

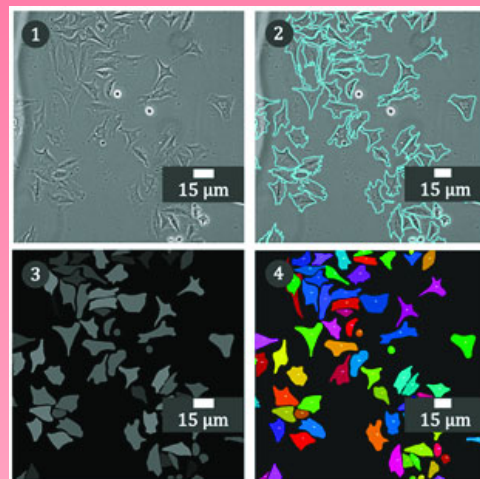
La Revue 3EI



Ressource publiée sur Culture Sciences de l'Ingénieur : <https://eduscol.education.fr/sti/si-ens-paris-saclay>



Journée 3EI
23 juin (voir page 3)



*Intelligence
Artificielle
et
Génie
Électrique
1^{ère} partie*

Publication trimestrielle du Cercle Thématique 13.01 de la SEE

ENSEIGNER L'ÉLECTROTECHNIQUE ET L'ÉLECTRONIQUE INDUSTRIELLE



Société de l'Électricité, de l'Électronique
et des Technologies de l'Information
et de la Communication

N°108 – Avril 2022

Abonnez-vous à *La Revue* 3E.I

- > Outil didactique pour les filières préparant à l'enseignement technique
- > Trame indispensable à la formation continue des hommes et des femmes de terrain
- > 4 numéros par an : janvier, avril, juillet, octobre
- > 3 dossiers complets par an sur des sujets d'actualité + 1 N° Pratique pédagogique

Choisissez votre formule d'abonnement pour 2022:



Version papier

4 numéros : janvier, avril, juillet, octobre.
Distribution postale

France & UE	Hors UE
<input type="checkbox"/> 40 € TTC	<input type="checkbox"/> 59,18 € HT*



Version numérique

Accès aux publications numériques
ouvert pendant un an à compter de la date du paiement

France & UE	Hors UE
<input type="checkbox"/> 30 € TTC	<input type="checkbox"/> 29,38 € HT



Version duo

Version imprimée
+ version numérique

France & UE	Hors UE
<input type="checkbox"/> 50 € TTC	<input type="checkbox"/> 68,97 € HT*

Votre adhésion à la SEE

<input type="checkbox"/> Standard	<input type="checkbox"/> Retraité <input type="checkbox"/> Enseignant <input type="checkbox"/> Jeune actif (< 35 ans)	<input type="checkbox"/> Etudiant <input type="checkbox"/> En recherche d'emploi
125 € TTC	65 € TTC	15 € TTC

+ Votre abonnement 3E.I (Tarif réservé aux adhérents, version papier)

France & UE: 32 € TTC Hors UE: 51,34 € HT*

TVA de la revue 3E.I: 2,1%. Adhésion collective possible via des conventions de partenariat - Contactez-nous à : sg@see.asso.fr
* Prix final incluant des frais de transports de 20 €

Adresse de livraison		Adresse de facturation (Si différente)	
Nom* :		<input type="checkbox"/> Je joins le bon de commande administratif N° <input type="text"/> et je désire recevoir une facture au nom de mon employeur pour paiement à réception	
Prénom* :		Raison sociale de l'employeur :	
Adresse* :		Service : <input type="text"/> Activité (facultatif) : <input type="text"/>	
Code postal* :	Pays* :	Adresse :	
Ville* :		Code postal : <input type="text"/> Ville : <input type="text"/>	
Tél.* :		Pays : <input type="text"/> N° TVA : <input type="text"/>	
e-mail* :		N° TVA intracommunautaire : obligatoire pour règlement HT en UE hors de France e-mail* : <input type="text"/>	
*Obligatoire			
Votre règlement			
Je règle la somme de <input type="text"/> €			
par <input type="checkbox"/> Chèque à l'ordre de la SEE			
<input type="checkbox"/> Virement après réception de la facture <input type="checkbox"/> Carte bancaire (Visa, Eurocard/Mastercard)			
N° Carte <input type="text"/>			
Date de validité <input type="text"/>	N° cryptogramme <input type="text"/>	(3 derniers chiffres au dos de la carte)	
*Obligatoire			

BULLETIN À COMPLÉTER ET RENVOYER À : SEE - 17 rue de l'Amiral Hamelin - 75116 Paris - France
Tél. +33(0)156 90 37 17- abo@see.asso.fr

ABONNEMENT PLUS RAPIDE: www.see.asso.fr

Je consens à recevoir les autres diffusions de la SEE & de ses activités (congrès, soirées débats, revues, etc.) qui sont extérieures aux diffusions liées à mon abonnement.

Conformément aux dispositions légales et réglementaires en matière de données personnelles, les informations recueillies sur ce formulaire sont enregistrées dans un fichier informatisé par la SEE (Société de l'électricité, de l'électronique et des technologies de l'information et de la communication) pour la mise en place et le suivi de l'abonnement souscrit ainsi que pour l'envoi de courriers, e-mails de réabonnements. Elles sont conservées et sont destinées à être utilisées par la SEE et les prestataires techniques de la SEE afin de permettre la bonne réception du magazine et d'assurer le service client. Vous pouvez exercer votre droit d'accès aux données vous concernant par courrier : SEE - Service abonnements 17 rue de l'Amiral Hamelin 75116 Paris ou par le formulaire de contact du site web : www.see.asso.fr. Offre valable du 01/10/2021 au 30/09/2022 dans la limite des quantités disponibles.





SOCIÉTÉ de l'ELECTRICITE, de l'ELECTRONIQUE et des TECHNOLOGIES de l'INFORMATION et de la COMMUNICATION.

17, rue de l'Amiral Hamelin, 75116 PARIS
Tél : 01 56 90 37 17
site web : www.see.asso.fr

La Revue 3EI
publication trimestrielle
de la SEE

SEE, association reconnue d'utilité publique par le décret du 7 décembre 1886
Siret 785 393 232 00042, APE 9412 Z, n° d'identification FR 44 785 393 232

3EI : Enseigner l'Électrotechnique et l'Électronique Industrielle

<p>La Revue 3EI, Édition SEE, 17 rue de l'Amiral Hamelin 75116 PARIS</p> <p>Directeur de la publication François GERIN Président de la SEE</p> <p>Rédacteur en Chef Franck LE GALL</p> <p>Adresser les propositions d'article à : revue3ei@gmail.com</p> <p>Communication : Mme. Mélisande DE LASSENCE Communication1@see.asso.fr 01 56 90 37 17</p> <p>Promotion et Abonnements : (4 numéros par an) Janvier, Avril, Juillet, Octobre. Tél : 01 56 90 37 17 abo@see.asso.fr</p> <p>Tarifs 2022 :</p> <p>Version PAPIER :</p> <p>France et UE (TTC)..... 40 € Pays hors UE (HT)..... 49,18 €</p> <p>Version NUMERIQUE :</p> <p>France et UE (TTC)..... 30 € Pays hors UE (HT)..... 29,38 €</p> <p>Version DUO (Papier+Num.) :</p> <p>France et UE (TTC)..... 50 € Pays hors UE (HT) 58,97 €</p> <p>Impression : Dupliprint 733 rue Saint-Léonard 53100 Mayenne Tel : 02 43 11 09 00 Couv : O.P. : All. – TFR : 0 – C. : PEFC Corp. : O.P. : Esp. – TFR : 0 – C. : PEFC</p> <p>Dépôt Légal : Avril 2022</p> <p>Commission Paritaire 1222 G 78028 ISSN 1252-770X</p>	<p style="text-align: right;">Sommaire du n° 108</p> <p>p. 2 <i>Éditorial,</i></p> <p style="text-align: center;">Thème : Intelligence Artificielle et Génie Electrique</p> <p>p. 3 <i>Introduction à l'apprentissage automatique</i> Ludovic de Mattéis, Solal Nathan</p> <p>p. 14 <i>Introduction à l'apprentissage profond</i> Steeven Janny, Ludovic de Mattéis, Wenqi Shu-Quartier</p> <p>p. 26 <i>Stabilisation d'un pendule inversé à l'aide d'un apprentissage par renforcement</i> Guenolé Cherot</p> <p>p. 37 <i>Apprentissage supervisé – Comportement aérien d'un drone</i> Olivier Tourvieille</p> <p>p. 45 <i>Suivi Cellulaire par traitement d'images et apprentissage</i> Mélanie Pietri</p> <p>p. 50 <i>L'intelligence artificielle au service de la mobilité urbaine à Rennes</i> par LACROIX Group Reynholds Reinette</p> <p style="text-align: center;">Hors Thème :</p> <p>p. 53 <i>Le soleil se lève à l'ouest !</i> Arnaud Brugier</p> <p>p. 64 <i>Réseaux de distribution : Compensation, Filtrage, Amortissement</i> Jacques Courault</p> <p>p. 75 <i>Le résonateur piézoélectrique pour l'électronique de puissance</i> E. Bigot</p>
--	--

Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans la présente édition, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective et, d'autre part, les analyses et courtes citations justifiées par le caractère scientifique ou d'information de l'œuvre dans laquelle elles sont incorporées. Toutefois des copies peuvent être utilisées avec l'autorisation de l'éditeur. Celle-ci pourra être obtenue auprès du Centre Français du Droit de Copie, 20, rue des Grands Augustins, 75006 Paris, auquel la Revue 3EI a donné mandat pour la représenter auprès des utilisateurs. (loi du 11 mars 1957, art.40 et 41 et Code Pénal art. 425).

Numéro 108 de la revue 3EI

« Thème : IA et Génie Electrique »

L'intelligence artificielle (IA), comme la robotique, est une discipline aux contours peu précis. Si on laisse de côté l'intelligence artificielle forte, encore lointaine, qui vise à simuler un comportement humain dans l'ensemble des champs de la connaissance et des interactions, l'intelligence artificielle dite faible est une famille de méthodes utilisées pour résoudre des problèmes.

Le terme intelligence vient de la ressemblance avec le raisonnement humain. Avant les années 2010, les systèmes dit intelligents calculaient rapidement des solutions aux problèmes à partir de lois comportementales codées par des ingénieurs spécialistes du problème en question. Cela allait du jeu d'échec électronique pour débutant au système de délestage du réseau électrique.

Depuis l'apparition des premiers neurones artificiels inspirés de nos neurones biologiques, la ressemblance avec l'intelligence humaine est descendue au niveau de l'apprentissage. Avec des algorithmes complexes pour optimiser l'apprentissage de centaines de couches composées de milliers de nœuds, les réseaux de neurones modernes se sont éloignés de leur modèle biologique et offrent aujourd'hui des perspectives de progrès dont on peine à envisager les limites.

Chaque avancée de l'IA, ses victoires aux échecs puis au jeu de go, ses créations musicales ou picturales, ses traductions instantanées, ses diagnostics médicaux interrogent sur les métiers qui lui seraient inaccessibles et sur la nature de l'intelligence humaine.

En permettant de traiter des systèmes complexes difficilement descriptibles par des équations ou des arbres (images, voix humaines, ...), les réseaux de neurones sont un outil très prometteur également pour les sciences de l'ingénieur, que ce soit en les utilisant dans leurs champs de prédilection (le traitement d'images, de la voix, du langage) ou directement pour le contrôle de machines ou la modélisation de matériaux. Cela explique leur apparition dans les formations d'ingénieur, notamment dans le programme de sciences de l'ingénieur de CPGE, et le thème de ces 2 numéros de la revue 3EI (avril et juillet 2022).

L'objectif de ces articles est de donner les bases nécessaires pour mettre en œuvre les algorithmes de l'IA développés dans les laboratoires d'IA. Cela comprend des articles de présentation des concepts (introduction à l'intelligence artificielle, les réseaux de neurones, l'apprentissage par renforcement, les réseaux récurrents et la prédiction), des travaux pratiques (utilisant Matlab pour certains, des bibliothèques python pour d'autres) et des exemples d'applications de l'IA dans l'industrie et dans la recherche.

« Hors Thème »

En complément du thème de l'IA que nous retrouverons dans le numéro de juillet de la revue 3EI, nous vous proposons trois articles hors thème qui devraient satisfaire la curiosité de chacun d'entre nous.

L'article d'Arnaud Brugier au titre évocateur « le Soleil se lève à l'Ouest » expose en détails les nombreux projets photovoltaïques qui voient le jour tout au bout du monde dans la Presqu'île de Crozon. Participant à la sécurisation de l'approvisionnement énergétique, ces projets permettent de dynamiser le territoire et sont créateurs d'activité. Ils illustrent aussi des politiques publiques volontaristes et pragmatiques de réduction des consommations, de développement des ENR et d'utilisation des ressources locales sur ce territoire enclavé et ce, tout en favorisant l'implication citoyenne et les échanges. Une belle expérience qui donnera peut-être envie dans d'autres territoires...

Faisant suite à son article sur les réseaux de transport publié dans le numéro de janvier, M. Courault traite ici des réseaux de distribution. Il y présente les contraintes et les solutions développées pour assurer la qualité de fourniture au client.

Nous clôturons ce numéro d'avril avec l'article de M. Bigot qui présente l'intérêt de l'emploi de résonateurs piézoélectriques dans le domaine de la conversion de puissance. Le modèle équivalent du résonateur piézoélectrique et les contraintes associées à son emploi sont présentés avant de traiter un exemple d'application à l'aide du logiciel LTSpice.

Le Comité de Publication de la Revue 3EI

Faites connaître notre revue
Vous en assurez la pérennité

La Revue 3EI

Comité de publication

Morgan ALMANZA

Hamid BEN AHMED

Arnaud BRUGIER

Jacques COURAULT

Jean FAUCHER

Gilles FELD

Jean Michel GAY

Jean-Philippe ILARY

Anthony JUTON

Chérif LAROUCI

Marie-Michèle LE BIHAN

Franck LE GALL

Denis LABROUSSE

Pascal LOOS

Marc PETIT

Sylvain PIETRANICO

Oviglio SALA

Jean-François SERGENT

Jean-Claude VANNIER

Journée 3EI

Journée du DER Nikola TESLA

23 et 24 juin 2022

Ecole Normale Supérieure de Paris-Saclay

Présentation

Les journées conjointes de la revue 3EI et du Département d'Enseignement et de Recherche Nikola TESLA de l'ENS Paris-Saclay sont l'occasion de rencontres entre les différents acteurs du génie électrique et de la science de l'information pour amorcer une réflexion sur la place de nos formations dans le paysage actuel et avec les mutations de la société.

Ces deux journées auront lieu les jeudi 23 juin (pour la Journée 3EI) et vendredi 24 juin (pour la journée du DER Nikola TESLA) à l'Ecole Normale Supérieure de Paris-Saclay.

