que les multitrajets sont nombreux. Les expériences ont été conduites à titre de preuve de concept, sans recherche d'une optimisation du CNN pour une estimation de paramètres en milieu multi-interféré. De même malgré l'amélioration des performances apportée par du *Distributional Loss*, la présence de plusieurs multitrajets demande potentiellement un meilleur réglage des constructions des distributions a priori, notamment de la valeur de l'écart-type de la distribution et du nombre de bins divisant l'intervalle des *labels* possibles. Les conclusions que l'on peut pour l'instant tirer à partir des essais de régression sur un multitrajet en environnement multi-interféré avec des multitrajets aléatoirement générés est la difficulté croissante de la tâche de régression face au nombre de multitrajets. Les performances actuelles des modèles testés sont notamment insatisfaisantes pour le paramètre de phase à partir de 3 multitrajets et sont sujets à amélioration.

7.3 Estimation des paramètres de plusieurs multitrajets en environnement multi-interféré

En présence de plusieurs multitrajets, le problème d'estimation de paramètres par régression devient un problème de multi régression¹. Soit un couple d'images I & Q représentant des sorties de corrélateur subissant $M \in \mathbb{N}^*$ multitrajets. L'objectif est la recherche pour chaque multitrajet $i \in \llbracket 1, M \rrbracket$ du quadruplet de paramètres $(\Delta\tau_{MP}^i, \Delta f_D^i, \Delta\alpha_{MP}^i, \Delta\phi_{MP}^i)$. Nous proposons deux pistes de recherche : une approche itérative avec l'estimation de multitrajet d'amplitudes la plus forte à la plus faible et une approche en simultané où les paramètres des M multitrajets sont estimés en une seule fois.

7.3.1 Régression de multi multitrajets : approche itérative

L'idée de l'approche itérative est d'estimer tour à tour les paramètres des multitrajets présents, de celui de plus forte amplitude à la plus faible. L'approche se résume par le pseudo-code suivant :

1. une détection de multitrajet est effectuée sur des images I & Q . En cas de non-détection, l'image est considérée comme non impactée par un multitrajet. En cas de détection confirmée, on passe alors à l'étape suivante,
2. l'estimation des quatre paramètres est effectuée par le biais de CNN-HL,
3. une fois l'estimation des paramètres effectuée, les images de sorties de corrélateur \tilde{I}_{MP} & \tilde{Q}_{MP} correspondant au multitrajet sont construites,
4. une soustraction des images du multitrajet est effectuée sur les images I & Q afin d'obtenir I_c & Q_c ,
5. retour à l'étape 1 en remplaçant I & Q par I_c & Q_c .

Cependant la méthode présente une vulnérabilité au niveau de l'étape 2. Si l'estimation du signal n'est pas assez précise, la suppression de l'étape 4 ne sera pas suffisante et la détection de multitrajet considérera qu'il y a encore un multitrajet, demandant ainsi plus d'itérations. Or plus il y aura de corrections, plus il y aura une dégradation du pic de corrélation correspondant au signal LOS sur les images corrigées dû à la précision d'estimation des paramètres du multitrajet. Un des cas les plus défavorables serait une détection fantôme, c'est-à-dire un faux positif de la part de l'algorithme de détection, avec un algorithme cherchant à supprimer un multitrajet n'existant pas et altérant inutilement la composante du signal LOS.

7.3.2 Régression de multi multitrajets : approche en simultané

Dans la deuxième approche, nous souhaitons éviter les problèmes liés à une estimation et une élimination itérative des multitrajets, afin notamment d'éviter une dégradation trop importante de la composante du LOS sur des images I & Q . Dans cette approche, les paramètres des M multitrajets sont tous estimés en même temps : la tâche d'estimation par régression d'un paramètre devient une tâche de régression sur un M -uplet : par exemple avec l'estimation des M retards $(\Delta\tau_{MP}^1, \dots, \Delta\tau_{MP}^M)$ où $\Delta\tau_{MP}^i$ correspond au retard du multitrajet $i \in \llbracket 1, M \rrbracket$.

La méthode présente un certain nombre d'avantages et d'inconvénients. Parmi les avantages, elle va permettre de corriger M multitrajets à la fois. Là où la méthode itérative aurait au

1. à ne pas confondre avec la régression multiple qui est une généralisation de la régression linéaire pour un nombre fixé de variables explicatives

7.3. Estimation des paramètres de plusieurs multitrajets en environnement

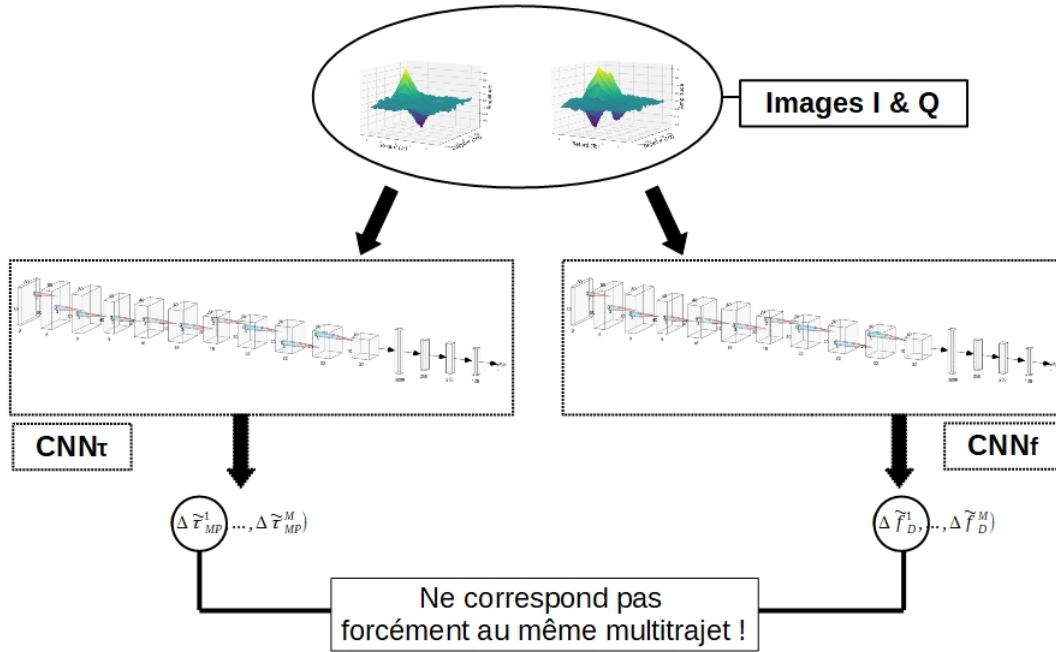


FIGURE 7.6 – Estimation de paramètres de multitrajets en parallèle. Sans ordonancement des multitrajets, on ne peut pas assurer la correspondance des paramètres estimés. Le problème est ici illustré avec les vecteurs d’estimation de retard et de fréquence Doppler. Les estimations $\Delta \tilde{\tau}_{MP}^i$ et $\Delta \tilde{f}_D^j$ ne correspondent pas forcément au même multitrajet.

minimum effectué M corrections sur les images, introduisant une dégradation du pic de corrélation du LOS à M reprises, ici l’opération de correction ne se fait qu’une fois. Cependant la détermination du M -uplet apporte une autre complexité. Le réseau de neurones doit exécuter M tâches de régression en parallèle, nécessitant la construction de suffisamment de *features* capables d’extraire les informations nécessaires concernant les M multitrajets.

La deuxième difficulté provient de l’attribution des paramètres : en s’appuyant sur les méthodes de suppression de multitrajets présentées en section 7.1, la suppression de chaque multitrajet nécessite l’estimation précise de son quadruplet de paramètres $(\Delta \tau_{MP}, \Delta f_D, \Delta \alpha_{MP}, \Delta \phi_{MP})$. Or lors de l’opération de régression, il faut s’assurer que les paramètres correspondent entre-eux. Prenons l’exemple illustré figure 7.6 : nous analysons les sorties de deux CNN, CNN_τ dédié à l’estimation du retard et CNN_f dédié à l’estimation de la fréquence Doppler (un seul CNN pourrait être entraîné à l’estimation des deux paramètres de retard et de fréquence Doppler à la fois; dans un soucis de clarté, un CNN est spécifique à l’estimation d’un type de paramètre). Suite à une paire d’images I & Q en entrée, CNN_τ obtient une sortie $(\Delta \tilde{\tau}_{MP}^1, \dots, \Delta \tilde{\tau}_{MP}^M)$ et CNN_f la sortie $(\Delta \tilde{f}_D^1, \dots, \Delta \tilde{f}_D^M)$. Rien n’assure que $\Delta \tilde{\tau}_{MP}^1$ et $\Delta \tilde{f}_D^1$ correspondent aux estimations d’un retard et d’une fréquence Doppler d’un même multitrajet. De manière générale, une sortie $\Delta \tilde{\tau}^i$ et une sortie $\Delta \tilde{f}_D^j$ ne correspondent pas forcément au même multitrajet $i \in \llbracket 1, M \rrbracket$. Pour contrer ce problème, une possibilité serait un apprentissage incitant le CNN à fournir les estimations classées selon les amplitudes des multitrajets. Par exemple, supposons M multitrajets tels que $\Delta \alpha_{MP}^1 > \Delta \alpha_{MP}^2 > \dots > \Delta \alpha_{MP}^M$: avec l’estimation des paramètres des multitrajets dans l’ordre décroissant de leur amplitude, une reconstruction des composantes des M multitrajets présents dans l’image à partir de l’estimation du quadruplet les caractérisant sera envisageable.