Programme thématique des journées (Prévisionnel)

Jeudi 23 juin 2022	Journée 3EI	Vendredi 24 juin 2022	DER Nikola TESLA
<p><u>Matinée : Plénière</u></p> <p>9h00 - 9h30 : Accueil et Café</p> <p>9h30 - 10h15 : Dominique Bellec</p> <p>10h15 - 11h00 : Les moyens de choisir son chemin par Hugo Descoubes, PRAG à l'ENSI Caen</p> <p>11h00 - 11h30 : Enseignement MOOC - STEEVE par François Louf et Javier Ojeda, ENS Paris-Saclay</p> <p>11h30 - 12h00 : Retour d'expériences sur l'enseignement à l'EPFL par Morgan Almanza, ENS Paris-Saclay</p> <p>12h00 - 12h30 : Session spéciale enseignant</p>		<p><u>Matinée : Plénières</u></p> <p>9h00 - 9h30 : Accueil et Café</p> <p>9h30 - 10h15 : Sylvie Retailleau, Présidente de l'université Paris-Saclay</p> <p>10h15 - 11h00 : Samson Lasaulce, Directeur de recherche CNRS - CentraleSupelec</p> <p>11h00 - 11h45 : Erwan Salaün, Directeur du centre d'excellence systèmes de Navigation, Safran Electronics & Defense</p> <p>11h45 - 12h30 : Thierry Ménard,</p>	
<p><u>Après-midi : Tables rondes thématiques</u></p> <p>14h00 - 15h30 : Outils pour l'enseignement à distance hybridation par Hervé Discours et l'IUT de Cachan</p> <p>14h00 - 15h30 : Evaluation de l'enseignement par Véronique Depouton, Professeur d'Anglais en charge de l'évaluation des enseignements, Paris-Saclay</p> <p>14h00 - 15h30 : IA et enseignement par Anthony Juton, PRAG ENS Paris-Saclay</p> <p>15h30 - 16h30 : Retour des tables rondes</p> <p>17h00 : Visite des salles de concours (Sur inscription)</p>		<p><u>Après-midi : Tables rondes thématiques</u></p> <p>14h00 - 14h30 : Présentation du Département d'Enseignement et de Recherche (DER) Nikola TESLA par Thomas Rodet, ENS Paris-Saclay</p> <p>14h30 - 16h00 : Table ronde thématique 1</p> <p>14h30 - 16h00 : Table ronde thématique 2</p> <p>16h00 : Visite du DER Nikola TESLA</p>	

Animations

Repas : 12h30 - 14h00 : Repas au Crous de l'ENS Paris-Saclay

Table ronde convivialité : Rencontres conviviales entre les participants aux journées

Concert LabOrchestra : Représentation d'un orchestre composé d'étudiants et d'un chef d'orchestre Nicolas Perrin à partir d'instruments pédagogiques détournés

Inscription aux journées

Pour l'instant les inscriptions ne sont pas activées, mais consulter le lien suivant qui vous permettra de vous inscrire : <https://ojedajavier.github.io/journees3eidertesla/>

INTRODUCTION À L'APPRENTISSAGE AUTOMATIQUE

LUDOVIC DE MATTEIS¹, SOLAL NATHAN²

¹ : élève du département Nikola Tesla de l'ENS Paris-Saclay, auteur de l'article "Personalized Human-Swarm Interaction Through Hand Motion" issu de son stage à l'EPFL sur l'apprentissage des mouvements intuitifs de l'utilisateur pour le contrôle d'une flotte de drone. ludovic.de_matteis@ens-paris-saclay.fr

² : élève du département Nikola Tesla de l'ENS Paris-Saclay, étudiant du M2 IA de l'Université Paris-Saclay, actuellement en stage au LISN (Laboratoire Interdisciplinaire des Sciences du Numérique) sur "Combatting Unemployment : Data, Recommendations and Fairness". solal.nathan@ens-paris-saclay.fr

Résumé : Cet article introduit le dossier sur l'intelligence artificielle en définissant les contours de l'IA puis, après un rappel des dates marquantes de sa courte histoire, recentre le propos sur l'apprentissage automatique en différenciant apprentissage supervisé, apprentissage non-supervisé et apprentissage par renforcement. Les réseaux de neurones et l'apprentissage par renforcement faisant l'objet d'articles propres, cette introduction illustre l'apprentissage supervisé par 2 algorithmes classiques, régression linéaire et K plus proches voisins, de même que l'apprentissage non supervisé avec l'algorithme des k-moyennes, ceci notamment afin de mettre en évidence l'importance des hyper-paramètres.

I/ Introduction

L'intelligence artificielle (IA) a pour but de créer des entités intelligentes. Dans ce sens, il y a une poursuite de cette capacité dont l'humain est maître, à résoudre des problèmes. « La capacité à résoudre des problèmes » est d'ailleurs une définition de l'intelligence que l'on retrouve régulièrement. Il n'existe pas de définition qui fasse consensus, bien que des chercheurs ont tenté de définir formellement l'intelligence. L'humain tente de répliquer son intelligence pour pouvoir automatiser certaines tâches.

Il est important de noter qu'il y a une différence forte entre l'IA et l'apprentissage automatique (*machine learning*). L'apprentissage automatique est une forme d'IA qui consiste en un système qui s'améliore grâce à l'expérience alors que l'IA peut être un simple ensemble de règle et d'heuristiques.

II/ Qu'est-ce que l'intelligence artificielle

L'intelligence artificielle (IA) est un domaine de l'informatique et des mathématiques rassemblant un ensemble de techniques algorithmiques et de théories permettant de réaliser des machines imitant l'intelligence humaine. Son but est de reproduire l'intelligence afin d'être capable de résoudre des problèmes complexes. Cet objectif passe par la modélisation de l'intelligence humaine en tant que phénomène, comme cela pourrait être fait dans le domaine de la physique, de la chimie ou de la biologie. L'intelligence artificielle est un domaine en pleine expansion, trouvant sa théorie et ses applications dans de très nombreux domaines, comme la théorie des probabilités, les neurosciences, la robotique, la théorie des jeux, la santé et les transports.

L'intelligence artificielle est généralement classée comme forte ou faible.

- L'IA forte a pour objectif de développer une machine disposant de capacités cognitives proches de l'humain. Cette machine serait capable de trouver seule une solution à un problème qui lui est encore inconnu. Bien que ce soit un sujet de recherche actif, l'IA forte n'existe pas à l'heure actuelle.

- L'IA faible, quant à elle consiste à développer des machines permettant de répondre à un problème précis. Elle dispose alors de données et d'aide humaine, et doit être capable de simuler le raisonnement humain, sans pour autant comprendre les phénomènes mis en œuvre.

Le domaine de l'intelligence artificielle est scindé en plusieurs sous-domaines imbriqués (présentés figure 1). L'apprentissage automatique est un sous-domaine de l'intelligence artificielle consistant à apprendre par l'expérience ou par une base de données des règles implicites pour répondre à un problème donné. Ce domaine s'oriente spécialement autour de l'analyse statistique de données d'entraînement. Dans le cadre de l'apprentissage automatique, nous verrons qu'il existe de nombreux algorithmes utilisant des modèles mathématiques variés. Le réseau de neurones est un de ces modèles, certainement le plus répandu et celui utilisé dans les domaines les plus diverses. L'apprentissage profond est un ensemble de techniques tirant parti des réseaux de neurones pour résoudre des problèmes complexes. Ces techniques sont très utilisées, notamment dans le domaine du traitement d'image, le traitement de séries temporelles (reconnaissance vocale...).

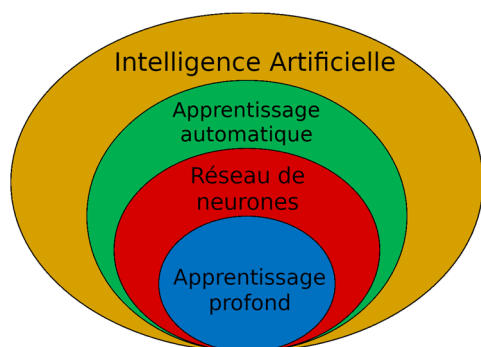


Figure 1 : Schéma de décomposition du domaine de l'intelligence artificielle et de ces sous-domaines

Toutes les techniques d'intelligence artificielles ne sont pas forcément de l'apprentissage automatique. Il est notamment possible de fournir directement à la machine les règles de prédiction sous forme d'une suite de conditions si-alors. Ces systèmes, appelés systèmes experts ont été très utilisés durant les années 1980 pour résoudre avec une grande précision des tâches données. Dans cet article ne nous intéresserons pas à ces systèmes experts et considérerons uniquement les techniques d'apprentissage automatique.

III/ Historique de l'intelligence artificielle

L'idée d'intelligence artificielle prend ses sources dans la première moitié du 20ème siècle. Dans l'article *Computing Machinery and Intelligence*, publié en 1950, le mathématicien britannique Alan Turing pose la question « les machines peuvent-elles penser ? ». Dans ce même article, il décrit le comportement des machines (ordinateurs au sens moderne du terme) comme un « jeu de l'imitation » du raisonnement humain. Il définit entre autres le test de Turing, censé déterminer si la machine parvient à simuler le comportement humain. À la même époque et indépendamment, le mathématicien américain Warren Weaver publie un mémo sur la traduction automatique des langues mentionnant l'idée qu'une machine pourrait reproduire un comportement relevant de l'intelligence humaine. A cette même époque, le neurologue américain Warren McCulloch publie un article sur le fonctionnement des neurones et modélise le fonctionnement d'un neurone par un circuit électrique simple. Cet événement marque le début de la recherche sur les réseaux de neurones artificiels. Cependant, les limites technologiques ne permettent pas de fort développement des réseaux de neurones avant de nombreuses années.

L'intelligence artificielle devient officiellement reconnue comme un domaine scientifique en 1956 lors d'une conférence au Dartmouth College aux Etats-Unis. Un des premiers programmes d'intelligence artificielle est présenté en 1959 par Arthur Samuel, un informaticien américain. Il présente un programme jouant aux dames, capable de s'améliorer au fil des parties, battant le quatrième meilleur joueur des Etats-Unis. Le terme d'apprentissage automatique (*machine learning*) est utilisé pour la première fois lors de la présentation de ce programme.

Cependant, après les années 1950 et 1960, marquées par un fort engouement pour le domaine de l'IA, les avancées n'aboutissent pas et les gouvernements réduisent les financements pour les projets d'intelligence artificielle, privilégiant des domaines donnant des débouchés concrets plus rapidement. Cette période de ralentissement du domaine appelée « l'Hiver de l'IA » durera jusqu'au années 1980.

Le développement des performances informatiques et l'essor d'Internet durant le début des années 1990 permet de relancer le domaine de l'intelligence artificielle. L'arrivée de grandes quantités de données (Big Data) permet un entraînement plus précis des systèmes d'IA. Ces systèmes sont alors développés plus globalement et utilisés dans de nombreux domaines comme les diagnostics médicaux. C'est également à cette époque que se développent sérieusement les premiers algorithmes d'apprentissage profond (*deep learning*), en particulier les réseaux de neurones artificiels.

L'intelligence artificielle profite d'une véritable sortie médiatique à partir de 1997 quand l'ordinateur Deep Blue, développé par IBM, bat le champion du monde d'échecs Garry Kasparov.

Depuis cette époque, l'IA s'intègre dans la société, poussée notamment par l'augmentation de la connectivité informatique mondiale. Les projets de recherche en intelligence artificielle se multiplient, devant plus complexes et s'étendant à de nouveaux domaines, comme la vision par ordinateur et la robotique.

Certaines organisations telles que ImageNet, Human Connectome Project et OpenIA contribuent au développement de l'IA en proposant des bases de données et des algorithmes de bases permettant à tous d'appréhender les concepts de l'intelligence artificielle. De 2010 à 2017, ImageNet organise le concours ILSVRC (ImageNet Large Scale Visual Recognition Challenge), consistant en la conception de logiciels classifiant des objets et des scènes dans des images naturelles. Ces compétitions auront permis de fournir de très importants jeux de données encore utilisés pour tester de nouveaux algorithmes d'apprentissage automatique.

Depuis les années 2010, les réseaux de neurones reviennent sur le devant de la scène, poussés par l'augmentation de la puissance de calcul et par la mise à disposition de bases de données plus importantes, favorisant l'utilisation des algorithmes d'apprentissage profond. Les techniques d'apprentissage profond imposent rapidement leur suprématie dans le domaine de la classification d'image. Basé également sur des réseaux de neurones comme modèles de prédiction, l'ordinateur AlphaGo bat en 2017 le champion du monde de jeu de Go (alors considéré comme le jeu le plus complexe). L'ordinateur AlphaZero (successeur d'AlphaGo) est alors capable dès fin 2017 de surpasser les joueurs humains des jeux les plus complexes (Go, Echecs, Shogi...) et ceci en apprenant uniquement seul.

De nos jours, l'intelligence artificielle trouve des applications dans des domaines très variés. On retrouve

ainsi ces algorithmes utilisés pour le traitement d'image, le traitement automatique du langage (chatbot, GPT3...), la reconnaissance vocale (Alexa, Siri, Cortana, Rasa...) et pour la traduction automatique (pourtant hors de portée des méthodes traditionnelles). De nombreux algorithmes sont également utilisés pour de la prédiction d'évolution temporelle de phénomènes complexes (cours de la bourse, épidémiologie...).

IV/ Qu'est-ce que l'apprentissage automatique

IV.1/ Définition

Le domaine de l'intelligence artificielle a pour objectif de parvenir à simuler l'intelligence humaine et en particulier l'apprentissage de nombreuses tâches. Deux méthodes sont alors possibles pour apprendre :

- L'apprentissage par cœur consiste à mémoriser explicitement tous les exemples possibles afin de pouvoir les restituer
- L'apprentissage par généralisation a pour objectif d'extraire des règles implicites à partir d'une quantité d'exemples afin de les réappliquer à de nouvelles situations jamais rencontrées.

L'apprentissage par cœur est relativement aisé pour une machine à condition de disposer des exemples. En revanche, l'apprentissage par généralisation est difficile car il demande d'extraire des règles qui ne sont pas explicitement mentionnées dans les exemples. Ce défi constitue le cœur l'apprentissage automatique.

Comme expliqué précédemment, l'apprentissage automatique est un champ de l'IA porté sur l'analyse statistiques de données d'apprentissage. Historiquement, cette branche est définie comme le développement de machines capable d'apprendre sans avoir été explicitement programmée à apprendre une tâche.

IV.2/ Exemple

Figure 2 : Exemple de chiffres manuscrits

Extrait de la base de données MNIST, couramment utilisée en apprentissage automatique

Un exemple répandu d'apprentissage automatique est la reconnaissance automatique des chiffres manuscrits. Des exemples de ces chiffres sont présentés figure 1. Ces caractères présentent une grande diversité en forme, orientation, épaisseur de traits... Il est difficile dans ce cas d'écrire une liste explicite de règles permettant de discriminer les chiffres. On va donc

chercher à mettre en œuvre un algorithme permettant d'extraire des règles implicites en se basant sur des données d'exemples. Ces règles peuvent par la suite être appliquées sur de nouveaux chiffres afin de les reconnaître.

IV.3/ Les phases de l'apprentissage automatique

De manière générale, les algorithmes d'apprentissage automatique se séparent en plusieurs phases.

- Phase d'entraînement (ou d'apprentissage) : le modèle choisi est soumis à un grand nombre d'exemples significatifs. Le système cherche alors à apprendre des règles implicites en se basant sur ces données (appelées données d'entraînement). Cette phase d'entraînement précède généralement l'utilisation du modèle, bien que certains systèmes continuent d'apprendre indéfiniment s'ils disposent d'un retour sur les résultats (on appelle cela de l'apprentissage en ligne).
- Phase d'inférence : Le modèle entraîné peut être utilisé sur de nouvelles entrées.

Les entrées fournies lors de la phase d'inférence peuvent être traitées même si elles n'ont pas été vues par le modèle lors de la phase d'apprentissage. En effet, grâce à l'extraction de règles implicites, le modèle peut se généraliser à des entrées inconnues.

IV.4/ Applications et limites

En se libérant de l'explicitation de règles, l'apprentissage automatique peut s'étendre à des problèmes complexes dans de nombreux domaines. Des algorithmes d'apprentissage automatique peuvent alors être trouvés notamment dans le domaine de la santé avec l'aide au diagnostic médical et la création de vaccins (en 2019, SAM met au point un vaccin contre la grippe), dans le domaine de la robotique avec de la vision par ordinateur ou des commandes vocales, dans l'industrie avec de la détection de panne et de la gestion de systèmes. On peut également citer les algorithmes d'apprentissage automatique utilisés sur certains systèmes de voitures autonomes, les algorithmes présents dans les jeux (AlphaZero, Stockfish, DeepBlue, AlphaGo...) ou encore des systèmes utilisés en astronomie pour améliorer les observations.

Cependant, bien que l'apprentissage automatique soit flexible et permettent de résoudre des problèmes complexes, il présente quelques limites dont il faut avoir conscience :

- La précision des prédictions n'est jamais de 100 %, il existe donc une probabilité (même faible) de se tromper un jour, ce qui peut être bloquant pour certaines applications.
- De nombreux algorithmes agissent de manière opaque, il est difficile pour un utilisateur de comprendre comment le système abouti à une prédiction, même si la structure du modèle est

connue. Un enjeu majeur de l'IA est de permettre l'explicabilité des modèles.