Une autre difficulté s’ajoute pour l’application de la méthode du *Distributional Loss* puisque

cela reviendrait à construire M distributions en sortie. Une juxtaposition des M distributions à priori n'est pas possible sur l'architecture actuelle en raison de la couche de *softmax*.

7.4 Améliorations

Dans le cadre de la thèse, nous avons effectué des estimations de paramètres par régression via un CNN améliorée par le *Distributional Loss*, méthode de *soft labelling* s'appuyant sur la divergence de KL pour corriger les distributions de prédiction du réseau. Plusieurs points d'améliorations pourraient être apportés :

- un changement de structure neuronale. Le réseau utilisé pour les différentes expériences de la thèse est un VGG. Son utilisation fut encouragée par plusieurs facteurs comme les bonnes performances obtenues lors du défi ILSVRC édition 2014 [20] et sa simplicité d'implémentation. Face à l'évolution très rapide des réseaux de neurones, il convient de dire que le VGG n'est pas l'une des structures neuronales les plus récentes. Lors de tests en parallèle des expériences, les performances d'estimation de paramètres de multitrajets par des réseaux *Inception* [109] ont été comparées avec celles du CNN-HL. Malgré une complexité neuronale beaucoup plus poussée, le réseau *Inception* arrivait à atteindre les performances proposées par le VGG sans les dépasser. Cependant, les expériences en section 7.2 montrent les limites du modèle quant à l'estimation de multitrajets en environnement multi-interféré. Les travaux de recherche de la thèse se concentraient sur la preuve de concept d'une régression des paramètres de multitrajet sur des images de sorties de corrélateur. Maintenant que la démonstration de sa possibilité est actée, une étude plus poussée du paramétrage du réseau ou une évolution vers une architecture alternative est souhaitable,
- la technique de *soft labelling* introduit au chapitre 5 a permis d'améliorer les performances de régression du CNN, rendant ses estimations sur images en 20×20 pixels plus précises qu'avec de la régression sur des *labels* continus sur des images 80×80 . La nouvelle distribution introduite, inspirée de la loi de Von Mises, a montré un potentiel d'amélioration bien plus élevé que la Gaussienne tronquée pour le paramètre de phase au comportement périodique. Pour ces deux distributions, les performances peuvent encore être améliorées en affinant des paramètres comme le choix du nombre de bins ou de l'écart-type,
- les expériences sur l'estimation de multitrajet en présence d'autres multitrajets plus faibles en section 7.2 ont montré des performances d'estimation de paramètres équivalentes pour des images de 20×20 pixels ou des images de 80×80 pixels. Les résultats ont permis de constater une différence de performances d'estimation notable lors du travail sur images en 10×10 pixels. Cela signifie qu'entre 20×20 et 10×10 la perte d'informations sur le signal en sortie du corrélateur est suffisamment importante au point de provoquer un effondrement des capacités d'estimation du réseau, malgré des performances améliorées par l'utilisation du *Distributional Loss*. Cependant le point d'amélioration n'est peut-être pas dans le nombre de sorties de corrélateur utilisées mais dans le calcul des sorties elles-mêmes. Une piste envisagée est de construire des images avec des sorties de corrélateur différentes. Plutôt que de faire une construction de sorties uniformément séparées (voir section 2.5), l'idée est de construire les sorties avec un échantillonnage différent des intervalles $[-\frac{3}{2} T_c, \frac{3}{2} T_c]$ et $[-250 \text{ Hz}, 250 \text{ Hz}]$: avec une génération des sorties selon une interpolation de Tchebychev par exemple.

7.5 Perspectives à plus long terme

À la suite de la preuve de concept de la régression de paramètres de multitrajet par CNN, les prochaines recherches sur l'estimation de multitrajet devraient se focaliser sur leur caractère temporel. En effet, nous avons uniquement considéré des images figées en sortie de corrélateur. Or les multitrajets évoluent dans le temps et possèdent une signature temporelle, potentiellement spécifique et propre à chaque multitrajet. De plus les multitrajets provoquent une variation des points d'équilibre des DLL et PLL et peuvent les attirer vers un nouveau point d'équilibre. La prise en compte de la dimension temporelle des multitrajets comme leur signature et leur influence sur l'équilibre des boucles de poursuite apporterait davantage d'informations que celles contenues dans des images I & Q réalisées à un instant t .

Les futures recherches devraient aussi porter un intérêt sur l'optimisation des poids du CNN, afin de réduire le plus possible sa dimension tout en conservant de bonnes performances pour pouvoir embarquer l'algorithme dans des récepteurs standards ou des VPU. D'autre part, une autre piste à explorer serait au contraire de développer des algorithmes plus complexes que les CNN étudiés afin de travailler sur des images de plus petite résolution (qui ont montrées une difficulté d'analyse croissante par les CNN testés), et ainsi s'approcher d'un nombre de sorties de corrélateur plus réalisables (ou calculées différemment comme indiqué dans la section précédente). De même dans un contexte plus large, les régressions sur images devraient davantage faire recours à la technique de *soft labelling* par *Distributional Loss*. Le gain de performances est conséquent et permet d'obtenir de meilleures performances sur des images de résolution réduite.

Conclusion

Face au problème de multitrajet perturbant le fonctionnement du corrélateur d'un récepteur GNSS, la thèse a eu pour objectif d'apporter une solution en proposant une méthode d'estimation des paramètres de retard, fréquence Doppler, amplitude et phase du signal parasite à l'aide d'algorithmes de *Deep Learning* appliqués sur des images de sorties de corrélateur. C'est, à notre connaissance, la première fois qu'un CNN a été appliqué sur des images de sorties de corrélateur en vue d'une tâche de régression. Une quantité de données assez conséquente étant nécessaire pour l'apprentissage des algorithmes de *Deep Learning*, nous avons construit les différents *datasets* des expériences à partir d'un générateur entièrement paramétrable et capable de calculer un nombre flexible de sorties de corrélateur. Les signaux directs et multitrajets synthétiques ont été aléatoirement générés dans un premier temps. Afin d'entraîner le modèle avec des multitrajets aux paramètres plus réalistes, nous avons réalisé une interface entre le générateur et un simulateur de propagation de signaux GNSS en environnement 3D, capable de reproduire le phénomène de multitrajet dans des environnements physiques simulés comme dans des canyons urbains d'un centre-ville ou dans un quartier de gratte-ciels. Les paramètres des multitrajets collectés ont permis d'alimenter le générateur d'images et de construire les images de sorties de corrélateur correspondant aux multitrajets simulés.

Inspirées des travaux dans [21], les images formées à partir de sorties de corrélateur permettent de donner un maximum d'informations sur le signal LOS et le multitrajet. L'information du multitrajet étant répartie sur les deux images, l'estimation de paramètres devient une tâche de régression multicanaux. Les réseaux de neurones à convolution ayant prouvé leur efficacité en termes d'analyse d'images pour de la classification et dans une moindre mesure pour de la régression, nous nous sommes naturellement dirigés vers leur utilisation pour réaliser la tâche d'estimation multicanaux. Afin d'améliorer les performances de régression du réseau, nous avons, à notre connaissance, réalisé le premier couplage entre un CNN et la technique du *Distributional Loss*. Méthode introduite dans [23], elle consiste à remplacer les *labels* sous forme de variables numériques continues par des distributions de probabilité d'espérances égales aux *labels*. Avec l'utilisation de distributions sous la forme d'une Gaussienne tronquée pour les paramètres de retard, fréquence Doppler et d'amplitude et avec l'introduction d'une distribution en loi de Von Mises plus adaptée à des paramètres définis sur un intervalle périodique comme la phase du multitrajet, nous avons démontré la possibilité d'estimation de paramètres de multitrajets par CNN. Le couplage entre le CNN et la technique de *Distributional Loss* est un succès. L'amélioration de la précision d'estimation par rapport au CNN apprenant sans *soft labelling* est notable, quels que soient les niveaux de bruits, la puissance des multitrajets, la résolution des images I & Q ou les caractéristiques physiques des environnements de simulation ayant permis la génération des multitrajets. Dans certains cas, la robustesse apportée par le *Distributional Loss* permet d'obtenir de meilleures performances sur des images avec une résolution seize fois moins élevée par rapport au CNN apprenant sans *soft labelling*. Les expériences à partir de la chaîne de génération des données réalisant l'interface entre un simulateur de pro-

pagation d'ondes et le générateur de sorties de corrélateur ont démontré deux choses : le CNN est capable d'effectuer une tâche de régression multicanaux sur des images de multitrajets de paramètres physiquement réalistes et la technique de *Distributional Loss* conserve sa capacité d'amélioration de performances sur ces données plus réalistes.

Des expériences préliminaires sur des *datasets* d'environnements contenant plusieurs multitrajets ne font que confirmer les meilleures performances du CNN-HL, la différence étant flagrante pour le paramètre de phase. La prochaine étape découlant de ces travaux de thèse se dirige intuitivement vers la suppression de la composante de multitrajet qui était la motivation principale de l'estimation des paramètres de multitrajets. En plus de la suppression du signal interférant, les futures recherches devraient prendre en compte la dimension temporelle des multitrajets et la lier aux estimations de paramètres. Les résultats de la thèse sont encourageants à propos de l'utilisation d'images de sorties de corrélateur. Des travaux de recherche ayant montré la possibilité de détection de multitrajets par CNN comme dans [18] et les expériences de cette thèse ayant prouvé la possibilité d'estimation des paramètres des multitrajets avec une précision satisfaisante, l'ensemble de ces résultats devrait inciter les chercheurs et industriels à envisager davantage de sorties de corrélateur (sous forme d'images par exemple) au niveau des récepteurs de prochaines générations. L'espace mémoire de taille raisonnable occupé par le couplage CNN et *Distributional Loss* et des expériences préliminaires menées sur des VPU [19] laissent également entrevoir un futur avec une nouvelle génération de récepteurs avec des technologies *Machine Learning* embarquées.

Glossaire

A

AdaGrad *Adaptive Gradient Descent.*

Adam *Adaptive Moment Estimation.*

ADC *Analog to Digital Converter.*

AWGN *Additive White Gaussian Noise.*

B

BPSK *Bi-Phase Shift Keying.*

C

C/A *Coarse/Acquisition.*

CDG *Charles-De-Gaulle.*

CmC *Code-minus-Carrier.*

CNN *Convolutional Neural Network.*

CNN-HL *Convolutional Neural Network - Histogram Loss.*

D

DL *Deep Learning.*

DLL *Delay-Locked Loop.*

E

ELS *Early-Late slope.*

ENAC *École Nationale de l'Aviation Civile.*

E-P-L *Early, Prompt and Late.*

ECEF *Earth-Centered, Earth-Fixed.*

ESA *European Space Agency.*

EUSPA *Agence de l'Union européenne pour le programme spatial.*

F

FFT *Fast Fourier Transform.*

FIMLA *Fast Iterative Maximum-Likelihood Algorithm.*

G

GNSS *Global Navigation Satellite System.*

GPS *Global Positioning System.*

H

HL *Histogram Loss.*

I

IA *Intelligence Artificielle.*

ILSVRC *ImageNet Large Scale Visual Recognition Challenge.*

IRM *Imagerie par Résonance Magnétique.*

ITU *International Telecommunication Union.*

J

JSR *Jamming to Signal Ratio.*

K

KL *Kullback-Leibler.*

L

LEO *Low Earth Orbit.*

LOS *Line-Of-Sight.*

M

MAE *Mean Absolute Error.*

MAPE *Mean Absolute Percentage Error.*

MEDLL *Multipath Estimating Delay-Locked Loop.*

MEO *Medium Earth Orbit.*

MIDLL *Multipath Insensitive Delay-Locked Loop.*

ML *Machine Learning.*

MLP *Multi-Layer Perceptron.*

MNIST *Mixed National Institute of Standards and Technology.*

Moderated MP *Moderated Multipath.*

MSE *Mean Square Error.*

N

NCA *Neighbourhood Component Analysis.*

NLOS *Non Line-Of-Sight.*

NMEA *National Marine Electronics Association.*

NYC *New York City.*

P

PLL *Phase-Locked Loop.*

PNT *Positioning, Navigation and Timing.*

PRN *Pseudo-Random Noise.*

R

Random MP *Random Multipath.*

RF *Radio-Frequency.*

RGB *Red-Green-Blue.*

RINEX *Receiver Independent Exchange Format.*

RMS *Root-Mean-Square.*

RMS-Prop *Root Mean Square Propagation.*

RMSE *Root-Mean-Square Error.*

RNN *Recurrent Neural Network.*

S

SE-Nav *SE-Nav.*

SEL *Stacking Ensemble Learning.*

SigNav *SIGnal processing and NAVigation.*

SINR *Signal to Interference plus Noise Ratio.*

SNR *Signal-to-Noise Ratio.*

SPS *Standard Positioning Service.*

SQM *Signal Quality Monitoring.*

Strong MP *Strong Multipath.*

SVM *Support Vector Machine.*

SVR *Support Vector Regression.*

T

TK-MEDLL *Teager and Kaiser Multipath Estimating Delay-Locked Loop.*

TOA *Time Of Arrival.*

U

URA *User Range Accuracy.*

V

VGG *Visual Geometry Group.*

VL *Very Late.*

VPU *Vision Processing Unit.*

W

Weak MP *Weak Multipath.*

Bibliographie

- [1] European GNSS Agency, “GSA GNSS market report,” tech. rep., European Global Navigation Satellite Systems Agency, Luxembourg, 2020. Dernier accès : 2023-12-06.
- [2] W. Vigneau, O. Nouvel, M. Manzano-Jurado, C. C. Sanz, H. Abdulkader, D. Roviras, J. Juan, and P. Holsters, “Neural networks algorithms prototyping to mitigate GNSS multipath for LEO positioning applications,” in *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, pp. 1752–1762, 2006.
- [3] Y. Quan, L. Lau, G. W. Roberts, X. Meng, and C. Zhang, “Convolutional neural network based multipath detection method for static and kinematic GPS high precision positioning,” *Remote Sensing*, vol. 10, no. 12, p. 2052, 2018.
- [4] A. Van Dierendonck, P. Fenton, and T. Ford, “Theory and performance of narrow correlator spacing in a GPS receiver,” *Navigation*, vol. 39, no. 3, pp. 265–283, 1992.
- [5] N. Jardak, A. Vervisch-Picois, and N. Samama, “Multipath insensitive delay lock loop in GNSS receivers,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 2590–2609, 2011.
- [6] H. Qin, X. Xue, and Q. Yang, “GNSS multipath estimation and mitigation based on particle filter,” *IET Radar, Sonar & Navigation*, vol. 13, no. 9, pp. 1588–1596, 2019.
- [7] M. Panahi, N. Sadhasivam, H. R. Pourghasemi, F. Rezaie, and S. Lee, “Spatial prediction of groundwater potential mapping based on convolutional neural network (CNN) and support vector regression (SVR),” *Journal of Hydrology*, vol. 588, p. 125033, 2020.
- [8] L.-T. Hsu, “GNSS multipath detection using a machine learning approach,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2017.
- [9] C. Savas and F. Dervis, “Multipath detection based on K-means clustering,” in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, pp. 3801–3811, 2019.
- [10] T. Suzuki and Y. Amano, “NLOS multipath classification of GNSS signal correlation output using machine learning,” *Sensors*, vol. 21, no. 7, p. 2503, 2021.
- [11] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, *et al.*, “Learning algorithms for classification : A comparison on handwritten digit recognition,” *Neural networks : the statistical mechanics perspective*, vol. 261, no. 276, p. 2, 1995.
- [12] D. Yi, Z. Lei, and S. Z. Li, “Age estimation by multi-scale convolutional network,” in *Asian conference on computer vision*, pp. 144–158, Springer, 2014.
- [13] S. Li and A. B. Chan, “3D human pose estimation from monocular images with deep convolutional neural network,” in *Computer Vision—ACCV 2014 : 12th Asian Conference*

- on *Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12*, pp. 332–347, Springer, 2015.
- [14] S. Mahendran, H. Ali, and R. Vidal, “3D pose regression using convolutional neural networks,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 2174–2182, 2017.
- [15] M. Ueda, K. Ito, K. Wu, K. Sato, Y. Taki, H. Fukuda, and T. Aoki, “An age estimation method using 3D-CNN from brain MRI images,” in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, pp. 380–383, IEEE, 2019.
- [16] S. Miao, Z. J. Wang, and R. Liao, “A CNN regression approach for real-time 2D/3D registration,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1352–1363, 2016.
- [17] A. Guillard, P. Thevenon, and C. Milner, “Using convolutional neural networks to detect GNSS multipath,” *Frontiers in Robotics and AI*, vol. 10, p. 1106439, 2023.
- [18] E. Munin, A. Blais, and N. Couellan, “Convolutional neural network for multipath detection in GNSS receivers,” in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pp. 1–10, IEEE, 2020.
- [19] E. Munin, A. Blais, and N. Couellan, “GNSS multipath detection using embedded deep CNN on intel® neural compute stick,” in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. 2018–2029, 2020.
- [20] K. Simonyan and A. Zisserman, “Very deep convnets for large-scale image recognition,” *Computing Research Repository*, 2014.
- [21] A. Blais, N. Couellan, and E. Munin, “A novel image representation of GNSS correlation for deep learning multipath detection,” *Array*, vol. 14, p. 100167, 2022.
- [22] R. Yozevitch, B. B. Moshe, and A. Weissman, “A robust GNSS LOS/NLOS signal classifier,” *Navigation : Journal of The Institute of Navigation*, vol. 63, no. 4, pp. 429–442, 2016.
- [23] E. Imani and M. White, “Improving regression performance with distributional losses,” in *International conference on machine learning*, pp. 2157–2166, PMLR, 2018.
- [24] G. Brain, “Tensorflow version 2.4.1.” <https://www.tensorflow.org/>, 9 Janvier 2021. Dernière version : 2023-11-14.
- [25] T. Gonzalez, “Code python CNN-regression.” Disponible sur <ssh://git@git.recherche.enac.fr/cnn-regression/git-these.git>, 2023. Dernière mise à jour : 2023-12-21.
- [26] J. B.-Y. Tsui, *Fundamentals of global positioning system receivers : a software approach*. John Wiley & Sons, 2005.
- [27] A. T. Balaei, A. G. Dempster, and J. Barnes, “A novel approach in detection and characterization of CW interference of GPS signal using receiver estimation of C/N0,” in *Proceedings of IEEE/ION PLANS 2006*, pp. 1120–1126, 2006.
- [28] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system : theory and practice*. Springer Science & Business Media, 2012.
- [29] C. Macabiau, O. Julien, P. Thevenon, A.-C. Escher, and A. Martineau, “GNSS pour l’aviation civile,” décembre 2016. cours Telecom Lab - ENAC.
- [30] United States. Department of Defense, “Global positioning system standard positioning service signal specification,” *United States Coast Guard Navigation Center*, 2020.
- [31] S. K. Shanmugam, *New Enhanced Sensitivity Detection Techniques for GPS L1 C/A and Modernized Signal Acquisition*. PhD thesis, University of Calgary, 2008.