- La correction et la détection d'erreur est difficile dans la plupart des cas, ce qui rend parfois le développement et l'entraînement du modèle difficile. Cette limite est directement liée à la question d'explicabilité.
- La plupart des algorithmes d'apprentissage automatique sont de complexité exponentielle. Ainsi, il peut être nécessaire de faire quelques approximations sur le problème, permettant de gagner en efficacité, au prix de la précision du modèle.
- Les performances du modèle sont fortement dépendantes des données d'entraînement, ce qui demande au concepteur de choisir des données suffisamment représentatives pour que le modèle soit généralisable.
- Un problème récurrent dans l'apprentissage automatique est le fléau de la dimension. Il s'agit d'un phénomène arrivant lorsque l'on cherche à analyser des données de grande dimension. La figure 2 illustre ce problème pour des données en 2 et 3 dimensions. Lorsque la dimension de l'espace augmente, son volume croît rapidement, les données qui étaient avant représentatives deviennent éparpillées et un plus grand nombre de données est alors nécessaire. Ce problème demande alors l'utilisation lorsque c'est possible de méthode de réduction de dimensionnalité (PCA, ICA, NMF...)

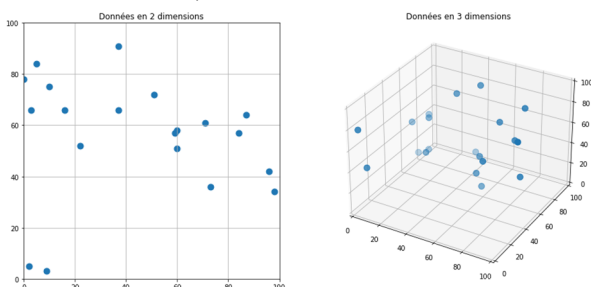


Figure 3 : Illustration du problème du fléau de la dimension
Pour avoir une bonne représentation statistique il faut un plus grand nombre de données en 3 dimensions

Il ne faut pas non plus omettre les nombreuses questions éthiques autour de l'apprentissage automatique et même plus généralement autour de l'intelligence artificielle. On peut citer pour cela l'exemple de l'algorithme COMPAS, utilisé par les autorités américaines pour évaluer le risque de récidive des criminels. En 2016, une enquête de l'organisation ProPublica a montré que l'algorithme était biaisé, jugeant les Afro-Américains avec un risque bien plus élevé. Cet exemple montre que les algorithmes d'apprentissage automatique peuvent donner des modèles biaisés et restent dépendant du concepteur et des données accessibles.

V/ Les types d'apprentissage automatique

V.1/ Apprentissage supervisé

On parle d'apprentissage supervisé lorsque l'on dispose de données d'entraînement étiquetées, c'est à dire dont on connaît la sortie voulue. En notant les N entrées x_i et les sorties cibles associées y_i , on dispose de l'ensemble de données $D = \{x_i, y_i\}_{i \in [1, N]}$. L'objectif est d'entraîner le modèle choisi pour qu'il puisse prédire correctement la sortie pour des entrées non étiquetées.

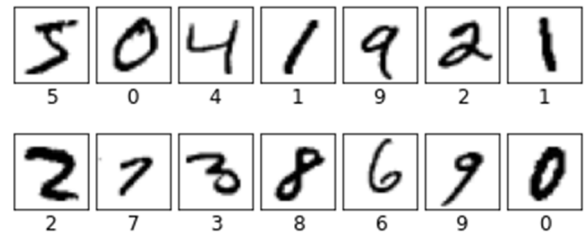


Figure 4 : Ensemble de données étiquetées
Chaque image d'entrée est associée avec la prédiction voulue

L'apprentissage supervisé est généralement utilisé pour de la régression ou de la classification :

- La régression est utilisée lorsque la sortie à prédire peut prendre des valeurs continues, il s'agit d'une variable réelle. Exemple : un algorithme prédisant la consommation électrique d'une installation ou un algorithme prédisant le cours d'actions en bourse.
- La classification est une tâche consistant à choisir une classe (valeur) parmi toutes celles possible. Exemple : Un algorithme prédisant le chiffre manuscrit sur l'image d'entrée ou un algorithme classifiant une tumeur comme « bénigne » ou « maligne »

Il est important de noter que ces deux familles, classification et régression ne sont pas exhaustives et qu'il existe des algorithmes réalisant des prédictions parmi des vecteurs de grande dimension, entre régression et classification. Les exemples les plus communs de modèles entre régression et classification se retrouvent dans le traitement automatique du langage où les mots peuvent être modélisés comme des combinaisons de lettres, donnant alors lieux à des vecteurs de très grande dimension.

Parmi les algorithmes d'apprentissage supervisé classiques, on peut citer l'algorithme des k plus proches voisins, la régression linéaire, la régression logistique et les modèles tels que les réseaux de neurones, les arbres de décision, les forêts aléatoires et les machines à support de vecteur.

V.2/ Apprentissage non-supervisé

On parle cette fois d'apprentissage non supervisé si les données ne sont pas étiquetées. On dispose donc de données d'entrée dont on ne connaît pas la sortie associée. L'ensemble de données est donc $D = \{x_i\}_{i \in [1, N]}$ et l'objectif du système est

d'identifier des caractéristiques communes aux données d'entraînement.



Figure 5 : Ensemble de données non-étiquetées
Les images d'entrées, extraites de la base de données MNIST Fashion, ne sont pas associées à une sortie cible

L'apprentissage non-supervisé est principalement composé des algorithmes de regroupement (clustering). Ces algorithmes cherchent à séparer les données d'entrées en un nombre donné de groupes. Chaque élément du groupe doit avoir des caractéristiques proches de celles des éléments du même groupe mais des caractéristiques relativement éloignées de celles des autres groupes. Ces algorithmes regroupent donc les entrées en familles afin de les étiqueter automatiquement. Par exemple un algorithme de regroupement peut être utilisé pour regrouper des patients dans le but de prévoir les réactions possibles à certains traitements.

Parmi les algorithmes d'apprentissage non-supervisé les plus courants, on peut citer l'algorithme des k-moyennes, l'analyse en composante principale, le DBSCAN, la décomposition en valeurs singulières et certains réseaux de neurones

V.3/ Apprentissage par renforcement

L'apprentissage par renforcement consiste à apprendre en interagissant avec son environnement. Un système de récompenses permet de renforcer les bons choix du modèle et pénaliser les mauvais choix. Le modèle est initialisé aléatoirement et adapte sa stratégie en fonction de son expérience durant la phase d'entraînement. Il s'agit d'un fonctionnement itératif. A chaque itération le modèle réalise une tâche et un score lui est attribué en fonction de ses performances. Les paramètres du modèle sont ensuite légèrement modifiés et la tâche est répétée. Un meilleur score oriente le choix des paramètres vers certaines valeurs, assurant que le modèle se rapproche à terme de résultats satisfaisants. Le choix des récompenses et pénalités à appliquer est donc très important car il conditionne les résultats du modèle final.

Les algorithmes génétiques, régulièrement utilisés sont un type particulier d'algorithme d'apprentissage par renforcement, se basant sur le principe de la sélection naturelle et de la génétique. Lors de la phase d'apprentissage, de nombreux individus sont créés. Tous sont basés sur le même modèle mais avec des paramètres différents et génère donc des comportements différents. Tous les individus sont soumis à la même tâche et un score leur est attribué en fonction de leurs performances. Les meilleurs individus sont sélectionnés et la génération d'après est créée à partir d'une mutation de ces derniers, c'est à dire un léger changement de leurs

paramètres. Les individus deviennent alors plus performants à mesure des générations.

Le principe de l'apprentissage par renforcement a notamment été utilisé pour entraîner l'ordinateur AlphaGo, ayant vaincu le champion du monde du jeu de Go en 2019. Le modèle avait alors la possibilité de jouer contre lui-même ou contre d'autres machines afin d'apprendre la stratégie à adopter. On peut aussi appliquer ces algorithmes pour l'entraînement d'un modèle pour la navigation de voiture autonome. Le modèle de comportement est ici complexe et peut être modélisé par le biais d'un réseau de neurone. L'utilisation d'apprentissage profond (réseaux de neurones) est aujourd'hui majoritaire pour l'apprentissage par renforcement en raison de la complexité des modèles. L'apprentissage par renforcement est également couramment utilisé pour des problèmes complexes d'optimisation sous contrainte, en attribuant des récompenses en fonction du critère d'optimisation et des pénalités selon les contraintes voulues.

L'apprentissage par renforcement est une méthode d'apprentissage automatique permettant d'entraîner un modèle en le faisant interagir avec son environnement. L'agent dispose d'un modèle initial qui doit être optimisé et est placé au sein de son environnement (généralement simulé). Il doit alors atteindre son objectif en faisant certaines actions, basée sur des observations, provenant généralement de capteurs. Un système de récompense et de pénalité permet de guider l'agent vers les bonnes actions à réaliser et d'améliorer le modèle utilisé pour la prise de décision. Un autre article de cette revue détaille les différentes méthodes d'apprentissage par renforcement

V.4/ Choix d'un modèle de machine learning

Nous avons vu qu'il existe une multitude d'algorithmes d'apprentissage automatique, il est donc question maintenant de se demander lequel utiliser. Un premier critère de choix concerne le type d'apprentissage voulu ou possible. Ainsi, on choisira de faire de l'apprentissage supervisé, de l'apprentissage non supervisé ou encore de l'apprentissage par renforcement en fonction de ce qui est possible pour la tâche voulue et en fonction des données à disposition. Cependant, il reste dans chaque catégorie une multitude d'algorithmes possibles. Il a été démontré qu'il n'existe pas d'algorithme performant mieux que les autres pour tous les problèmes.

Ce théorème appelé "No Free Lunch Theorem" déclare que tous les algorithmes d'optimisation ont des performances égales si elles sont moyennées sur tous les problèmes possibles. Autrement dit, si un algorithme A offre de meilleurs résultats qu'un algorithme B sur une tâche, alors il existe obligatoirement au moins une tâche pour laquelle les résultats de A seront moins bons que ceux donnés par B. Cela empêche donc de choisir systématiquement le même algorithme pour toutes les tâches. En effet, type d'apprentissage et le modèle sont

choisis en fonction des hypothèses réalisées pour un problème donné. Ces hypothèses ne seront pas validées pour toutes les tâches possibles et le modèle sera alors plus ou moins correct en fonction du problème traité. Une certaine expertise est donc nécessaire pour choisir un modèle à partir de la tâche considérée, des données disponibles et des contraintes qui s'appliquent sur le système (mémoire, puissance de calcul...). De nouvelles méthodes, telles que AutoML, permettent d'aider le concepteur dans le choix d'un modèle adapté à son problème.

Il est important de réévaluer les hypothèses pour chaque problème considéré et d'en déduire quel algorithme sera plus performant dans le cas étudié. Par exemple, pour un problème donné d'apprentissage supervisé, on peut faire l'hypothèse que la sortie prend des valeurs continues et choisir un algorithme de régression ou bien faire l'hypothèse que les valeurs sont discrètes et suffisamment peu nombreuses pour utiliser un algorithme de classification. Dans ce second cas, on peut également supposer que le modèle de classification est suffisamment linéaire pour utiliser l'algorithme des k plus proches voisins ou au contraire opter pour un modèle plus complexe basé sur un réseau de neurones. En somme, le concepteur d'un système d'apprentissage automatique doit analyser les données à disposition et l'objectif voulu pour formuler des hypothèses cohérentes et qui permettront de choisir un algorithme.

Dans les parties suivantes, nous présenterons des algorithmes classiques d'apprentissage supervisé et non-supervisé.

VI/ Apprentissage supervisé

VI.1/ Régression linéaire

VI.1.1/ Principe de fonctionnement

L'objectif de la régression linéaire est d'exprimer une variable de sortie y en fonction de la variable d'entrée x de manière linéaire, c'est à dire $y = ax + b$. Ce modèle a donc deux paramètres A et B , dont il faut trouver les valeurs optimales durant la phase d'apprentissage. Plusieurs techniques existent pour estimer ces paramètres, les plus répandues étant la méthode des moindres carrés, la méthode des déviations et la méthode du maximum de vraisemblance.

La régression linéaire est un algorithme d'apprentissage supervisé, on dispose alors de N couples entrée-sortie constituant l'ensemble de données $D = \{x_i, y_i\}_{i \in [1, N]}$. Ce sont ces données connues qui vont permettre d'estimer les paramètres du modèle. Nous allons prendre ici l'exemple de la méthode des moindres carrés, plus répandue que les autres. Le principe des algorithmes de régression linéaire reste similaire pour les autres méthodes.

Dans le cas de la méthode des moindres carrés, on cherche les paramètres A et B permettant de minimiser une fonction coût $C = \sum_{i=1}^N (y_i - (ax_i + b))^2$. Cette fonction correspond à la somme des écarts au carré entre

les prédictions et les valeurs attendues. Ces écarts à minimiser sont appelés résidus.

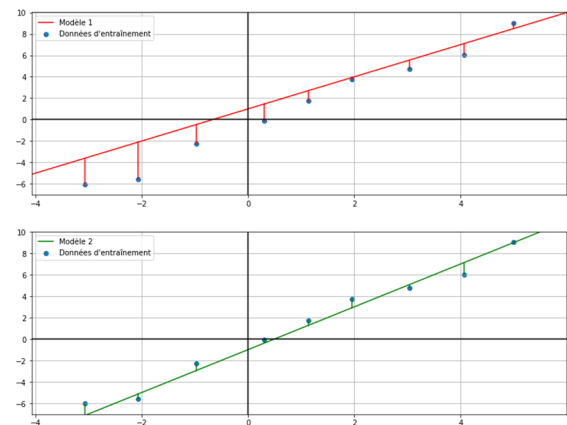


Figure 6 : Deux modèles de régression linéaire
Le premier modèle présente des écarts importants entre les valeurs prédites et attendues tandis que le second minimise les carrés de ces écarts

On peut calculer le coefficient de régression linéaire pour estimer si deux variables peuvent être liées linéairement. Ce coefficient est calculé à partir des écarts types σ_x et σ_y des variables et à partir de la covariance entre les variables d'entrée et de sortie : $R = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$. Si le coefficient est proche en valeur absolue de 1, cela signifie que les variables peuvent être liées linéairement. Il s'agit donc d'un moyen pour vérifier que la régression linéaire est un choix cohérent de modèle.

VI.1.2/ Applications

Les applications de la régression linéaire sont nombreuses. On peut par exemple imaginer une mesure du courant traversant une résistance pour différentes valeurs de tension appliquée à ses bornes. Le mesure de courant peut être bruitée, empêchant de déterminer directement la valeur de la résistance traversée. On peut utiliser une régression linéaire pour estimer la valeur de cette résistance en minimisant l'écart entre le courant mesuré et la valeur estimée par le modèle $U=RI$. Cette même méthode s'applique pour estimer des paramètres dans de nombreux domaines.

En pratique, il existe de nombreux logiciels permettant de réaliser une régression linéaire. La plupart des tableurs disposent cette fonctionnalité et certains langages de programmation proposent des bibliothèques permettant de faire cette régression (scikit learn pour Python, opérateur '\ pour Matlab...)

VI.1.3/ Limites

Bien qu'il ait l'avantage d'être facilement et rapidement implémentable, le modèle de régression linéaire présente de nombreuses limites. Tout d'abord, comme nous l'avons vu, cette méthode est limitée à l'estimation de modèles linéaires, ce qui restreint énormément les applications possibles. Le principe de la régression linéaire peut cependant s'étendre à des régressions polynomiales. Ces dernières sont plus complexes à mettre en œuvre car elles demandent plus de paramètres (et de choisir le nombre de paramètres) et font face aux problèmes de sur-apprentissage et de sous-

apprentissage, qui seront abordés plus loin dans cet article.

D'autre part, l'estimation obtenue est sensible au bruit de mesure introduit dans les données d'apprentissage. Ainsi, une donnée aberrante (très éloignée du modèle à estimer) influera sur les estimations des paramètres. Cette influence est plus ou moins importante selon la méthode de régression linéaire utilisée. En effet, la méthode des moindres carrés considère le carré des résidus et une valeur aberrante décalera donc plus le modèle que dans le cas de la méthode des moindres déviations (qui utilise les valeurs absolues des résidus). Certaines méthodes permettent d'intégrer des connaissances *a priori*, limitant davantage l'effet de valeurs aberrantes.

Afin de réduire l'effet du bruit sur l'estimation, il est important d'avoir un nombre suffisamment important de données, ce qui contraint également le protocole expérimental.

VI.2/ Algorithme des K plus proches voisins

VI.2.1/ Principe

L'algorithme des K plus proches voisins, comme la régression linéaire, est un algorithme d'apprentissage supervisé, on dispose donc de l'ensemble de données $D = \{x_i, y_i\}_{i \in [1, N]}$. En revanche, il s'agit ici d'un modèle non paramétrique, c'est à dire que le modèle ne dispose pas de paramètres dont la valeur serait à optimiser (nous verrons plus tard que K n'est pas un paramètre mais un hyper-paramètre). Il s'agit également d'un algorithme de classification, les sorties y_i correspondent alors aux classes possible (et non pas des valeurs continues comme pour la régression).

L'entrée x_i est un vecteur de dimension p , contenant les différentes variables associées à une entrée. Pour chaque nouvelle entrée u , on mesure la distance entre u et les entrées x_i pour tout $i \in [1, N]$. On sélectionne ensuite les K éléments les plus proches (distance plus faible) de l'entrée. La classe prédite par l'algorithme correspond alors à la classe la plus fréquente dans les K plus proches voisins sélectionnés.

On peut utiliser différentes mesures de distance (euclidienne, Manhattan...) pour cet algorithme en fonction du problème étudié.

VI.2.1/ Exemple

Un exemple célèbre et couramment utilisé pour les K plus proches voisins se base sur la base de données Iris de Fisher. En 1936, le botaniste Edgar Anderson a mesuré 150 spécimens d'Iris de trois espèces différentes, les Iris Setosa, les Iris Versicolor et les Iris Virginica. Pour chaque spécimen, il a mesuré la longueur et la largeur des pétales et des sépales. Dans le cadre de cet exemple, nous allons nous intéresser uniquement à la longueur et largeur des pétales. Les entrées x_i sont donc vecteurs contenant la longueur des pétales et la largeur des pétales. Les sorties y_i sont un entier 0, 1 ou 2, correspondant à la classe du spécimen (setosa, versicolor ou virginica).

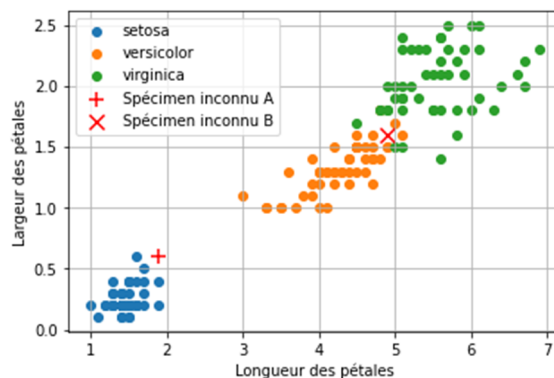


Figure 7 : Bases de données Iris de Fisher
Les 150 spécimens sont répartis en 3 espèces et positionnés en fonction de la longueur et largeur des pétales

On peut observer que les spécimens sont regroupés en fonction de leur espèce. En effet, des spécimens d'une même espèce ont des caractéristiques similaires. Il semble alors cohérent de prédire l'appartenance d'un spécimen à une espèce selon les autres spécimens proches (au sens de la distance euclidienne). Considérant pour cet exemple deux spécimens inconnus A et B et cherchons à prédire l'espèce à laquelle ils appartiennent.

- Le premier spécimen, appelé A est représenté par un '+' sur le graphique. Il s'agit d'un Iris avec des pétales de 1.9 cm de longueur et 0.6 cm de largeur. Cet Iris a des caractéristiques plus proches de celle des spécimens d'Iris setosa que de celle des autres spécimens. On peut alors classer cet Iris dans l'espèce setosa.
- Le second spécimen, appelé B, a des pétales de longueur 4.9 cm et de largeur 1.6 cm et est représenté par un 'x' sur le graphique. On peut alors regarder les 5 spécimens d'Iris connus les plus proches du spécimen B. On peut constater que parmi ceux-là il y a 1 Iris virginica et 4 Iris versicolor. On classe alors le spécimen B dans l'espèce versicolor.

Nous avons donc pu classer deux spécimens inconnus en utilisant la méthode des K plus proches voisins.

Notons que nous avons utilisés ici l'algorithme dans le cadre de classification mais celui-ci peut être étendu à de la régression. Dans ce cas, les éléments ne sont pas associés à une classe mais à une valeur. La sortie de l'algorithme correspond alors à la valeur moyenne des K plus proche voisins.

VI.2.2/ Applications et limites

L'algorithme des K plus proches voisins est fréquemment utilisé en classification comme en régression. On peut alors le retrouver dans le cadre de la reconnaissance de forme avec des entrées contenant la circonférence, l'aire ou encore la signature du contour de la forme et des sorties correspondant aux différentes formes possibles. Cet algorithme a l'avantage d'être relativement robuste si suffisamment d'exemple d'entraînement sont fournis.

Cet algorithme est également facilement implémentable dans différents langages de programmation.

Le choix de la valeur de K dans l'algorithme des K plus proches voisins a une influence forte sur la prédiction faite par l'algorithme.

- K trop petit : des éléments sortant de l'ordinaire influenceront plus facilement la prédiction. La généralisation du modèle pour de nouveaux éléments sera donc moins bonne, il s'agit du problème de sur-apprentissage (overfitting). Par exemple, si l'on choisit $K = 1$, un Iris versicolor ayant des pétales anormalement petits sera pris en compte par le modèle qui risque de mal classer un Iris aux dimensions trop proche de cette valeur aberrante.
- K trop grand : Le modèle prendra en compte des données trop éloignées et la classe majoritaire sera prédite trop souvent. Il s'agit cette fois du problème de sous apprentissage, le modèle n'utilise que trop peu les données d'entraînement. Par exemple, si l'on choisit $K=N$, tous les Iris seront pris en compte et la classe prédite sera la même peu importe la taille des pétales.

Ainsi, le choix de la valeur de K n'est pas aisé, il ne suffit pas de la prendre grande ou de la prendre petite. Différentes méthodes existent pour avoir une idée de la valeur de K possible. Il est également important de tester l'algorithme sur un ensemble de données connues afin de vérifier le fonctionnement de ce dernier. On peut alors séparer l'ensemble de données connues D en deux ensemble, l'ensemble d'entraînement D_e et l'ensemble de test D_t . Le premier permettra de réaliser la prédiction et le second permettra de vérifier les performances de l'algorithme et d'éviter les phénomènes de sur-apprentissage et de sous-apprentissage.

Un des principaux problèmes rencontrés lors de l'implémentation de l'algorithme des K plus proches voisins est le fléau de la dimension. En effet, pour que l'algorithme fonctionne de manière optimale, il faut un nombre suffisant de données d'entraînement afin que les points étudiés soient toujours proches d'exemples connus. Lors de l'étude de problèmes de plus grande dimension, il est donc primordial d'avoir un grand nombre de données d'entraînement. Pour cette raison, l'algorithme des K plus proches voisins devient rapidement inutilisable (au-delà de 4 ou 5 dimensions, le nombre de données nécessaires devient trop important)

VI.3/ Autres algorithmes d'apprentissage supervisé

Il existe un grand nombre d'algorithmes d'apprentissage supervisé. Hormis les algorithmes de régression linéaire ou polynomiale et l'algorithme des k plus proches voisins, déjà mentionnés précédemment, on peut citer les suivants.

- Régression logistique : On cherche ici à prédire le comportement d'une variable de sortie Y en fonction des entrées X , comme en régression

linéaire ou polynomiale. Cependant, dans le cadre de la régression logistique ordinaire, Y n'est pas un réel mais prend les valeurs 0 ou 1 (il s'agit d'une variable binaire). Elle permet d'évaluer l'effet de différentes variables explicatives sur une variable d'intérêt. Cette méthode d'apprentissage automatique est très utilisée dans le domaine du marketing pour évaluer la vente ou non de produits à la suite d'une décision ou dans le domaine médical pour évaluer la guérison ou non d'un patient.

- Arbre de classification : Les arbres de classification (ou arbre de décision) et de régression sont des méthodes permettant de créer un modèle de décision. Cette structure est constituée de nœuds racines, constituant les entrées de l'arbre, de nœuds internes, réalisant des opérations intermédiaires et de feuilles (ou nœuds terminaux), représentant la valeur des variables de sorties. Cette structure hiérarchique se base sur un enchaînement de décisions créant le passage d'un nœud à l'autre, jusqu'aux feuilles. L'arbre est généralement construit récursivement en cherchant pour chaque nœud la décision permettant de réaliser le meilleur partage de l'ensemble de données. Après une décision, on dispose alors de plusieurs sous-ensembles de données sur lesquels on applique le même processus. Les arbres de décision ont l'avantage d'avoir un processus de décision clair, garantissant une bonne explicabilité de la solution. Ils ont longtemps été utilisés avant le développement des réseaux de neurones artificiels. Il est possible de décomposer les problèmes plus complexes en utilisant plusieurs arbres de classification, on parle alors de forêt aléatoire.
- Réseaux de neurones artificiels : Les réseaux de neurones sont des modèles complexes pouvant être utilisés en apprentissage supervisé pour de la régression et de la classification. Ces modèles sont expliqués en détail dans un autre article de cette revue. Ils sont de plus en plus utilisés dans de très nombreux domaines car ils permettent de modéliser des phénomènes complexes. On les retrouve par exemple en traitement d'image pour de la reconnaissance de motif, dans le traitement de la parole et pour l'approximation de fonctions complexes longues à calculer.
- - Machine à support de vecteur (SVM) : Les SVM constituent une généralisation des classificateurs et estimateurs linéaires (tels que les modèles de régression linéaire) pour des données de plus grande dimension. Le modèle prend alors en entrée un vecteur de paramètre X et lui associe une valeur de sortie Y grâce à une fonction $h(X)$. Dans le cas de classification, on introduit des hyperplans séparateurs servant de critère de décision. L'espace de la valeur de sortie Y est séparé en différentes zones correspondant chacune à des prédictions. Par

exemple, on prédira une classe si la valeur de Y est positive et une autre si elle est négative.

VII/ Apprentissage non-supervisé

VII.1/ Algorithme des k-moyennes

VII.1.1/ Principe

Le premier exemple d'apprentissage non-supervisé que nous allons voir est l'algorithme des k-moyennes. Nous disposons cette fois de données $D = \{x_i\}_{i \in [1, N]}$ non-labelisées que nous allons chercher à classer automatiquement. Les k-moyennes est un algorithme de regroupement non supervisé qui cherche à créer k groupes d'éléments de caractéristiques proches. Afin de sélectionner les éléments proches pour les rassembler en un groupe, l'algorithme cherche à minimiser la variance intra-classe, c'est à dire la distance euclidienne entre les éléments d'une classe et son barycentre. En notant m_1, m_2, \dots, m_k les barycentres des classes C_1, C_2, \dots, C_k , on cherche alors :

$$m_1, m_2, \dots, m_k \text{ tels que } \sum_{i=1}^k \sum_{j \in C_i} \|x_j - m_i\|^2 \text{ est minimal}$$

Le déroulement de l'algorithme est le suivant :

- On choisit k positions initiales correspondant au centre des classes C_1, \dots, C_k , souvent aléatoirement.
- On associe chaque élément de la base de données D_e à la classe dont il est le plus proche. Les données sont alors réparties en k groupes (certains groupes peuvent être vides)
- Le barycentre des éléments de chaque groupe est recalculé et devient le nouveau centre des classes C_1, \dots, C_k
- On répète les étapes précédentes jusqu'à ce que les centres ne bougent plus entre deux itérations, il y a convergence.

VII.1.2/ Exemple

On peut utiliser à nouveau la base de données Iris de Fisher, en se plaçant cette fois dans le cas où les labels ne sont pas disponibles. On a alors les mesures de tous les spécimens mais nous ne connaissons pas leur appartenance aux trois espèces différentes. Il est alors possible de mettre en œuvre l'algorithme des K-moyennes pour séparer les spécimens dans K classes, nous permettant alors de retrouver approximativement les appartenances aux espèces. Dans notre cas, nous allons utiliser $k = 3$, supposant que le nombre d'espèce à extraire est connu. Il est possible d'appliquer le même algorithme pour un nombre k différent, ce qui donnera alors un nombre différent de classe.

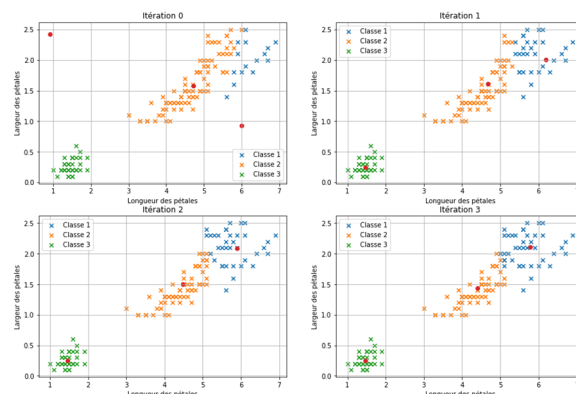


Figure 8: Résultat de l'algorithme des K moyennes sur la base de données Iris de Fisher

On utilise ici $K=3$ et l'on observe que les centres ne changent pas entre les deux dernières itérations

On peut observer la convergence rapide de l'algorithme, les centres des classes ne bougent plus au bout de 3 itérations. On obtient un résultat semblable à celui attendu (présenté en partie Apprentissage Supervisé - K plus proches voisins). Cependant, on peut observer que les spécimens sortant de l'ordinaire (par exemple avec des pétales longs pour son espèce) se retrouvent dans la mauvaise classe car ils sont trop éloignés de la moyenne de leur espèce.

VII.1.3/ Applications et limites

Comme nous avons pu le souligner en étudiant un exemple d'application, cet algorithme présente plusieurs limites dont il faut avoir conscience. Tout d'abord, le choix du nombre de classe K à utiliser est particulièrement important et ce choix peut souvent être difficile. Dans le cas où l'on choisit une valeur de k trop petite, pas assez de classes seront créées et des éléments seront regroupés à tort dans une même classe. A l'inverse, un choix de k trop grand créera trop de classe, séparant des éléments qui aurait dû être regroupés. Le choix de cette valeur dépend alors fortement de l'application et du cahier des charges associées. Il existe cependant des méthodes comme la méthode d'Elbow, permettant de choisir une valeur optimale de k en comparant les valeurs obtenues pour plusieurs k.

Nous avons également vu que cet algorithme ne considère pas les éléments éloignés de la moyenne de leur classe, ce qui classe donc un certain nombre d'éléments dans la mauvaise classe. Ce phénomène peut être réduit en prenant en compte plus de caractéristiques pour les éléments. Les éléments anormaux auront également tendance à faire dévier le barycentre, faisant potentiellement perdre en précision.

De plus, le choix initial des centres n'est pas toujours simple. Bien qu'une initialisation aléatoire puisse fonctionner, ce n'est pas toujours le cas et l'algorithme peut alors converger vers des groupes différents de ceux attendus.

Enfin, une des limites les plus restrictive de l'algorithme des K moyennes concerne la taille et la forme des clusters à identifier. En effet, cet algorithme est peu adapté pour des groupes de taille différentes et ne fonctionne pas pour des groupes de formes différentes.