- [32] M.-A. Fortin, J.-C. Guay, *et al.*, “Single frequency WAAS augmentation observations (L1 vs. L5) on a ground based GPS L1 C/A solution,” *Positioning*, vol. 5, no. 03, p. 70, 2014.
- [33] C. Savas, G. Falco, and F. Dovis, “A comparative performance analysis of GPS L1 C/A, L5 acquisition and tracking stages under polar and equatorial scintillations,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 227–244, 2020.
- [34] D. Manandhar, Y. Suh, and R. Shibasaki, “GPS signal acquisition and tracking—an approach towards development of software-based GPS receiver,” *Technical Report of IEICE*, 2004.
- [35] G. Blewitt, “Basics of the GPS technique : observation equations,” *Geodetic applications of GPS*, vol. 1, p. 46, 1997.
- [36] J. J. Spilker Jr, P. Axelrad, B. W. Parkinson, and P. Enge, *Global positioning system : theory and applications, volume I*. American Institute of Aeronautics and Astronautics, 1996.
- [37] H. Al Bitar, *Techniques avancées de traitement du signal GPS pour les services LBS*. PhD thesis, Toulouse, INPT, 2007.
- [38] O. Julien and C. Macabiau, “Spread spectrum communications,” 2018. Cours TELECOM Lab - ENAC.
- [39] R. Gold, “Optimal binary sequences for spread spectrum multiplexing (corresp.),” *IEEE Transactions on information theory*, vol. 13, no. 4, pp. 619–621, 1967.
- [40] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver*. Springer Science & Business Media, 2007.
- [41] F. Duquenne, S. Botton, F. Peyret, D. Bétaille, and P. Willis, *GPS : localisation et navigation par satellites*. Hermes science publications, 2005.
- [42] M. Tamazin, M. Karaim, and A. Noureldin, “GNSSs, signals, and receivers,” in *Multifunctional Operation and Application of GPS* (R. B. Rustamov and A. M. Hashimov, eds.), ch. 6, Rijeka : IntechOpen, 2018.
- [43] M. S. Braasch and A. Van Dierendonck, “GPS receiver architectures and measurements,” *Proceedings of the IEEE*, vol. 87, no. 1, pp. 48–64, 1999.
- [44] E. D. Kaplan and C. Hegarty, *Understanding GPS/GNSS : principles and applications*. Artech house, 2017.
- [45] C. O’driscoll, M. G. Petovello, and G. Lachapelle, “Choosing the coherent integration time for Kalman filter-based carrier-phase tracking of GNSS signals,” *GPS solutions*, vol. 15, pp. 345–356, 2011.
- [46] C. Fernández-Prades, “GNSS-SDR : Acquisition.” <https://gnss-sdr.org/docs/sp-blocks/acquisition/>, 2024.
- [47] M. Smyrnaio, S. Schn, M. Liso, and S. Jin, “Multipath propagation, characterization and modeling in GNSS,” *Geodetic sciences-observations, modeling and applications*, pp. 99–125, 2013.
- [48] L. Cheng, K. Wang, M. F. Ren, and X. Y. Xu, “Comprehensive analysis of multipath estimation algorithms in the framework of information theoretic learning,” *IEEE Access*, vol. 6, pp. 5521–5530, 2018.
- [49] R. D. Van Nee, “Multipath effects on GPS code phase measurements,” *Navigation*, vol. 39, no. 2, pp. 177–190, 1992.
- [50] O. M. Mubarak and A. Dempster, “Carrier phase analysis to mitigate multipath effect,” in *Proc. International Global Navigation Satellite Systems Society (IGNSS) Symposium, Sydney*, Citeseer, 2007.

- [51] J. Jang, M. Paonni, and B. Eissfeller, “CW interference effects on tracking performance of GNSS receivers,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 243–258, 2012.
- [52] T. Kos, I. Markezic, and J. Pokrajcic, “Effects of multipath reception on GPS positioning performance,” in *Proceedings ELMAR-2010*, pp. 399–402, IEEE, 2010.
- [53] A. Kerkhoff, R. B. Harris, C. P. Petersen, and A. Pickard, “Modifications to GPS reference station antennas to reduce multipath,” in *Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010)*, pp. 866–878, 2010.
- [54] B. R. Townsend, P. C. Fenton, K. J. Van Dierendonck, and D. R. Van Nee, “Performance evaluation of the Multipath Estimating Delay Lock Loop,” *Navigation*, vol. 42, no. 3, pp. 502–514, 1995.
- [55] R. D. Van Nee, J. Siereveld, P. C. Fenton, and B. R. Townsend, “The Multipath Estimating Delay Lock Loop : approaching theoretical accuracy limits,” in *Proceedings of 1994 IEEE Position, Location and Navigation Symposium-PLANS'94*, pp. 246–251, IEEE, 1994.
- [56] B. Townsend and P. Fenton, “A practical approach to the reduction of pseudorange multipath errors in a L1 GPS receiver,” in *Proceedings of the 7th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, UT, USA*, pp. 20–23, 1994.
- [57] A. Broumandan, J. Nielsen, and G. Lachapelle, “TOA estimation enhancement based on blind calibration of synthetic arrays,” in *2008 IEEE 68th Vehicular Technology Conference*, pp. 1–5, IEEE, 2008.
- [58] Y. Sun, Z. Yao, and M. Lu, “Extended double-delta correlator technique for GNSS multipath mitigation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 1758–1773, 2022.
- [59] Y. Zhang and C. Bartone, “Multipath mitigation in the frequency domain,” in *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No. 04CH37556)*, pp. 486–495, IEEE, 2004.
- [60] M. Caamano, O. G. Crespillo, D. Gerbeth, and A. Grosch, “Detection of GNSS multipath with time-differenced code-minus-carrier for land-based applications,” in *2020 European Navigation Conference (ENC)*, pp. 1–12, IEEE, 2020.
- [61] V. P. García and N. Woodhouse, “Multipath analysis using Code-Minus-Carrier technique in GNSS antennas.” https://www.taoglas.com/wp-content/uploads/pdf/Multipath-Analysis-Using-Code-Minus-Carrier-Technique-in-GNSS-Antennas-WhitePaper_VP_v6.pdf, 2020.
- [62] M. Sahnoudi and M. G. Amin, “Fast iterative maximum-likelihood algorithm (FIMLA) for multipath mitigation in the next generation of GNSS receivers,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4362–4374, 2008.
- [63] N. Blanco-Delgado and F. D. Nunes, “Multipath estimation in multicorrelator GNSS receivers using the maximum likelihood principle,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3222–3233, 2012.
- [64] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, “Particle filter-based prognostics : Review, discussion and perspectives,” *Mechanical Systems and Signal Processing*, vol. 72, pp. 2–31, 2016.
- [65] J. F. Kaiser, “On a simple algorithm to calculate the ‘energy’ of a signal,” in *International conference on acoustics, speech, and signal processing*, pp. 381–384, IEEE, 1990.