L'algorithme des k-moyennes est utilisé dans de nombreuses applications, comme le groupement de la clientèle permettant d'adapter les offres en se basant sur différents critères, l'extraction de couleur (par exemple sur l'histogramme d'une image) ou encore l'apprentissage semi-supervisé. En effet, il est possible d'imaginer des algorithmes à la frontière entre apprentissage supervisé et non supervisé. On dispose alors d'un ensemble de données étiquetées (en petite quantité) et de données non-étiquetées (en plus grande quantité). On peut alors créer des groupes en utilisant les techniques d'apprentissage non-supervisé puis étiqueter ces données en se basant sur la partie étiquetée de la base de données. Ce type d'apprentissage permet de limiter le travail humain nécessaire pour créer les bases de données permettant l'apprentissage supervisé et permet donc de traiter un plus grand nombre de données, améliorant de ce fait les performances des modèles.

VII.2/ Autres algorithmes d'apprentissage non-supervisé

Il existe plusieurs autres algorithmes d'apprentissage non-supervisé, utilisés dans différentes circonstances. Il est important de noter que ces algorithmes ne sont pas tous adaptés aux mêmes applications et certains seront plus performants que d'autres pour l'application voulue. Parmi les algorithmes couramment utilisés, on peut citer les suivants :

- **Modèle de mélange de gaussien (GMM) :** Il s'agit d'un modèle statistique permettant d'estimer la distribution de variables aléatoires en la modélisant comme une somme de distributions gaussiennes de paramètres différents. L'étape d'entraînement cherche alors à déterminer la moyenne et la variance des distributions gaussiennes en utilisant un critère de maximum de vraisemblance. Ces modèles sont fréquemment utilisés dans des problèmes de classification comme la reconnaissance de l'écriture manuscrite et la segmentation d'images par leur histogramme. Le modèle de mélange de gaussien permet, contrairement à l'algorithme des K moyennes, de réaliser des groupes de tailles différentes.
- **Analyse en composante principale (PCA) :** Cet algorithme permet de réduire les dimensions d'un ensemble de données, permettant alors un traitement plus simple et plus rapide. Cette méthode crée des variables décorrélatées en réalisant un changement de base. Les vecteurs de cette nouvelle base, appelés composantes principales, sont ensuite sélectionnés pour ne garder que celles portant le plus d'informations. Il est alors possible de minimiser la perte d'information tout en réduisant efficacement la dimensionnalité de l'ensemble. Cet algorithme est largement utilisé en neuroscience ou en analyse des mouvements où de nombreux signaux de grande dimension (activation de neurones ou mouvements des membres) sont utilisés pour traduire des informations plus simples (réponse à un stimulus ou déplacement dans une

direction). D'autre algorithme de réduction de dimensionnalité existe, comme la décomposition en valeur singulière ou l'analyse en composante indépendante (ICP).

VIII/ Choix et influence des hyper-paramètres

Nous allons nous intéresser dans cette partie à la notion de paramètres et d'hyperparamètres, fondamentale en apprentissage automatique. La différence entre ces deux termes est fondamentale et permet de comprendre comment les choix du concepteur influent sur le comportement final du système.

- Les hyper-paramètres sont des réglages du modèle et de l'algorithme d'apprentissage, ils décrivent le déroulement de l'algorithme et sont généralement choisis par le concepteur. Ils conditionnent les limites de fonctionnement et les performances du modèle et influent sur l'efficacité de l'apprentissage.

Exemple : Valeur de K dans l'algorithme des K plus proches voisins, nombre de couches d'un réseau de neurones, degré du modèle d'une régression polynomiale.

- Les paramètres sont des valeurs variables dans un modèle, ils définissent son comportement. Les valeurs des paramètres changent au cours de l'apprentissage afin de donner au modèle le comportement voulu, permettant de répondre à la tâche voulue.

Exemple : Valeurs de A et B dans le modèle de régression linéaire, valeurs des poids et des biais d'un réseau de neurones.

Certains algorithmes d'apprentissage automatique demandent de choisir de nombreux hyper-paramètres, il est donc très important de comprendre l'influence de chacun afin d'adapter les choix à l'application voulue. Il est par exemple possible de se demander quel degré choisir pour le modèle d'une régression polynomiale. Le modèle donnera des résultats généralement différents si l'on utilise un modèle linéaire ou un modèle d'ordre 3. Nous devons ici choisir un hyper-paramètre, ce choix est particulier à la tâche visée et un choix sera rarement bon pour plusieurs applications différentes.

Un mauvais choix d'hyper-paramètre peut détériorer l'étape d'apprentissage, conduisant à des paramètres qui ne sont pas optimaux et donc à un modèle imparfait. Il est fréquent de rencontrer des problèmes de sous-apprentissage et de sur-apprentissage.

- **Sous-apprentissage :** Il s'agit d'un cas où le modèle n'apprend pas suffisamment et se retrouve alors incapable de généraliser efficacement sa prédiction à des entrées inconnues. On peut dans ce cas voir que les performances sur les données d'entraînement et de validation sont médiocres. Ce phénomène arrive généralement lorsque la phase d'apprentissage est trop courte ou lorsque le modèle dispose de trop peu de paramètres, ces deux raisons étant généralement liées aux choix d'hyper-paramètres

- **Sur-apprentissage** : Dans ce cas, le modèle n'extrait pas les règles implicites mais apprend "par cœur" les données d'entraînement. Il est à nouveau incapable de généraliser efficacement ses prédictions à des nouvelles entrées. Ici on obtient une très bonne précision pour les données d'entraînement et de validation mais les tests donnent des résultats beaucoup moins bons. Ce phénomène arrive généralement quand un modèle contient plus de paramètres que nécessaire à cause d'un mauvais choix d'hyper-paramètres. En ensemble de données d'entraînement trop peu représentatives (ou trop peu nombreuses) conduit également à une forme de sur-apprentissage, le modèle ne se généralisant pas correctement pour des exemples inconnus.

L'article traitant de l'apprentissage profond présente un exemple plus détaillé de sur-apprentissage et de sous-apprentissage induits par un mauvais choix des hyper-paramètres du réseau de neurones.

IX/ Conclusion

Nous avons essayé de donner dans cet article une introduction à la fois large et utile à l'intelligence artificielle. Le domaine est vaste et requiert certains prérequis. Nous allons résumer ici quelques points clés à retenir :

- L'IA tente de reproduire l'intelligence pour aider l'humain dans de multiples domaines
- Apprentissage supervisé : le jeu de données d'entraînement est composé d'éléments étiquetés manuellement par des humains et la machine extrait des règles implicites lui permettant d'étiqueter de nouveaux exemples.
- Apprentissage non-supervisé : le jeu de données d'entraînement n'est pas étiqueté l'algorithme cherche alors de la similarité dans les données pour pouvoir les classer.
- Apprentissage par renforcement : un agent apprend en interagissant avec son environnement grâce à un système de récompenses favorisant les bonnes actions
- L'IA est un paradigme puissant pour modéliser des systèmes complexes, mais pas magique. Voici un rappel de quelques limitations auxquelles ces systèmes font face :
 - Le « No Free Lunch Theorem » indique qu'il faut choisir un algorithme spécifique pour résoudre la tâche voulue, il n'existe pas de méthode universelle
 - Les résultats sont toujours probabilistes et il existe toujours une possibilité d'erreur
 - Les phénomènes de sous-apprentissage et de sur-apprentissage demande de prendre des précautions lors du choix du modèle et des données d'entraînement
 - Le jeu de donnée d'entraînement doit être représentatif et non-biaisé

Les nombreuses techniques d'apprentissage automatique sont présentes dans un grand nombre de domaines d'application. Il est important de comprendre le principe de fonctionnement des techniques utilisées afin de s'assurer de la pertinence d'une méthode pour un problème donné.

X/ Bibliographie

- [1] Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016.
- [2] Artificial Intelligence: A modern approach, 2nd Ed., Dr. Afşar Saranlı and Stuart Russel and Peter Norvig, 2010.
- [3] Pattern Recognition and Machine Learning, Christopher M. Bishop, 2006.
- [4] Intelligence artificielle : triomphes et déceptions, Melanie Mitchell, postface de Douglas Hofstadter, traduit de l'anglais (Etats-Unis) par Christian Jeanmougin, Dunod, 2021

INTRODUCTION À L'APPRENTISSAGE PROFOND

STEEVEN JANNY¹, LUDOVIC DE MATTEIS², WENQI SHU-QUARTIER³

- ¹ : ancien élève du département Nikola Tesla de l'ENS Paris-Saclay, Doctorant au laboratoire LAGEPP sur l'établissement de modèles profonds pour la simulation et la prédiction de phénomènes physiques steeven.janny@insa-lyon.fr
- ² : élève du département Nikola Tesla de l'ENS Paris-Saclay, auteur de l'article "Personalized Human-Swarm Interaction Through Hand Motion" issu de son stage à l'EPFL sur l'apprentissage des mouvements intuitifs de l'utilisateur pour le contrôle d'une flotte de drone. ludovic.de_matteis@ens-paris-saclay.fr
- ³ : ancienne élève du département Nikola Tesla de l'ENS Paris-Saclay, Doctorante au laboratoire Neurospin du CEA sur l'application de l'apprentissage par ordinateur à la comparaison de la maladie d'Alzheimer avec son analogue chez le chimpanzé. wenqi.shu-quartier-dit-maire@cea.fr

Résumé : Ce second article présente l'outil à la source des progrès déterminants de l'IA au 21ème siècle : les réseaux de neurones et leur entraînement dit apprentissage profond. Après une présentation détaillée des réseaux de neurones, l'article développe les méthodes d'apprentissage en soulignant l'importance des données. Il présente ensuite quelques évolutions des réseaux de neurones : convolutionnels (CNN), récurrents (RNN), auto-encodeurs (AE), adversariaux génératif (GAN) et transformers pour finir sur une implémentation pratique d'un entraînement de réseau de neurones pour la classification de chiffres manuscrits.

Mots clés : IA, Intelligence artificielle, réseaux de neurones

I/ Introduction

Depuis les bureaux spacieux des grands groupes industriels, jusqu'aux locaux des incubateurs à start-ups accompagnant de jeunes entrepreneurs ambitieux, tous annoncent la venue d'une quatrième révolution industrielle sonnante le début de l'ère de l'intelligence artificielle et du Big Data. Comme ses grandes sœurs, cette révolution s'appuie sur une nouvelle sémantique dont certains éléments de langage vous sont sûrement déjà familiers : *Smart Tech*, *Cloud computing*, *IA*, etc... La quatrième révolution industrielle serait donc celle d'une nouvelle génération de machines et d'une automatisation plus poussée et plus généralisée, une transition permettant d'améliorer l'efficacité et la productivité. Pour cela, le monde moderne dispose de deux nouveaux outils redoutables : l'expansion fulgurante d'Internet, et les récentes avancées en matière d'intelligence artificielle.

La dernière décennie donne en partie raison à ces nouveaux prophètes grâce à un alignement des astres technologiques particulièrement opportun : (1) une explosion de la puissance de calcul des ordinateurs, (2) une mise à disposition de quantités massives de données numériques (le fameux *Big Data*) et (3) la remise au goût du jour d'une technologie vieille de 60 ans dont le potentiel est enfin pleinement libéré grâce aux deux derniers points, nous voulons bien sûr parler des réseaux de neurones.

L'apprentissage profond (c'est le nom de la discipline) est un domaine aujourd'hui soutenu par une communauté très active d'industriels et de chercheurs, et compte plusieurs succès à son palmarès. Ses premières grandes avancées sont concentrées sur la vision par ordinateur : reconnaissance d'objets, localisation de défauts dans les chaînes de production, détection de maladies grâce à l'imagerie médicale, etc... Récemment, le champ d'application s'est élargi : traitement du langage naturel (GPT-3 développé par OpenAI, chatbots), robotique (apprentissage par renforcement), reconnaissance vocale (Alexa, Siri, Cortana...), et même physique, avec notamment la réussite spectaculaire d'un modèle d'apprentissage profond pour la stabilisation du plasma dans le réacteur à fusion ITER, là où les techniques de contrôles traditionnelles avaient échoué.

Malheureusement, tout succès médiatique vient avec son lot de contre-vérités et d'exagérations. Nombre d'articles relatent chaque jour les dernières prouesses de l'IA. Souvent dithyrambiques, parfois même alarmistes, il est difficile de démêler la science du travail d'écriture. Après un premier article présentant différents domaines et méthodes de l'IA, cet article a pour ambition de poser les bases du fonctionnement et de la mise en œuvre des réseaux de neurones et de mettre en avant les principaux défis que ceux-ci doivent encore relever.

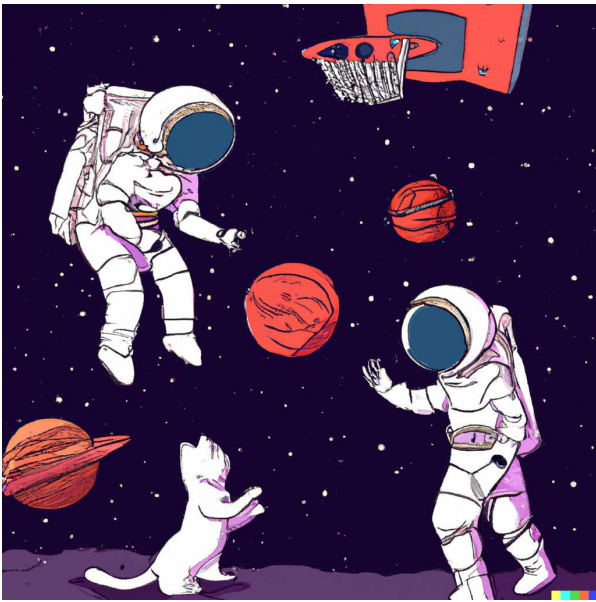


Figure 1 : Image générée par un réseau de neurones à partir de la phrase "Un astronaute jouant au basketball dans l'espace avec des chats, dessiné comme dans un livre pour enfants" (OpenAI Dall-E 2).

II/ Les réseaux de neurones comme modèles paramétriques

II.1/ Les données, ou l'or numérique

Votre âge, votre sexe, le contenu de votre dernier panier de courses, vos performances au dernier jeu mobile à la mode, toutes ces données ont un prix que bon nombre d'entreprises sont prêtes à acheter. Cette ruée vers l'or numérique est en partie motivée par le principe d'application du *deep learning*.

Mettons-nous quelques instants dans la peau d'un ingénieur en apprentissage profond. Nous nous intéressons donc à la recherche d'une fonction $y = f(x)$ dont nous ignorons complètement la formulation. Il peut s'agir d'une fonction arbitrairement complexe, comme la relation entre les images perçues par une caméra embarquée sur un véhicule et la trajectoire optimale que le véhicule devrait suivre pour atteindre un certain emplacement. À défaut de pouvoir écrire mathématiquement cette fonction, nous disposons de N mesures $(x_i, y_i)_{i=1...N}$ de celle-ci, réunies dans un grand jeu de données.

Le problème est alors le suivant : *comment construire une approximation de la fonction f à partir de ces seules mesures ?* Avant même de s'intéresser aux méthodes envisageables, on peut intuitivement se convaincre qu'une fonction complexe nécessitera plus de points de mesure. Par exemple, deux mesures suffisent pour approcher une fonction linéaire, mais au moins trois mesures sont nécessaires pour approcher une fonction quadratique. Un des principaux enjeux du Big Data est ainsi de collecter une large quantité de données afin de pouvoir approcher des fonctions de plus en plus complexes.

II.2/ Un détour par la régression linéaire

Pour approcher une fonction dont on ne connaît que quelques réalisations, la méthode classique consiste à choisir un modèle paramétrique, c'est-à-dire une fonction dont le comportement sera contrôlé par un certain nombre de paramètres qu'il nous faudra régler (on parle d'identification des paramètres).

La régression linéaire est l'exemple le plus trivial de cette technique : il s'agit de chercher les valeurs de a et b de sorte que l'équation $y = ax + b$ représente au mieux les points de mesure du jeu de données. L'identification de a et b se fait généralement en résolvant un problème de minimisation d'une mesure de l'erreur, comme la distance quadratique :

$$(a, b) = \arg \min_{(a,b)} J = \sum_{i=0}^N (y_i - (ax_i + b))^2$$

Cette solution s'adapte sans problème aux plus grandes dimensions en ré-écrivant le modèle linéaire sous forme matricielle $y = Wx + b$. La régression linéaire est intéressante car sa forme, très simple, permet d'obtenir rapidement la solution optimale au problème, c'est-à-dire l'obtention des valeurs de W et b donnant la plus petite erreur J possible. L'inconvénient, c'est que bien peu de phénomènes sont linéaires : il y a peu de chance pour qu'un modèle linéaire parvienne à prédire l'évolution du prix des actions en bourse.