- [66] J. YuanFa, T. LianJie, S. XiYan, Y. Suqing, K. H. Ghazali, and S. Khatuni, "An improved Multipath Estimating Delay-Lock-Loop method based on Teager-Kaiser operator," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-2, pp. 49–53, 2018.
- [67] Y.-S. Park and S. Lek, "Artificial neural networks : Multilayer perceptron for ecological modeling," in *Developments in environmental modelling*, vol. 28, pp. 123–140, Elsevier, 2016.
- [68] N. Sharma, R. Sharma, and N. Jindal, "Machine Learning and Deep Learning applications-a vision," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 24–28, 2021.
- [69] M. Gheisari, G. Wang, and M. Z. A. Bhuiyan, "A survey on deep learning in big data," in *2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*, vol. 2, pp. 173–180, IEEE, 2017.
- [70] R. Dahl, M. Norouzi, and J. Shlens, "Pixel recursive super resolution," in *Proceedings of the IEEE international conference on computer vision*, pp. 5439–5448, 2017.
- [71] A. Koul, S. Ganju, and M. Kasam, *Practical Deep Learning for Cloud, Mobile, and Edge : Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*. O'Reilly Media, 2019.
- [72] A. Pai, "CNN vs. RNN vs. ANN—analyzing 3 types of neural networks in deep learning," *Analytics Vidhya, Feb*, vol. 17, 2020.
- [73] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.
- [74] A. A. Barbhuiya, R. K. Karsh, and R. Jain, "CNN based feature extraction and classification for sign language," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 3051–3069, 2021.
- [75] V. R. K. Kolla, "Paws and reflect : A comparative study of Deep Learning techniques for cat vs dog image classification," *International Journal of Computer Engineering and Technology*, 2020.
- [76] A. Toshev and C. Szegedy, "Deeppose : Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1653–1660, 2014.
- [77] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection : A survey," *arXiv preprint arXiv :1901.03407*, 2019.
- [78] E. Poisson, C. Viard-Gaudin, and P. Lallican, "Réseaux de neurones à convolution : reconnaissance de l'écriture manuscrite non contrainte," *CIFED 2002*, 2001.
- [79] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [80] D. Gianazza, "Principles and methodology of machine learning," 2021. Cours mineure DATA ENAC.
- [81] Google, "Cours d'initiation au Machine Learning." <https://developers.google.com/machine-learning/crash-course?hl=fr>. Dernier accès 2021-04-15.
- [82] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications : A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.
- [83] R. Morales Ferre, A. de la Fuente, and E. S. Lohan, "Jammer classification in GNSS bands via machine learning algorithms," *Sensors*, vol. 19, no. 22, p. 4841, 2019.
- [84] H. Wang, J. Wang, J. Zhai, and X. Luo, "Detection of triple JPEG compressed color images," *IEEE Access*, vol. 7, pp. 113094–113102, 2019.

- [85] H. Wu, Y. Li, L. Zhou, and J. Meng, “Convolutional neural network and multi-feature fusion for automatic modulation classification,” *Electronics Letters*, vol. 55, no. 16, pp. 895–897, 2019.
- [86] A. Géron, “Hands-on Machine Learning with scikit-learn and tensorflow : Concepts, tools, and techniques to build intelligent systems,” in *Hands-On Machine Learning with Scikit-Learn and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems*, 2017.
- [87] L. Delmas, “Réseaux de neurones convolutifs : un cortex visuel virtuel pour la reconnaissance d’images.” *GNU Linux magazine*, october 2019.
- [88] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and Machine Learning*, vol. 4. Springer, 2006.
- [89] T. Shimobaba, T. Kakue, and T. Ito, “Convolutional neural network-based regression for depth prediction in digital holography,” in *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*, pp. 1323–1326, IEEE, 2018.
- [90] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, “A comprehensive analysis of deep regression,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 9, pp. 2065–2081, 2019.
- [91] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, “Regression convolutional neural network for improved simultaneous EMG control,” *Journal of neural engineering*, vol. 16, no. 3, p. 036015, 2019.
- [92] R. Yan and L. Shao, “Blind image blur estimation via deep learning,” *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1910–1921, 2016.
- [93] S. Chougule, N. Koznek, A. Ismail, G. Adam, V. Narayan, and M. Schulze, “Reliable multilane detection and classification by utilizing CNN as a regression network,” in *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.
- [94] D. Jagiwala and S. N. Shah, “Possibilities of AI algorithm execution in GNSS,” in *2022 URSI Regional Conference on Radio Science (USRI-RCRS)*, pp. 1–4, IEEE, 2022.
- [95] A. Elango, S. Ujan, and L. Ruotsalainen, “Disruptive GNSS signal detection and classification at different power levels using advanced deep-learning approach,” in *2022 International Conference on Localization and GNSS (ICL-GNSS)*, pp. 1–7, IEEE, 2022.
- [96] P. Ramezanpour, M. J. Rezaei, and M. R. Mosavi, “Deep-learning-based beamforming for rejecting interferences,” *IET Signal Processing*, vol. 14, no. 7, pp. 467–473, 2020.
- [97] O. G. Crespillo, J. C. Ruiz-Sicilia, A. Kliman, and J. Marais, “Robust design of a machine learning-based GNSS NLOS detector with multi-frequency features,” *Frontiers in Robotics and AI*, vol. 10, 2023.
- [98] Y. Sun and L. Fu, “Stacking ensemble learning for non-line-of-sight detection of global navigation satellite system,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022.
- [99] H. Xu, A. Angrisano, S. Gaglione, and L.-T. Hsu, “Machine learning based LOS/NLOS classifier and robust estimator for GNSS shadow matching,” *Satellite Navigation*, vol. 1, no. 1, pp. 1–12, 2020.
- [100] G. Zhang, B. Xu, and L.-T. Hsu, “GNSS shadow matching based on intelligent LOS/NLOS classifier,” *16th IAIN World Congr*, vol. 2018, 2018.
- [101] Q. Liu, Z. Huang, and J. Wang, “Indoor non-line-of-sight and multipath detection using deep learning approach,” *GPS Solutions*, vol. 23, pp. 1–14, 2019.

- [102] K. Bregar and M. Mohorčič, “Improving indoor localization using convolutional neural networks on computationally restricted devices,” *IEEE Access*, vol. 6, pp. 17429–17441, 2018.
- [103] B. Xu, Q. Jia, Y. Luo, and L.-T. Hsu, “Intelligent GPS L1 LOS/multipath/NLOS classifiers based on correlator-, RINEX- and NMEA-level measurements,” *Remote Sensing*, vol. 11, no. 16, p. 1851, 2019.
- [104] A. LeNail, “NN-SVG : Alexnet.” <https://alexlenail.me/NN-SVG/AlexNet.html>, 2022.
- [105] L. Datta, “A survey on activation functions and their relation with Xavier and He normal initialization,” *arXiv preprint arXiv :2004.06632*, 2020.
- [106] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent advances in adversarial training for adversarial robustness,” *arXiv preprint arXiv :2102.01356*, 2021.
- [107] F. Tramer and D. Boneh, “Adversarial training and robustness for multiple perturbations,” *Advances in neural information processing systems*, vol. 32, 2019.
- [108] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, “Robust physical-world attacks on Deep Learning visual classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.
- [109] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.
- [110] Y. Zhao, H. Zhang, and X. Hu, “Penalizing gradient norm for efficiently improving generalization in deep learning,” in *International Conference on Machine Learning*, pp. 26982–26992, PMLR, 2022.
- [111] E. Imani, *Distribution Losses for Regression*. PhD thesis, University of Alberta, 2019.
- [112] E. Lei, H. Hassani, and S. S. Bidokhti, “Out-of-distribution robustness in Deep Learning compression,” *arXiv preprint arXiv :2110.07007*, 2021.
- [113] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized out-of-distribution detection : A survey. arxiv,” *arXiv preprint arXiv :2110.11334*, 2021.
- [114] G. Gojić, V. Vincan, O. Kundačina, D. Mišković, and D. Dragan, “Non-adversarial robustness of Deep Learning methods for computer vision,” *arXiv preprint arXiv :2305.14986*, 2023.
- [115] A. Raj, Y. Bresler, and B. Li, “Improving robustness of deep-learning-based image reconstruction,” in *International Conference on Machine Learning*, pp. 7932–7942, PMLR, 2020.
- [116] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, “Understanding robustness of transformers for image classification,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10231–10241, 2021.
- [117] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri, “A closer look at accuracy vs. robustness,” *Advances in neural information processing systems*, vol. 33, pp. 8588–8601, 2020.
- [118] M. Wyatt, B. Radford, N. Callow, M. Bennamoun, and S. Hickey, “Using ensemble methods to improve the robustness of Deep Learning for image classification in marine environments,” *Methods in Ecology and Evolution*, vol. 13, no. 6, pp. 1317–1328, 2022.
- [119] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, “Measuring robustness to natural distribution shifts in image classification,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 18583–18599, 2020.