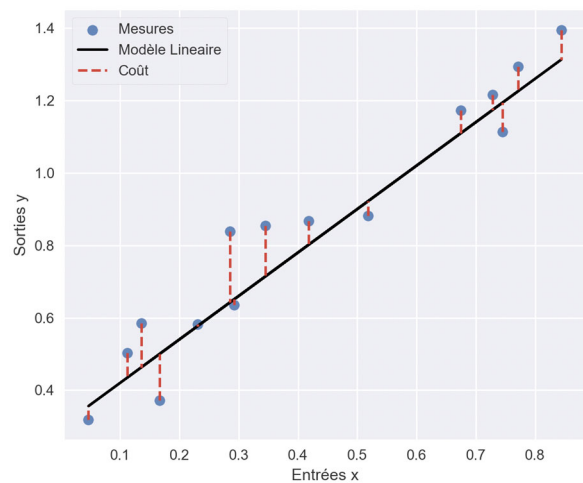


Figure 2 : La régression linéaire consiste à trouver la droite qui minimise la distance aux points de mesures, représentée en rouge sur la figure

II.3/ Un modèle plus puissant

Tout comme la régression linéaire, le réseau de neurones est un autre modèle paramétrique qui, sous sa forme la plus simple, consiste en un enchaînement de modèles linéaires séparés par de petites fonctions non-linéaires :

$$y = W_1 \sigma(W_2 \dots \sigma(W_n x + b_n) \dots + b_2) + b_1$$

Cela peut paraître surprenant, mais cette nouvelle structure, appelée perceptron multi-couche (*multi-layer perceptron* ou *MLP*) est *in fine* incroyablement plus puissante que la régression linéaire. Mathématiquement,

il s'agit d'un approximateur universel, autrement dit, un MLP peut approcher n'importe quelle fonction avec une précision arbitraire, si tant est qu'il contienne suffisamment de couches linéaires enchaînées.

Il y a bien sûr une contrepartie : dans la quasi totalité des cas, il est impossible de trouver les valeurs optimales de $W_1, \dots, W_n, b_1, \dots, b_n$ et le mieux que l'on puisse garantir est de trouver un minimum local de J . Cet aspect a longtemps fait fuir les chercheurs en intelligence artificielle, jusqu'à ce que l'on s'aperçoive que ce minimum local était déjà incroyablement performant pour de nombreuses tâches.

II.4/ Petit passage par la biologie

Si les réseaux de neurones s'appellent ainsi, c'est effectivement parce qu'ils sont en quelque sorte reliés à ce que l'on connaît du fonctionnement d'un neurone biologique. Le perceptron (Franck Rosenblatt, 1958) est inspiré du fonctionnement du cerveau : un neurone reçoit des impulsions électriques x depuis ses axones, qu'il combine entre elles. Si le signal résultant Wx est supérieur à un certain seuil $-b$, le neurone est activé (et émet à son tour une nouvelle impulsion électrique). Mathématiquement, cela se met sous la forme suivante :

$$y = \sigma(Wx + b)$$

où σ est la fonction signe, renvoyant 0 si l'entrée est négative et 1 sinon. En 1969, Minsky et Papert assènent un coup terrible au perceptron en mettant en évidence les faiblesses théoriques qui rendent les perceptrons très peu efficaces (impossibilité de représenter une fonction OU EXCLUSIF). Cette découverte reléguera au placard le perceptron pour une petite décennie, jusqu'aux travaux de Rumelhart et Hinton, qui en 1986 proposent de combiner plusieurs perceptrons organisés en réseau pour former le MLP.

La véritable révolution surviendra au début des années 2010, avec des succès retentissants comme la reconnaissance d'images avec des modèles remplaçant les couches linéaires par des convolutions sur des images (comme *AlexNet* en 2012), ou la résolution du jeu de Go (avec *AlphaGo* en 2015). Aujourd'hui, les modèles véritablement utilisés en apprentissage profond n'ont plus grand-chose à voir avec le neurone biologique.

III/ Architecture d'un réseau de neurones

III.1/ Définition et structure générale

Comme expliqué précédemment, les réseaux de neurones permettent de modéliser des phénomènes complexes. Ils sont basés sur l'enchaînement de modèles linéaires et de fonctions non linéaires. Les transformations $h_i = \sigma(W_i h_{i-1} + b_i)$ sont appelées couches. On distingue la couche d'entrée, qui reçoit le vecteur d'entrée x , la couche de sortie, qui renvoie un vecteur de sortie y (sans lui appliquer de fonction non linéaire σ), et les couches cachées, qui reçoivent et renvoient des vecteurs intermédiaires h_i . Ces vecteurs intermédiaires sont appelés représentations cachées (ou

représentations latentes) et sont des variables internes au réseau de neurones. Les non-linéarités σ , quant à elles, sont appelées fonctions d'activation et nous les présenterons plus en détail dans la suite de cette partie. Il est intéressant de noter que les fonctions d'activation peuvent en théorie être différentes pour chaque couche (même si ce n'est généralement pas le cas).

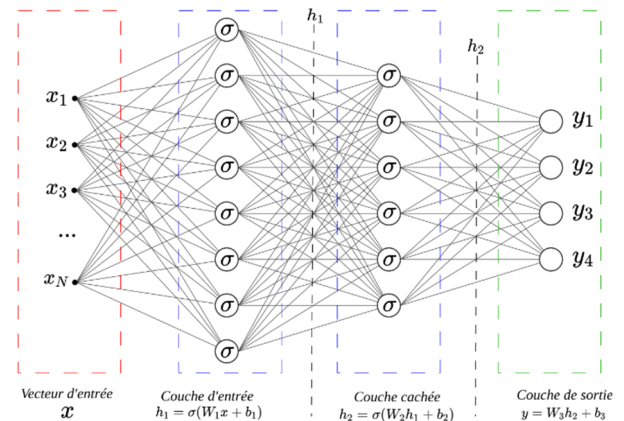


Figure 3 : Représentation classique d'un MLP.
Le modèle renvoie un vecteur de sortie y correspondant au vecteur d'entrée x donné

III.2/ Paramètres et hyperparamètres

Le réseau de neurones dispose d'un certain nombre de paramètres et hyperparamètres. Il est important de bien distinguer ces deux notions :

- les paramètres sont des éléments variables du modèle qui définissent le comportement du modèle. Les paramètres changent au cours de l'apprentissage de manière à obtenir un modèle répondant à la tâche demandée. Dans le cas du MLP, les paramètres sont l'ensemble des poids W_i et des biais b_i .
- les hyperparamètres sont des réglages du modèle et de l'algorithme d'apprentissage, choisis par le concepteur et qui ne sont pas modifiés par l'optimisation. Ils conditionnent eux aussi le comportement du modèle et doivent donc être choisis judicieusement. Dans le cas d'un MLP, les hyperparamètres sont (entre autres) le nombre de couches cachées, les dimensions des représentations cachées, les fonctions d'activations utilisées et le taux d'apprentissage.

Il est important de bien choisir les hyperparamètres de notre modèle car ces derniers influent sur la phase d'apprentissage et sur les performances finales du modèle. Il s'agit là d'une des principales difficultés de l'utilisation de réseaux de neurones. Le concepteur doit avoir une idée au moins approximative de l'influence de chacun des hyperparamètres pour pouvoir les choisir correctement.

III.3/ Les fonctions d'activation

La fonction d'activation σ utilisée sur chacune des couches est une fonction non-linéaire qui s'applique sur chaque élément des vecteurs h_i . Il existe différentes

fonctions d'activations utilisées classiquement, ayant chacune leurs avantages et leurs inconvénients. Nous n'allons en présenter ici que quelques-unes parmi les plus courantes.

- **Fonction ReLU (Rectified Linear Unit)** : Cette fonction est définie comme égale à 0 sur \mathbb{R}^- et égale à l'identité sur \mathbb{R}^+ . Il s'agit de la fonction d'activation la plus couramment utilisée dans le cas général. Il existe de nombreuses variantes de la fonction ReLU réalisant différentes approximations en 0 dans le but de la rendre dérivable. Cependant, la fonction ReLU présente l'avantage d'être linéaire par morceaux, donc rapide à calculer et présentant une dérivée simple.

- **Fonction Sigmoid** : Cette fonction est définie par

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Les valeurs renvoyées par cette fonction sont comprises entre 0 et 1. Cette fonction est souvent utilisée sur la dernière couche du réseau de neurones pour des problèmes de classification.

- **Fonction tangente hyperbolique** : Cette fonction est définie par :

$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Elle renvoie des valeurs comprises entre -1 et 1.

Elle présente principalement l'avantage de renvoyer des valeurs bornées et donc d'éviter les divergences lors des prédictions.

Une illustration de ces différentes fonctions est présentée figure 4.

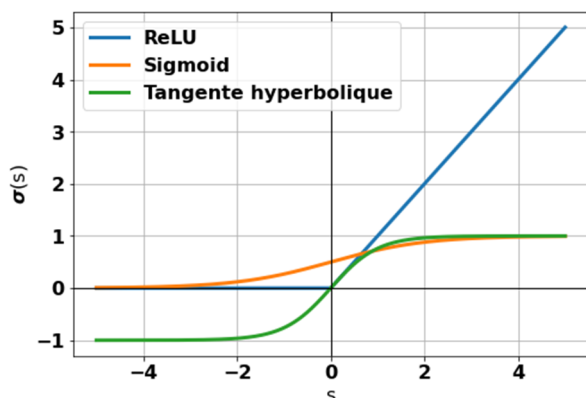


Figure 4 : Évolution des différentes fonctions d'activations présentée. Les tracés sont faits pour des valeurs des neurones comprises entre -5 et 5

III.4/ Exemple applicatif : classification de chiffres manuscrits

Pour illustrer le fonctionnement d'un réseau de neurones, nous nous intéressons à la classification de chiffres manuscrits. 60 000 exemples peuvent être obtenus grâce à la base de données MNIST. Il s'agit ici d'apprentissage supervisé, on dispose d'exemples d'entrées x_i étiquetées avec les sorties correspondantes y_i souhaitées.

Il est possible de générer un vecteur d'entrée en prenant les pixels un à un. On obtient donc un vecteur d'entrée x de taille $28 \times 28 = 784$. La sortie y est constituée de 10 valeurs, chacune associée à une classe possible comprise entre 0 et 9 (estimée par le réseau de neurones). La classe estimée correspond alors à celle pour laquelle la valeur est la plus élevée. On peut en

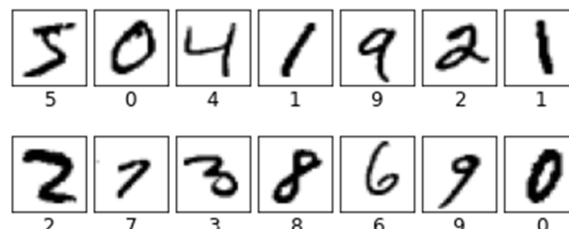


Figure 5 : Exemples d'éléments de la base de données. Les entrées sont des images de 28x28 pixels et les sorties sont des classes entre 0 et 9, correspondant au chiffre identifié.

quelque sorte voir cette valeur comme un indice de confiance qu'a le modèle en une classe : plus cette valeur est élevée (en comparaison aux autres), plus le MLP estime que cette classe est la bonne.

Avant de pouvoir entraîner le modèle (c'est-à-dire trouver les valeurs des paramètres), il nous faut choisir un certain nombre d'hyperparamètres comme le nombre de couches cachées, la taille de ces couches et la fonction d'activation à utiliser. Ce problème étant assez simple, il fonctionne assez bien avec une unique couche cachée de 200 neurones. On observe expérimentalement que la fonction d'activation ReLU permet dans ce cas d'avoir de bons résultats en limitant le nombre de calculs nécessaires et donc le temps d'apprentissage.

Cependant, nous avons expliqué précédemment que la couche de sortie est constituée de 10 valeurs qui sont comparées les unes aux autres afin de choisir une classe à prédire. Pourtant, les valeurs de la couche de sortie (dans le cas de la fonction d'activation ReLU) peuvent théoriquement prendre des valeurs infiniment grandes. Il devient alors difficile de comparer le vecteur de sortie du MLP avec un vecteur cible comme on le voudrait. On choisit alors de normaliser les valeurs de la couche de sortie entre 0 et 1 à l'aide d'une couche appelée Softmax.

La couche Softmax est une couche particulière qui ne traite pas les éléments du vecteur indépendamment. Le nouveau vecteur de sortie $Z = f(Y)$ est défini par :

$$Z_i = \frac{\exp(y_i)}{\sum_{j=1}^N \exp(y_j)}$$

Cette fonction a pour objectif de ramener les valeurs de la couche de sortie entre 0 et 1 afin de les interpréter comme des probabilités (même si ce ne sont pas vraiment des probabilités, plutôt des taux de confiance du modèle).

On dispose donc d'un modèle prenant en entrée une image x de chiffres manuscrits et renvoyant un vecteur y de 10 valeurs permettant de prédire l'appartenance à une classe. On a également un grand nombre de données d'entraînement étiquetées qui serviront d'exemples pour

entraîner le modèle (la classe d'un élément de la base d'apprentissage est convertie en un vecteur de taille 10 avec un 1 sur la classe réelle et neuf 0 partout ailleurs). À présent, il ne nous manque plus qu'à entraîner correctement les paramètres du réseau de neurones afin que ce dernier ait le comportement voulu et classe correctement nos images.

IV/ Comment entraîner un réseau de neurones ?

IV.1/ Minimiser un coût pour résoudre un problème

Dans l'exemple précédent, le réseau de neurones a pour but de classer l'image qu'on lui fournit en entrée : autrement dit, la sortie souhaitée ne peut prendre que des valeurs discrètes. On parle alors de problème de classification. (Remarquez cependant que la sortie prédite se trouve sous la forme d'un vecteur y listant les "probabilités" pour chaque classe possible.) Lorsque la sortie du modèle prend des valeurs continues, comme dans le cas d'un réseau de neurones qui doit prédire le niveau de l'eau dans 5 ans d'une zone à risque, on parle de problème de régression.

Ces deux types de problèmes sont assez similaires dans leur résolution. Seule la fonction de coût change : cette fonction évalue quantitativement la distance entre la prédiction du modèle et la réponse attendue au cours de l'entraînement, et sa minimisation à l'aide d'algorithmes itératifs permet de guider la mise à jour des valeurs des paramètres, i.e. les poids W_i et les biais pour un MLP. En voici une liste non exhaustive :

- Pour la régression : on utilise souvent une fonction de coût quadratique, qui est l'erreur quadratique moyenne entre les sorties prédites \hat{y}_i par le modèle pour les N données d'entraînement et les sorties réelles y_i correspondantes :

$$J = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

On peut aussi utiliser l'erreur absolue moyenne, ou encore la fonction de coût de Huber qui permet d'être moins sensible aux outliers (valeurs plus éloignées du modèle que les autres).

- Pour la classification : on utilise presque toujours la fonction d'entropie croisée. Dans le cas d'une classification binaire (i.e. lorsqu'il n'y a que deux classes possibles, 0 ou 1), la fonction de coût est calculée comme suit, où \hat{y}_i représente donc la "probabilité" (estimée par le modèle) qu'a le i -ième échantillon d'être dans la classe 1 :

$$J = -\frac{1}{N} \sum_{i=1}^N (y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i))$$

Si nous avons affaire à un problème de classification multi-classes, il suffit de calculer séparément l'entropie croisée pour chacune des classes à chaque donnée d'entraînement et d'en faire la somme. Mais parfois, on préfère convertir le problème en une succession de problèmes de

classification binaire avant de fusionner leurs réponses. Deux stratégies existent : la première, celle du "un-contre-le-reste", consiste à apprendre autant de classificateurs qu'il y a de classes, et chaque modèle doit estimer la probabilité d'une classe contre celles de toutes les autres. La deuxième, qui est paradoxalement préférable lorsque le nombre de classes est grand, s'appuie sur du "un-contre-un" : un classificateur est appris pour chaque paire de classes, et la classe prédite est celle qui a remporté le plus de matchs en un contre un.