- [120] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Robust object recognition with cortex-like mechanisms,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 411–426, 2007.
- [121] M. Z. Darestani, A. S. Chaudhari, and R. Heckel, “Measuring robustness in Deep Learning based compressive sensing,” in *International Conference on Machine Learning*, pp. 2433–2444, PMLR, 2021.
- [122] V. Sharmanska, D. Hernández-Lobato, J. Miguel Hernandez-Lobato, and N. Quadrianto, “Ambiguity helps : Classification with disagreements in crowdsourced annotations,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2194–2202, 2016.
- [123] X. Geng, “Label distribution learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.
- [124] Q.-H. Phan, S.-L. Tan, I. McLoughlin, and D.-L. Vu, “A unified framework for GPS code and carrier-phase multipath mitigation using support vector regression,” *Advances in Artificial Neural Systems*, vol. 2013, 2013.
- [125] G. Zhang, C. Wang, B. Xu, and R. Grosse, “Three mechanisms of weight decay regularization,” *arXiv preprint arXiv :1810.12281*, 2018.
- [126] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” *arXiv preprint arXiv :1708.04552*, 2017.
- [127] C. F. Gonçalves dos Santos and J. P. Papa, “Avoiding overfitting : A survey on regularization methods for convolutional neural networks,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1–25, 2022.
- [128] Y. Tian and Y. Zhang, “A comprehensive survey on regularization strategies in machine learning,” *Information Fusion*, vol. 80, pp. 146–166, 2022.
- [129] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, “Learning from noisy labels with deep neural networks : A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [130] B. Frénay and M. Verleysen, “Classification in the presence of label noise : a survey,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.
- [131] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, “Deep learning with noisy labels : Exploring techniques and remedies in medical image analysis,” *Medical image analysis*, vol. 65, p. 101759, 2020.
- [132] X. Liang, X. Liu, and L. Yao, “Review—a survey of learning from noisy labels,” *ECS Sensors Plus*, vol. 1, no. 2, p. 021401, 2022.
- [133] B. Quost, T. Denoeux, and S. Li, “Parametric classification with soft labels using the evidential EM algorithm : linear discriminant analysis versus logistic regression,” *Advances in Data Analysis and Classification*, vol. 11, pp. 659–690, 2017.
- [134] Q. Nguyen, H. Valizadegan, and M. Hauskrecht, “Learning classification models with soft-label information,” *Journal of the American Medical Informatics Association*, vol. 21, no. 3, pp. 501–508, 2014.
- [135] Y. Xue and M. Hauskrecht, “Efficient learning of classification models from soft-label information by binning and ranking,” in *Proceedings of the... International Florida AI Research Society Conference. Florida AI Research Symposium*, vol. 2017, p. 164, NIH Public Access, 2017.

- [136] S. Xiao, N. Sang, X. Wang, and X. Ma, “Leveraging ordinal regression with soft labels for 3D head pose estimation from point sets,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1883–1887, IEEE, 2020.
- [137] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, “Regularizing neural networks by penalizing confident output distributions,” *arXiv preprint arXiv :1701.06548*, 2017.
- [138] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng, “Deep label distribution learning with label ambiguity,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2825–2838, 2017.
- [139] M. G. Bellemare, W. Dabney, and R. Munos, “A distributional perspective on reinforcement learning,” in *International conference on machine learning*, pp. 449–458, PMLR, 2017.
- [140] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv :1503.02531*, 2015.
- [141] J. C. Peterson, R. M. Battleday, T. L. Griffiths, and O. Russakovsky, “Human uncertainty makes classification more robust,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9617–9626, 2019.
- [142] A. Botchkarev, “Performance metrics (error measures) in Machine Learning regression, forecasting and prognostics : Properties and typology,” *arXiv preprint arXiv :1809.03006*, 2018.
- [143] L. Gordon and M. Hudson, “A characterization of the von mises distribution,” *The annals of statistics*, vol. 5, no. 4, pp. 813–814, 1977.
- [144] I. T. U. Radiocommunication, “Effects of building materials and structures on radiowave propagation above about 100 MHz,”
- [145] C. Zhao, W. Shi, and Y. Deng, “A new Hausdorff distance for image matching,” *Pattern Recognition Letters*, vol. 26, no. 5, pp. 581–586, 2005.
- [146] J.-B. Pagot, O. Julien, P. Thevenon, F. A. Fernandez, and M. Cabantous, “Signal quality monitoring for new GNSS signals,” *Navigation : Journal of The Institute of Navigation*, vol. 65, no. 1, pp. 83–97, 2018.
- [147] A. Žilinskas, “Practical mathematical optimization : An introduction to basic optimization theory and classical and new gradient-based algorithms,” *Interfaces*, vol. 36, no. 6, pp. 613–615, 2006.
- [148] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” *SIAM review*, vol. 60, no. 2, pp. 223–311, 2018.
- [149] D. Kim and J. Cho, “Improvement of anomalous behavior detection of GNSS signal based on TDNN for augmentation systems,” *Sensors*, vol. 18, no. 11, p. 3800, 2018.
- [150] Y. LeCun, P. Simard, and B. Pearlmutter, “Automatic learning rate maximization by on-line estimation of the hessian’s eigenvectors,” *Advances in neural information processing systems*, vol. 5, 1992.
- [151] S. Mallat, “Descente de gradient et rétro-propagation,” mars 2019. Cours Collège de France.
- [152] D. P. Kingma and J. Ba, “Adam : A method for stochastic optimization,” *arXiv preprint arXiv :1412.6980*, 2014.
- [153] G. Hinton, N. Srivastava, and K. Swersky, “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent,” *Cited on*, vol. 14, no. 8, p. 2, 2012.

Bibliographie

- [154] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization.,” *Journal of machine learning research*, vol. 12, no. 7, 2011.
- [155] Association Le Labo M, “Petite histoire des Captcha et de leurs côtés obscurs.” <https://asso.le-labo-m.fr/2020/06/petite-histoire-des-captcha-et-de-leurs-cotes-obscurs/>. Dernier accès : 2023-08-04.

Annexe A

Trajectoire d'un récepteur sur SE-Nav

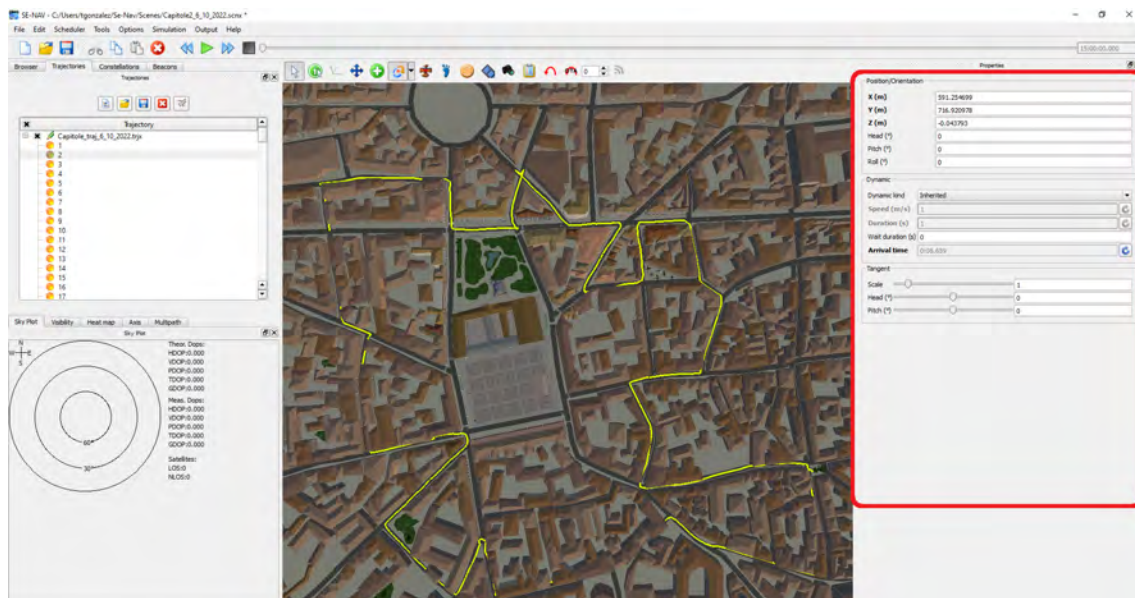


FIGURE A.1 – Interface à partir de laquelle s'effectue la construction des trajectoires du récepteur (jaune). Le tableau (rouge) pour la sélection d'un des points de la trajectoire donne des renseignements sur les coordonnées 3D du point dans le repère simulé et permet de régler la cinétique du récepteur.

Annexe B

Puissance du signal reçu

La puissance totale reçue P_{Rx} est la somme des différentes puissances totales reçues sur les polarisations circulaires gauche ($P_{Rx,L}$) et droite ($P_{Rx,R}$).

$$P_{Rx,R} = 10 \log \left(\frac{P_{Tx} G_{Rx,R} (G_{Tx,R} S_{RR}^2 + G_{Tx,L} S_{RL}^2)}{L_{FS} L_A L_{ET} L_{RT}} \right)$$

$$P_{Rx,L} = 10 \log \left(\frac{P_{Tx} G_{Rx,L} (G_{Tx,R} S_{LR}^2 + G_{Tx,L} S_{LL}^2)}{L_{FS} L_A L_{ET} L_{RT}} \right)$$

où

- P_{Tx} la puissance transmise,
- $G_{Tx,R}$ gain copolaire de l'antenne de transmission,
- $G_{R,L}$ gain de polarisation croisée de l'antenne de réception,
- $G_{R,R}$ gain copolaire de l'antenne de réception,
- $G_{Tx,L}$ gain de polarisation croisée de l'antenne de transmission,
- $S = \begin{pmatrix} S_{RR} & S_{RL} \\ S_{LR} & S_{LL} \end{pmatrix}$ la matrice de diffusion,
- L_{FS} perte en espace libre,
- L_A pertes atmosphériques,
- L_{ET} perte de câble au niveau du transmetteur,
- L_{RT} perte de câble au niveau du récepteur.

$$P_{Rx} = P_{Rx,R} + P_{Rx,L}$$

Annexe C

Influence de l'hyper-paramètre contrôlant la largeur de la distribution de probabilité dans le processus de *soft labelling*

TABLE C.1 – Influence de l'écart-type σ sur avg_{MAE} pour l'estimation du paramètre $\Delta\tau_{MP}$ en $10^{-2}T_c$.