La phase d'entraînement d'un réseau de neurones est donc entièrement consacrée à la minimisation d'une fonction de coût, qui représente l'écart entre les prédictions du modèle pour les données d'entraînement et les sorties réelles.

IV.2/ L'algorithme de descente du gradient

La fonction de coût ne sert pas seulement à dire au modèle s'il a bien fait son travail de prédiction. Elle permet surtout de guider l'algorithme de minimisation sur le terrain, en lui montrant comment ajuster les valeurs des paramètres à chaque itération pour tendre vers la minimisation de la fonction coût. En effet, lorsqu'un problème d'optimisation n'est pas soluble de manière déterministe, comme pour les réseaux de neurones qui représentent un problème trop complexe pour qu'on puisse calculer une solution analytique, il existe des algorithmes itératifs qui permettent de trouver une solution approchée, à condition toutefois que la fonction à minimiser soit dérivable, ce qui est le cas ici. Classiquement, on utilise l'algorithme de descente du gradient : voyons ce qu'il a sous le capot.

Visualisez-vous les yeux bandés sur le flanc d'une montagne : pour arriver en bas de la montagne, c'est-à-dire au point d'altitude minimale, vous allez évaluer du pied la pente à l'endroit où vous vous trouvez, puis vous ferez un pas dans la direction descendante, et vous recommencerez jusqu'à atteindre un creux, n'est-ce pas ? Mathématiquement, la montagne peut être vue comme la représentation graphique de la fonction coût, et le gradient d'une fonction en un point correspond grossièrement à la pente en ce point. Si le gradient est positif et élevé, cela signifie que la pente est grande et monte vers la droite, donc on peut faire un grand pas vers la gauche ; si le gradient est négatif et petit, la pente est faible et descend vers la droite, donc on fait un petit pas vers la droite (de peur de dépasser le minimum et de devoir revenir sur nos pas). Autrement dit, à chaque itération de l'algorithme, on met à jour chaque paramètre en lui retirant une valeur proportionnelle au gradient du coût en ce point (à une constante multiplicative près, appelée taux d'apprentissage η , fixée arbitrairement) :

$$W_j \leftarrow W_j - \eta \times \frac{\partial J}{\partial W_j}$$

$$b_j \leftarrow b_j - \eta \times \frac{\partial J}{\partial b_j}$$

Il faut donc pouvoir calculer la valeur du gradient de la fonction coût par rapport à chaque paramètre. Pour cela, on réalise deux passages dans le réseau de neurones à chaque itération (ou *epoch*) :

- une propagation de l'information vers l'avant (*forward propagation*): on fournit les données d'entraînement au réseau en entrée pour obtenir les valeurs des prédictions \hat{y} en sortie, en appliquant à chaque couche de neurones la formule suivante:

$$h_j = \sigma(W_j h_{j-1} + b_j)$$

(h_0 étant donc ici la donnée d'entraînement x qu'on met en entrée du modèle, et le signal h_M final représente la sortie prédite \hat{y}).

- une propagation de l'information vers l'arrière (*backward propagation*): on calcule, à partir des prédictions \hat{y} , la valeur des gradients du coût par rapport aux poids w_j et biais b_j du réseau, dont les expressions sont obtenues par la règle de la chaîne en "remontant" les couches de neurones. Prenons l'exemple du coût quadratique:

$$J = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

$$\text{où } \hat{y}_i = \sigma(W_M \cdot \sigma(W_{M-1} \dots + b_{M-1}) + b_M)$$

est la sortie prédite pour la i -ième donnée d'entraînement et s'exprime en fonction des paramètres W_j et b_j . Reformulons \hat{y}_i de manière à mettre en évidence l'approche itérative du calcul du gradient : on a donc :

$$\hat{y}_i = h_m = \sigma(W_M \cdot h_{M-1} \dots + b_{M-1})$$

pour un MLP à M couches. Puisqu'on dérive des fonctions composées, les gradients du coût par rapport aux paramètres de la dernière couche s'expriment par:

$$\frac{\partial J}{\partial W_M} = \frac{\partial J}{\partial h_M} \times \frac{\partial h_M}{\partial W_M}$$

et

$$\frac{\partial J}{\partial b_M} = \frac{\partial J}{\partial h_M} \times \frac{\partial h_M}{\partial b_M}$$

Il suffit à présent d'exprimer séparément chacune des dérivées:

$$\left\{ \begin{array}{l} \frac{\partial J}{\partial h_M} = \frac{2}{N} \sum_{i=1}^N (\hat{y}_i - y_i) \text{ pour le coût quadratique} \\ \frac{\partial h_M}{\partial W_M} = h_{M-1} \cdot \sigma'(W_M \cdot h_{M-1} + b_M) \\ \frac{\partial h_M}{\partial b_M} = \sigma'(W_M \cdot h_{M-1} + b_M) \end{array} \right.$$

Ainsi, la rétropropagation consiste surtout à calculer les termes $\frac{\partial J}{\partial h_j}$ successifs, puisque les autres gradients sont faciles à déduire. La formule de récurrence ci-dessous nous permet de le faire, en remontant les couches jusqu'à arriver au bout du réseau de neurones:

$$\frac{\partial J}{\partial h_{j-1}} = W_j \cdot \sigma'(W_j \cdot h_{j-1} + b_j) \frac{\partial J}{\partial h_j}$$

Une fois tous les gradients calculés, on peut mettre à jour les valeurs des paramètres.

L'initialisation des paramètres se fait généralement en prenant une valeur choisie au hasard dans $[-1, 1]$, suivant une loi uniforme ou normale, même s'il existe d'autres stratégies plus sophistiquées (initialisation de Xavier, He...). Le critère d'arrêt se présente souvent sous la forme d'un nombre limité d'itérations afin de limiter la durée de la phase d'entraînement. Enfin, le choix du taux d'apprentissage peut se révéler délicat: un pas trop grand pourrait vous faire louper le minimum et empêcher l'algorithme de converger, mais un pas trop petit peut aussi être pénalisant en vous faisant avancer trop lentement... À vous de tester plusieurs valeurs pour que la fonction de coût converge proprement. Souvent, il est utile de commencer l'entraînement avec un taux suffisamment élevé pour atteindre rapidement des résultats satisfaisants, puis de baisser progressivement le pas afin d'affiner le modèle.

IV.3/ En pratique, ça donne quoi ?

L'un des problèmes que présente l'algorithme de descente du gradient réside dans la possibilité qu'on reste bloqué dans un minimum local. Autrement dit, au lieu de parvenir en bas de la montagne, vous êtes malheureusement descendu dans une petite grotte et l'algorithme ne vous permet pas d'en sortir. Compte tenu des courbes d'erreurs très accidentées dessinées par les réseaux de neurones, il existe une multitude de minima locaux. De ce fait, l'apprentissage global converge rarement vers le minimum global de la fonction d'erreur lorsqu'on applique les algorithmes basés sur le gradient global. L'apprentissage avec l'algorithme de descente du gradient stochastique est une solution permettant de mieux explorer ces courbes d'erreurs: au lieu d'utiliser toutes les N données d'entraînement à chaque itération pour calculer les gradients, on ne va en prendre que quelques-uns, choisis au hasard, en mini-lots (ou *mini-batches*). Cela permet de plus d'éviter le stockage d'un trop grand nombre de données pour des fonctions ayant parfois quelques milliers de paramètres. Pour des réseaux de neurones plus petits, on peut se permettre de choisir des algorithmes d'optimisation plus coûteux en calculs, qui font intervenir des moments d'inertie (imaginez-vous à nouveau sur la montagne, le fait d'avoir un peu d'élan vous permettrait de sortir d'un creux ponctuel sur votre descente), ou des mises à jour automatiques du pas (si la pente est la même depuis un certain temps, vous pouvez vous permettre de faire des pas plus grands par exemple). Citons-en quelques-unes : Adagrad, Adam, RMSProp, Nesterov...

Certains problèmes récurrents peuvent empêcher le bon déroulement de la phase d'entraînement, notamment la disparition du gradient (*vanishing gradient*), ou son explosion (*exploding gradient*). Dans la première situation, les gradients deviennent de plus en plus petits lorsque l'algorithme progresse vers les couches inférieures du réseau, ce qui amenuise considérablement l'effet de la mise à jour des poids par descente de gradient et empêche une convergence rapide. Fixer un taux d'apprentissage plus petit, ou

choisir des fonctions d'activation σ qui ne sont pas nulles pour les valeurs négatives (comme la fonction Leaky ReLU, dont la dérivée est égale à 1 lorsque l'entrée est positive et à une petite constante lorsque l'entrée est négative). Dans la deuxième situation, les gradients deviennent de plus en plus grands, ce qui fait diverger l'algorithme: en effet, lorsqu'on a un réseau très profond, comme les gradients de chaque couche sont multipliés entre eux au cours de la *backpropagation*, on peut très rapidement avoir un gradient qui explose de manière exponentielle. Le problème peut être partiellement résolu en utilisant un taux d'apprentissage suffisamment faible, ainsi qu'en choisissant judicieusement les fonctions d'activation et la fonction de coût, ou en normalisant les données d'entraînement (i.e. en leur soustrayant leur moyenne et en les divisant par leur écart-type, et en gardant ces valeurs caractéristiques pour traiter les données de test de la même manière).

Enfin, notons l'existence du package Autograd dans Pytorch qui effectue automatiquement les calculs de gradients à partir de la fonction coût en collectant toutes les opérations de manière dynamique dans un graphe.

V/ L'apprentissage profond n'est pas l'Eldorado numérique

V.1/ De l'importance des données

Pour un industriel, les possibilités qu'offre l'apprentissage profond sont immenses : la clé de voûte, c'est une grande quantité de données sur le problème que l'on souhaite résoudre. Vous souhaitez détecter automatiquement la présence de tumeurs sur des images de scanner ? Les photos et les dossiers médicaux associés sont déjà là, dans la base de données de la clinique. Un algorithme pour savoir s'il faut accorder ou non un crédit à un client ? Les livres de compte de votre établissement bancaire feront l'affaire. Optimiser l'organisation de votre magasin pour vendre plus ? C'est facile, il suffit de jeter un œil aux tickets de caisse, facilement identifiés au profil du client grâce à sa carte de fidélité. Laissons de côté les fondements éthiques de telles pratiques (cela mériterait un article entier) et tempérons notre excitation : le deep learning n'a besoin que de données pour apprendre, il est vrai, mais encore faut-il que ces données soient propres et suffisamment nombreuses.

Le nettoyage des bases de données est un travail ennuyeux, ingrat, et pourtant absolument indispensable et terriblement chronophage. Il s'agit principalement de corriger toutes les petites anomalies qui pourraient gêner l'entraînement d'un réseau de neurones. Pour un ordinateur, "Dupon", "dupont" et "DUPONT" sont trois personnes différentes. Il s'agit également d'encoder les données en un format compréhensible pour le réseau, c'est-à-dire sous forme d'un vecteur mathématique. La question est vite réglée pour une image, qui n'est qu'une matrice de nombres, mais l'encodage du sexe d'une personne, par exemple, est déjà plus complexe. Nous avons vu dans l'exemple précédent que la solution

classique consiste à passer par un vecteur one-hot rempli de zéro à l'exception de la ligne correspond à la classe. La tâche se complique très largement lorsque l'on souhaite travailler avec des données textuelles (avis sur un site de e-commerce, description des produits, etc...). Il y a trop de mots dans le dictionnaire pour imaginer passer par un vecteur one-hot. On utilise plutôt des algorithmes appelés word2vec qui associe un vecteur à valeurs réelles et de grande dimension à chaque mot (de sorte que, la somme du vecteur "roi" et "femme" est très proche du vecteur du mot "reine").

Mais supposons que nos données soient déjà parfaitement propres : nos employés ont respecté à la lettre les règles de formatage, et celles-ci sont déjà stockées au format vectoriel. Reste encore un problème, et celui-ci est de taille : avons-nous assez de données pour entraîner un réseau de neurones ? Le jeu de données doit représenter avec suffisamment de finesse les variations de la fonction que l'on cherche à apprendre. C'est comme essayer de reproduire le plan d'une ville avec quelques photos aériennes : si vous n'avez pas assez de photos, la tâche est impossible. Il existe bien quelques techniques d'augmentation de données (symétrie des images, rotations, ...), mais en définitive, on ne peut pas faire de miracle.

On peut imaginer des variantes du problème de manque de données, dans lesquelles les données seraient riches sur un certain domaine, mais très pauvres ailleurs. L'entreprise Apple en a fait les frais en 2017. Plusieurs plaintes ont mis en évidence le fait que sa nouvelle application de déverrouillage par reconnaissance de visage était moins fiable sur les visages asiatiques que sur les visages européens. Evidemment, les développeurs de la multinationale n'ont pas volontairement codés un comportement discriminatoire dans leur application. Le problème venait en réalité des données d'entraînements, qui contiennent majoritairement des visages de personnes blanches de peau, originaires d'Amérique ou d'Europe. Le réseau de neurones n'ayant vu que très peu de visages asiatiques, celui-ci n'a pas été capable de généraliser ce qu'il avait appris sur les visages européens.

Plus facilement identifiables, les événements rares nécessitent une adaptation dans la manière de construire la fonction de coût pendant l'entraînement. Par exemple, lorsqu'une banque cherche à détecter les fraudes à la carte bancaire, celle-ci doit exploiter des bases de données contenant des millions de transactions licites, mais un nombre dérisoire de transactions officiellement détectées comme illégales. Ce déséquilibre peut par exemple être pris en compte en pondérant plus fortement les erreurs sur les événements rares dans la fonction de coût.

En définitive, tout est une question de *généralisation* : dans quelle mesure un réseau de neurones pourra prédire correctement sur des données différentes de celles vues lors de l'entraînement ? Cette question est centrale en apprentissage automatique.

V.2/ Sur- et sous-apprentissage

La mise en place d'un réseau de neurones demande d'effectuer de nombreux choix arbitraires pour choisir les hyperparamètres (nombre de couches, forme du réseau, fonctions d'activation, etc...). Ceux-ci peuvent également impacter la capacité de généralisation du réseau. Le sous-apprentissage, par exemple, apparaît lorsque le réseau de neurones ne contient pas assez de paramètres pour apprendre la fonction (voir Fig.3d). On corrige cela en augmentant le nombre de couches et la taille des représentations cachées.

Mais même en apprentissage profond, le mieux est l'ennemi du bien. En augmentant le nombre de paramètres, on peut se retrouver dans la situation contraire du sur-apprentissage (voir Fig.6c). Le réseau de neurones exploite sa grande capacité pour apprendre par cœur les données d'entraînement, et devient incapable de généraliser ce qu'il a appris aux nouvelles mesures. Le sur-apprentissage (ou *overfitting*) est une épée de Damoclès permanente qui menace (et terrifie) tous ceux travaillant en deep learning. Un algorithme peut fonctionner parfaitement en laboratoire, sur les données d'entraînement, mais être parfaitement inutilisable en situation réelle ! Heureusement, il est possible de limiter les risques en respectant quelques règles simples. La première, et la plus importante de ces règles, nécessite de découper le *dataset* en trois ensembles de mesures totalement disjoints.

L'entraînement du réseau de neurones (la descente de gradient) à proprement parler s'effectue sur la totalité des données de l'ensemble d'entraînement. Il représente entre 60 et 80% des données. L'évaluation des performances du modèle entraîné est réalisée sur un deuxième ensemble, appelé ensemble de *validation*. C'est notamment ce dernier qui servira à régler les hyperparamètres du modèle. Insistons encore un peu sur le mot "disjoint" : comme les exemples de l'ensemble de validation n'ont jamais été vus par le réseau de neurones pendant l'entraînement, on pourra détecter le sur-apprentissage lorsque l'écart entre l'erreur d'entraînement et l'erreur de validation dépassera un seuil raisonnable. Cette séparation a valeur de loi en apprentissage profond : évaluer son modèle sur des données vues en entraînement est considéré comme de la triche et de la malhonnêteté !