σ	Random MP	Données SE-Nav	
		Toulouse	NYC
$\frac{w}{4}$	1.15 ± 0.18	0.67 ± 0.09	1.65 ± 0.52
$\frac{w}{2}$	0.96 ± 0.11	0.58 ± 0.06	1.32 ± 0.25
w	0.81 ± 0.11	0.64 ± 0.05	1.14 ± 0.19
$2w$	0.9 ± 0.13	0.65 ± 0.04	1.29 ± 0.19
$4w$	1.10 ± 0.14	0.74 ± 0.08	1.45 ± 0.33
$6w$	1.15 ± 0.25	0.67 ± 0.11	1.48 ± 0.19
$8w$	1.30 ± 0.20	0.81 ± 0.08	1.75 ± 0.40

TABLE C.2 – Influence de l'écart-type σ sur avg_{MAE} pour l'estimation du paramètre Δf_D en Hz.

σ	Random MP	Données SE-Nav	
		Toulouse	NYC
$\frac{w}{4}$	1.27 ± 0.16	0.82 ± 0.04	0.85 ± 0.09
$\frac{w}{2}$	1.14 ± 0.15	0.81 ± 0.10	0.83 ± 0.07
w	0.95 ± 0.08	0.79 ± 0.04	0.82 ± 0.11
$2w$	1.02 ± 0.10	0.78 ± 0.05	0.89 ± 0.09
$4w$	1.13 ± 0.15	0.72 ± 0.05	1.03 ± 0.15
$6w$	1.24 ± 0.11	0.70 ± 0.05	1.27 ± 0.16
$8w$	1.48 ± 0.19	0.70 ± 0.03	1.30 ± 0.17

TABLE C.3 – Influence de l'écart-type σ sur avg_{MAE} pour l'estimation du paramètre $\Delta\alpha_{MP}$ en % de l'amplitude de LOS.

σ	Random MP	Données SE-Nav	
		Toulouse	NYC
$\frac{w}{4}$	2.07 ± 0.20	1.47 ± 0.48	1.34 ± 0.29
$\frac{w}{2}$	1.52 ± 0.23	1.50 ± 0.36	1.04 ± 0.24
w	1.45 ± 0.19	1.02 ± 0.14	1.12 ± 0.34
$2w$	1.46 ± 0.25	1.28 ± 0.35	0.97 ± 0.26
$4w$	1.48 ± 0.22	1.17 ± 0.37	0.97 ± 0.19
$6w$	1.65 ± 0.21	1.30 ± 0.33	0.90 ± 0.20
$8w$	1.64 ± 0.19	1.25 ± 0.19	1.10 ± 0.20

TABLE C.4 – Influence de l'écart-type κ sur avg_{MAE} pour l'estimation du paramètre $\Delta\phi_{MP}$ en degrés.

κ	Random MP	Données SE-Nav	
		Toulouse	NYC
0.25	8.85 ± 0.96	5.70 ± 0.40	7.76 ± 0.90
0.5	6.89 ± 0.45	4.90 ± 0.72	6.62 ± 0.73
1	6.61 ± 0.63	4.83 ± 0.96	6.64 ± 1.38
2	6.00 ± 0.52	4.22 ± 0.90	6.69 ± 1.16
4	5.34 ± 0.37	4.76 ± 0.91	6.47 ± 1.42
8	4.78 ± 0.57	4.56 ± 0.81	6.14 ± 1.49
16	5.05 ± 0.82	4.92 ± 1.56	5.44 ± 0.88
32	5.14 ± 1.17	4.57 ± 1.10	6.31 ± 1.88
64	4.59 ± 0.58	4.64 ± 1.09	7.32 ± 4.59
128	4.28 ± 0.38	4.78 ± 1.19	5.88 ± 1.18
256	5.08 ± 1.67	4.70 ± 0.81	6.08 ± 1.15
512	5.07 ± 0.69	5.13 ± 1.96	5.54 ± 0.85
1024	6.23 ± 0.69	4.07 ± 0.55	6.11 ± 1.18
2048	6.38 ± 1.09	4.75 ± 1.14	6.49 ± 0.87
4092	4.90 ± 1.31	5.01 ± 0.97	6.51 ± 1.15
8196	9.10 ± 1.91	4.39 ± 1.07	6.63 ± 1.04
10000	8.56 ± 1.21	4.85 ± 1.24	6.50 ± 1.05

Annexe D

Notes explicatives sur les réseaux de neurones

Fonction d'activation

Un réseau de neurones est un modèle linéaire composé de nœuds d'entrée, intermédiaire et de sortie. Son caractère linéaire est conservé par combinaison linéaire. La juxtaposition de couches cachées linéaires supplémentaires revient à faire une combinaison linéaire de combinaisons linéaires. L'ajout de fonctions d'activation non-linéaires « brise » la linéarité du réseau de neurones, lui permettant d'apprendre le comportement complexe des données [73, 83, 81]. Quelques exemples de fonctions d'activation sont définis dans le tableau D.1. Une des propriétés des fonctions d'activation concerne leur différentiabilité en tout point réel. C'est une propriété intéressante dans le cadre des réseaux de neurones car le processus de rétropropagation de gradient au cours de l'apprentissage s'appuie sur le calcul de dérivées [147].

TABLE D.1 – Exemples de fonction d'activation utilisée en couche cachée.

ReLU (Unité Linéaire Rectifiée)	Sigmoïde
$ReLU : \mathbb{R} \rightarrow \mathbb{R}^+$ $x \mapsto \max(0, x)$	$sigmoïde : \mathbb{R} \rightarrow [0, 1]$ $x \mapsto \frac{1}{1+e^{-x}}$
Tangente hyperbolique	Leaky ReLU
$tanh : \mathbb{R} \rightarrow [-1, 1]$ $x \mapsto \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$leakyReLU : \mathbb{R} \rightarrow \mathbb{R}$ $x \mapsto \max(\epsilon x, x)$ avec $\epsilon > 0$

Remarque : La non-linéarité et la non-convexité de l'algorithme rendent la tâche de recherche de l'optimum global des paramètres du réseau θ_{opt} difficile voire impossible dans un temps raisonnable [148]. L'apprentissage est aussi plus difficile pour une profondeur de réseau plus importante, le nombre de poids et biais à régler devenant plus important [149].

Optimisation de la fonction de perte

L'apprentissage d'un réseau de neurones est assimilable à une recherche d'optimum dans un espace de recherche : on cherche $\theta_{opt} \in \Theta$ minimisant la moyenne des pertes des données du *mini-batch*. La moyenne des pertes pour un paramètre θ est calculée suite à un traitement *forward* des données par le réseau afin de générer les prédictions pour chaque donnée. Le minimum global, bien qu'optimal, est rarement atteint. On se contente généralement de trouver plusieurs minimaux locaux afin de les comparer : plusieurs séries d'apprentissage sont effectuées sur des ensembles d'entraînement, de validation et de test différents, des réseaux avec des poids et biais initiaux différents. Le mélange aléatoire des données permet d'éviter les effets de classement pathologique (pouvant nuire aux performances de la descente de gradient stochastique) [81]. L'initialisation différente des réseaux de neurones permet d'observer la capacité de l'algorithme à converger vers les minimums locaux.

La recherche du minimum sans l'information du gradient revient à effectuer un effort calculatoire en $O(\theta^3)$ avec θ les paramètres du réseau. Ce même effort en utilisant l'information contenue dans le gradient baisse la complexité en $O(\theta^2)$ [88].

Batch size

L'objectif de l'apprentissage est d'ajuster le paramètre θ afin que les prédictions de la fonction N_θ soit en moyenne les plus proches possibles des *labels* des données. Un processus d'entraînement se déroule selon un nombre d'*epochs*. Une *epoch* est défini comme étant le nombre de fois qu'une donnée $x \in D_T$ de l'ensemble d'apprentissage est traitée par l'algorithme (D_T l'ensemble d'entraînement du modèle). On pose $N_T = \text{Card}(D_T) \in \mathbb{N}^*$ le cardinal de l'ensemble d'entraînement. Le calcul de pertes est effectué sur échantillon de taille m :

- si $m = 1$, un seul échantillon de l'ensemble d'entraînement est utilisé pour le calcul de perte et de descente du gradient. L'avantage de cette méthode est la rapidité de son exécution très peu coûteuse. Elle introduit cependant beaucoup de bruit puisque l'échantillon réduit à une donnée n'est pas représentatif de l'ensemble de données d'entraînement. Pour le cas $n = 1$, on parle alors de descente de gradient stochastique (SGD),
- si $m = N_T$ le cardinal de l'ensemble des données d'entraînement. La descente de gradient calculée sera en moyenne optimale. Le principal désavantage concernant cette méthode réside dans le coût calculatoire qui est maximal puisque effectuée sur toutes les données d'entraînement. On parle alors de descente de gradient par *batch*,
- si $1 < m < N_T$ alors le calcul est effectué sur un échantillon fini de l'ensemble d'entraînement. Cette méthode concilie à la fois une rapidité de calcul car moins coûteux (le nombre d'*epochs* est cependant augmenté afin de s'assurer du traitement de toutes les données présentes dans l'ensemble d'entraînement) que les calculs pour la descente de gradient par *batch* [150] et une meilleure représentativité des données que la méthode de descente de gradient stochastique. On parle alors de descente de gradient par *mini-batch*. C'est cette dernière méthode qui est généralement appliquée aux algorithmes de *Machine Learning*. Tous les réseaux de neurones utilisés dans les travaux de cette thèse ont été entraînés sur des *mini-batch*.

Une itération correspond à un calcul de pertes sur un ensemble contenant m données. À la fin de chaque itération l'erreur est propagée et les poids et biais du réseau sont réactualisés. Si $m = N_T$ alors l'*epoch* contient un seul calcul de pertes, donc une seule itération et le réseau sera réactualisé à une seule reprise. Si $m = 1$ alors il y aura N_T calculs de pertes, soit N_T itérations avec une réactualisation du réseau après le traitement de chaque donnée. Si $1 < m < N_T$, il

y aura $\frac{N_T}{m}$ itérations, soit une mise à jour des poids et biais du réseau après le traitement de chaque *mini-batch*.

La taille du *mini-batch* est un hyper-paramètre traduisant le compromis entre complexité calculatoire et variance des gradients : plus m est grand et plus on diminue la variance du gradient calculé. On s'approche de la vraie descente de la moyenne du gradient et la convergence vers l'optimum. Cependant augmenter m accroît le nombre de calculs requis [151].

Remarques :

- Si m n'est pas un diviseur de N_T , alors les $N_m = \lfloor \frac{N_T}{m} \rfloor$ premiers calculs de *mini-batch* effectueront un calcul de pertes sur m données en posant $\lfloor a \rfloor$ la partie entière inférieure d'un réel $a \in \mathbb{R}$. La $N_m + 1$ ème itération calculera la perte sur $N_T - m \times \lfloor \frac{N_T}{m} \rfloor$ données et le réseau aura réajusté ses paramètres $N_m + 1$ fois,
- Dans le cas du *mini-batch*, le calcul des pertes devient une minimisation du risque empirique, les mesures de pertes étant effectuées sur un échantillon de données extraites de l'ensemble d'entraînement [81, 80].

Minimisation des pertes par rétropropagation du gradient de l'erreur

L'ajustement des poids et biais du réseau s'effectue par rétropropagation du gradient. Il s'agit de calculer les pertes issues du *mini-batch* et de corriger les poids et biais en fonction du gradient calculé. Pour un *mini-batch* $D_m \subseteq D_T$, avec $m = \text{Card}(D_m)$ et $E(\theta)$ le calcul de pertes du réseau de paramètre θ sur D_T , l'ajustement du vecteur de poids à une itération $i \in \mathbb{N}$ d'une *epoch* $e + 1$ est défini comme étant [88] :

$$\theta_e^{i+1} = \theta_e^i - \eta \nabla E(\theta_e^i) \quad (\text{D.1})$$

avec

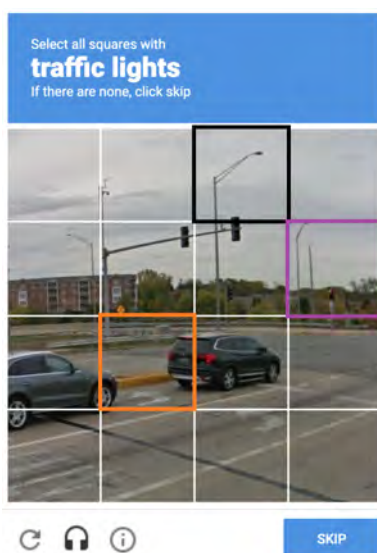
- ∇E le gradient de E ,
- $\eta > 0$ le pas d'apprentissage,
- θ_e^i le vecteur de poids du réseau à l'itération $i \in \llbracket 1 ; N_m \rrbracket$ de l'*epoch* $e \in \mathbb{N}^*$.

Le pas d'apprentissage η est un hyper-paramètre ajustant la correction des pondérations lors de la rétropropagation du gradient. Il contrôle le compromis entre vitesse de convergence entre le point de départ et un minimum (le plus souvent local) et précision du résultat. Un pas d'apprentissage trop élevé peut provoquer des oscillations autour du minimum, pouvant aboutir en une divergence de la fonction *loss*. A l'inverse, un pas d'apprentissage trop faible aura tendance à allonger le temps de convergence de l'algorithme. Le pas d'apprentissage n'est pas toujours fixe et peut évoluer pendant l'apprentissage : en fonction de l'*epoch*, des valeurs de poids, de perturbations aléatoires [150] ou d'autres paramètres : c'est le cas pour des techniques d'optimisation comme *Adaptive Moment Estimation* (Adam) [152], *Root Mean Square Propagation* (RMS-Prop) [153] ou *Adaptive Gradient Descent* (AdaGrad) [154]. La correction des poids s'effectue selon la direction opposée au gradient maximum de l'erreur moyenne calculée sur D_m . Avec une erreur initiale pouvant être grande en raison de l'initialisation aléatoire de θ , chaque exemple traité va permettre d'ajuster les poids dans la direction minimisant l'erreur moyenne. Répétée un certain nombre d'itérations, le paramètre θ du réseau tend à minimiser la fonction de perte [73].

L'ajustement du paramètre θ est donc effectué au cours d'une boucle d'entraînement sur un nombre $N_e \times N_i$ avec N_e le nombre d'*epochs* d'apprentissage et N_i le nombre d'itérations dans une *epoch*.

Annexe E

Robustesse envers un apprentissage avec *labels* ambigus



Le test Captcha est un dispositif de sécurité informatique déployé sur les sites web afin de les protéger d'attaques par robots. Ce test consiste en la *labellisation* de plusieurs zones d'une image (figure E.1). Afin d'établir le *label* des différentes zones, il y a une collecte massive d'annotations effectuées manuellement par des utilisateurs [122]. Cela peut poser problème dans le cas d'images ambiguës, comme celle présentée en figure E.1 où les sections encadrées présentent des ambiguïtés. Par exemple la section d'encadrement noir contient un petit morceau du haut du feu tricolore, qui ne sera pas considéré comme un feu tricolore par certains utilisateurs. En fonction de ceux-ci, cette section sera différemment *labellisée* et sera sujette à imprécision.

FIGURE E.1 – Test captcha : sélection des sections contenant des feux tricolores. On remarque que certaines sections présentent une ambiguïté au niveau de la présence d'un feu tricolore [155]. La section noire ne contient qu'un petit morceau du haut du feu, presque indiscernable. La section mauve contient un feu tricolore en arrière-plan de l'image. La section orange contient uniquement le bas du poteau du feu de signalisation.

Annexe F

Comparaison de performances entre les distributions Gaussienne tronquées et selon une loi de Von Mises

Nous reprenons l'exemple des images de compas cette fois-ci sur un ensemble de données de 1 000 images de *labels* $y \in [0, 360]$ afin de comparer les performances de prédiction d'un CNN-HL faisant appel à des densités de probabilité construites selon une Gaussienne tronquée avec un CNN-HL utilisant des densités de distributions suivant une loi de Von Mises. L'histogramme des *soft labels* est construit sur le support $[0, 2\pi]$ avec une subdivision du support en $k = 20$ bins. 10 *runs* ont été effectuées avec un renouvellement des ensembles d'entraînement, validation et généralisation entre chaque *run*. L'entraînement s'est déroulé sur 30 *epochs* avec une taille de *batch* égal à 100. Les performances obtenues par les différents CNN sont résumées dans le tableau F.1. Pour le CNN effectuant de la régression sans *soft labelling*, la *Mean Square Error* (MSE) est utilisée pour comparer les *labels* et les prédictions obtenues.

TABLE F.1 – Erreur d'estimation du cap en degrés par le CNN-Reg et deux distributions distinctes : Gaussienne tronquée et loi de Von Mises.

MAE	CNN-Reg	CNN-HL Gaussienne tronquée	CNN-HL Von Mises
Erreur moyenne minimum	0.76°	4.37°	0.74°
Erreur moyenne maximum	79.63°	7.72°	1.15°
avg_{MAE}	25.10°	5.52°	0.91°
Moyenne de l'écart-type	33.25°	0.92°	0.12°

Les performances d'estimation de cap montrées dans le tableau E.1 mettent en évidence le gain de performances apporté par l'utilisation d'une distribution de Von Mises. Les deux techniques HL permettent au CNN d'obtenir de meilleures performances comparées au cas sans *soft labelling* : le CNN-Reg montre une moyenne de performances de 25.10° contre 5.52° et 0.91° pour les deux distributions testées. Le passage en *soft labelling* permet de diviser par quatre l'erreur moyenne effectuée sur les estimations. Avec Von Mises, la différence est d'autant plus prononcée puisqu'en moyenne l'erreur est 27 fois moins importante que le CNN-Reg, et 6 fois moins importante que le CNN-HL avec Gaussienne tronquée.