Dans le cas idéal, on ajoute un troisième ensemble de test permettant d'évaluer la version finale du modèle, c'est-à-dire après avoir fixé tous les hyper-paramètres. En effet, on peut également sur-apprendre les hyper-paramètres du modèle, d'où cette nouvelle étape de sécurité.

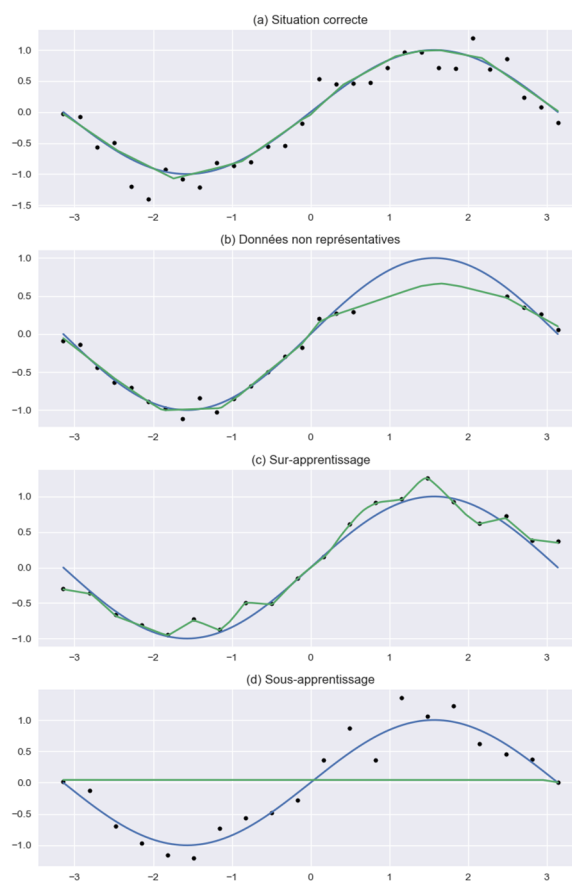


Figure 6 : Résultat d'entraînement d'un MLP à une couche cachée à partir des mesures (points noirs) issues d'une fonction sinus. (a) Les données représentent bien la fonction, et le MLP est bien choisi. (b) Certaines données ont été supprimées, le réseau y est moins performant. (c) Le MLP est sur-dimensionné, celui-ci ne généralise pas. (d) Cette fois, le réseau est sous-dimensionné et ne parvient pas à apprendre quoi que ce soit.

V.3/ Mesures de performances

Félicitations ! Vous avez surmonté toutes les difficultés liées à vos données après plusieurs heures à les nettoyer. Vous avez soigneusement fabriqué vos trois ensembles *train/validation/test* et l'algorithme de descente de gradient semble finalement avoir convergé après plusieurs heures à monopoliser votre GPU. Voici venu le moment crucial : l'évaluation des performances du modèle. Cette étape vient (bien sûr) s'effectuer sur les données de test, mais la métrique utilisée va varier selon le type de problème. Certaines tâches ont d'ailleurs des métriques qui leur sont propres, comme la détection d'objets dans une image, les problèmes de suivi de mouvement, où encore les tâches de robotique. Loin de nous l'idée d'en dresser une liste exhaustive, étudions plutôt celles que vous serez le plus probablement amené à utiliser.

Pour les problèmes de régression, il n'y a pas grand-chose de nouveau : la solution simple utilise la distance euclidienne moyenne entre votre prédiction \hat{y} et valeur attendue y . On la retrouve généralement sous l'appellation RMSE (pour *root mean squared error*) :

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^N (y_i - \hat{y}_i)^2}$$

Pour les problèmes de classification, les choses se corsent. Pour l'instant, concentrons-nous sur un problème à deux classes (ce qui correspond à une classification binaire, comme un algorithme qui détecte la présence de défauts sur une chaîne de fabrication à partir des photos d'un composant : y a-t-il des défauts, oui ou non ?). La première idée, qui vient d'ailleurs assez naturellement, serait de compter le nombre de bonnes prédictions par rapport au nombre de prédictions total, autrement dit, la précision du modèle en langage courant. C'est effectivement une bonne mesure des performances, que l'on retrouve d'ailleurs quasi systématiquement. Mais cela ne suffit pas !

A priori, la plupart des composants de la chaîne de production sont de bonne facture (disons, 90%). Les données de test traduisent également de cette répartition et contiennent 90% de photos de "bon composant". Comment dans ce cas interpréter les performances d'un réseau de neurones qui aurait appris à prédire "Bon" pour absolument tous les composants ? Sa précision serait de 90%, ce qui semble plutôt bon. Pourtant, il est évident que celui-ci est parfaitement inutile ! Il faut donc une autre métrique, et pour cela, il faut différencier les vrais positifs (les composants correctement identifiés comme défectueux) des faux positifs (les composants défectueux identifiés comme bons). On fait de même avec les vrais négatifs et les faux négatifs. Pour mesurer précisément les performances d'un algorithme de classification binaire, on introduit une matrice de confusion dans laquelle chaque ligne correspond à une classe réelle et chaque colonne, à une classe prédite. Ainsi, habituellement, la première ligne de la première colonne correspond au nombre de composants défectueux (positifs) identifiés comme tels, c'est-à-dire le nombre de vrais positifs.

Cette matrice peut se lire de plusieurs façons (cf. figure 7). La diagonale par exemple correspond à la précision dont nous discutons dans le paragraphe précédent. En lisant la dernière ligne, on évalue la capacité du modèle à détecter les pièces valides. La première colonne nous donne une indication sur la fiabilité du réseau lorsque celui-ci prédit qu'un composant est défectueux. La matrice de confusion permet de repérer immédiatement le problème décrit précédemment : si le réseau donne toujours la même réponse, alors l'une des colonnes de la matrice sera entièrement nulle.

Le terme précision en français prête à confusion dans son emploi dans un problème de classification. On y préfère d'ailleurs le mot anglais *accuracy* ou encore *exactitude* en français, car la précision définit une autre métrique qui n'a pas tout à fait la même signification. La précision est en réalité le rapport entre le nombre de vrais positifs, divisé par le nombre d'éléments détectés comme positifs. Par exemple, si la précision de votre

détecteur de défaut est de 70%, alors sur 100 pièces détectées comme à jeter, 30 d'entre elles étaient parfaitement fonctionnelles.

		Classe prédite	
		Défaut	OK
Classe réelle	Défaut	Vrais positifs	Faux négatif
	OK	Faux positifs	Vrais négatifs

Accuracy : $\frac{\text{Vrais positifs} + \text{Vrais négatifs}}{\text{Vrais positifs} + \text{Faux positifs} + \text{Faux négatifs} + \text{Vrais négatifs}}$

Précision : $\frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux positifs}}$

Rappel : $\frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux négatifs}}$

Figure 7 : Matrice de confusion et formules des différentes métriques pour les problèmes de classification.

Une troisième métrique s'intéresse aux pièces défectueuses qui ne sont pas détectées par le réseau. Il s'agit du rappel. Ainsi, un rappel de 80% indique que sur 100 composants défectueux, seulement 80 ont été correctement identifiés. Le jonglage entre précision et rappel est une gymnastique qui demande un peu de pratique pour être parfaitement à l'aise. En résumé, la précision juge le nombre de composants jetés pour rien, et le rappel le nombre de mauvais composants qui ont été ignorés. Ces métriques s'étendent aux problèmes de classification dans plusieurs catégories. On calcule dans ce cas la précision, l'*accuracy* et le rappel pour chaque classe.

VI/ Le Deep Learning en pratique

VI.1/ Structures des réseaux

Théoriquement, le MLP est à lui seul suffisamment puissant pour approcher n'importe quelle fonction que l'on souhaite apprendre. En pratique, il y a plusieurs prérogatives à cela : nous avons déjà évoqué l'importance des données, et l'influence des hyperparamètres. Quand bien même toutes ces conditions seraient réunies, il n'y a, en fait, aucune garantie pour que la descente de gradient parvienne à trouver les meilleures valeurs des poids possibles, et si possible dans un délai raisonnable. On doit donc se résoudre à inventer des structures de réseau de neurones qui facilitent l'apprentissage sur certaines tâches. C'est ce qu'on appelle la *bitter lesson* en apprentissage profond : tous les efforts que nous mettons en œuvre aujourd'hui pour inventer de nouvelles structures de modèles profond est vain, car l'évolution de domaine et l'augmentation des puissances de calculs des ordinateurs auront permis l'émergence de nouvelles méthodes plus simples qui résoudront le problème avec au moins autant de succès.

Mais aujourd'hui, le MLP ne suffit pas ! Il est cependant toujours utilisé comme brique élémentaire de construction pour des modèles plus complexes. Éloignons-nous donc maintenant des principes introductifs pour nous intéresser aux pratiques plus modernes du deep learning. Voici un petit bestiaire des modèles classiques :

Les **réseaux convolutionnels** (CNN) (ou réseau de neurones convolutifs) sont bien plus adaptés au traitement de l'image que ne l'est le MLP. Pour cela, les transformations linéaires du MLP sont remplacées par des convolutions 2D, qui ont l'avantage d'utiliser moins de paramètres, et surtout d'être invariants aux translations (déplacer un objet dans l'image ne change pas drastiquement les calculs du réseau de neurone).

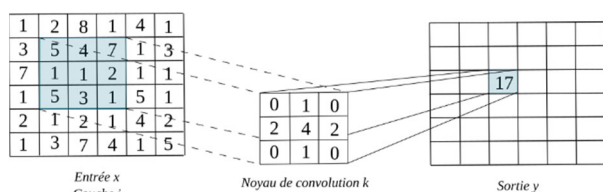


Figure 8 : Exemple d'une convolution 2D entre une matrice d'entrée x et un noyau de convolution k . Les seuls paramètres à calculer sont les poids du noyau de convolution.

Les réseaux convolutionnels sont à l'origine du regain d'intérêt pour les réseaux de neurones au début des années 2010 en surpassant de très loin les performances des méthodes de vision par ordinateur plus classiques. La pratique courante consiste à appliquer une série de couches de convolution (avec leurs fonctions d'activation) sur une image, puis d'aplatir le résultat final en un long vecteur qui passe ensuite dans un MLP qui prédit alors la grandeur d'intérêt. Il existe également quelques utilisations de réseaux convolutionnels en 1D et 3D (pour l'audio et la vidéo par exemple).

Les **réseaux récurrents** (RNN) sont adaptés à l'analyse de séries temporelles, en ajoutant une mémoire au réseau. Chaque donnée x_t de la séquence est utilisée pour mettre à jour un vecteur caché h_t

$$h_{t+1} = \sigma(W_h h_t + W_x x_t + b)$$

Ce vecteur caché peut être utilisé de plusieurs façons : pour une tâche de classification (par exemple, prédire si une action va prendre de la valeur ou non), on peut faire passer la dernière valeur de h_t dans un MLP. Ce principe est particulièrement utilisé par le traitement de flux vidéo et le traitement de l'écriture manuscrite. L'équation précédente est un peu passée de mode, et présente un mauvais comportement pendant la descente de gradient (le gradient à tendance à s'évanouir, c'est-à-dire valoir zéro, lorsque la séquence est longue). Il existe des structures plus modernes comme les Long Short-Term Memory (LSTM) ou plus récemment les réseaux Gated Recurrent Unit (GRU).

Les **auto-encodeurs** (AE) sont très pratiques dans beaucoup d'applications. Une donnée de grande dimension (généralement une image) est encodée par un réseau de neurones dans un vecteur contenant largement moins de valeurs, qui est ensuite décodé pour reconstruire la donnée d'origine. Il s'agit d'un

algorithme de compression très efficace, qui peut être détourné pour, par exemple, encoder une image et décoder l'image en profondeur correspondante. Cette structure en entonnoir est présente dans beaucoup de modèles profonds



Figure 9 : Aucun de ces visages n'existent dans la vraie vie, ils ont été générés par un GAN, un modèle à deux réseaux : un générateur d'images, et un discriminateur chargé de distinguer les vrais visages des faux.

Les **réseaux adversariaux génératif** (GAN) sont une idée astucieuse lorsque la tâche consiste à générer des données, comme des visages humains ou de la musique. Le principe consiste à entraîner deux réseaux de neurones antagonistes: le générateur fabrique des images à partir d'un vecteur d'entrée aléatoire et le discriminateur est entraîné à différencier une vraie donnée d'une donnée fabriquée par le générateur. Au début, la tâche est facile pour le discriminateur, mais se complique lorsque le générateur s'améliore. Les deux réseaux sont entraînés en simultané. Certains auto-encodeurs peuvent d'ailleurs être entraînés comme des GAN.

Tout récemment, une nouvelle structure a fait son apparition en révolutionnant les méthodes de traitement du langage : il s'agit des **transformers** (nous vous éviterons le jeu de mot avec les films de Michael Bay). Sans rentrer dans les détails, ces modèles reposent sur un mécanisme d'attention entre les mots d'une phrase. Ils sont à l'origine des immenses progrès des services de traduction en ligne, mais aussi de certains assistants virtuels.

La liste pourrait s'allonger indéfiniment, car c'est là l'une des libertés de l'apprentissage profond : toutes les structures sont bonnes à être testées. Le gros de la recherche se concentre donc sur la fabrication de modèles spécifiquement dédiés à une tâche. Nous pourrions mentionner par exemple les réseaux en graphe (GNN) qui excellent dans tous les problèmes pouvant se mettre sous cette forme particulière.

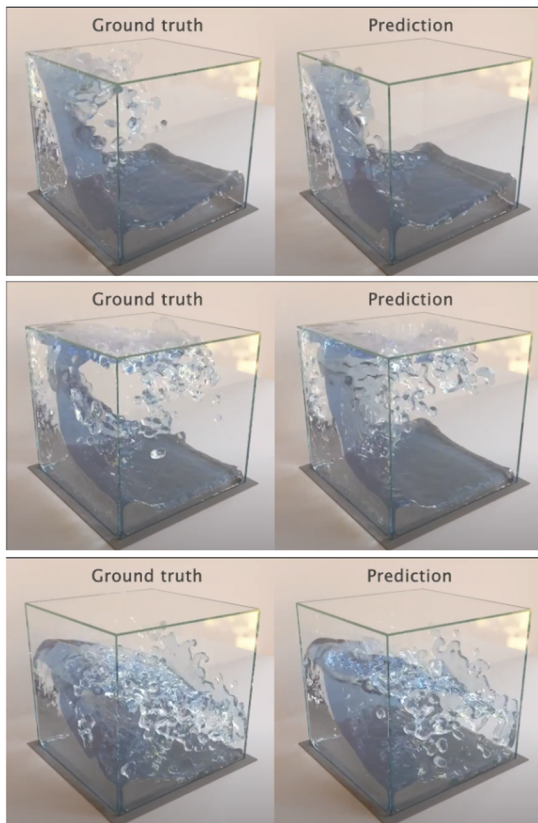


Figure 10 : Les mouvements du fluide prédit par un réseau de neurones en graphe, qui modélise le fluide par un ensemble de particules reliés entre eux.

VI.2/ Implémentation pratique

Il faut différencier la phase d'apprentissage d'un réseau de neurones de la phase d'inférence, c'est-à-dire le moment où celui-ci est réellement utilisé. L'entraînement est une étape très longue, en grande partie à cause de la descente de gradient, qui nécessite de calculer la dérivée du critère d'optimisation par rapport à un très grand nombre de paramètres. L'avantage, c'est qu'il s'agit d'une étape qui se parallélise très bien, d'où l'utilisation massive des cartes graphiques. Architecturalement, une carte graphique est composée de milliers de cœurs très simples, là où un processeur n'est composé que d'une dizaine de cœurs en moyenne, mais bien plus puissants. Le GPU permet donc d'effectuer beaucoup plus d'opérations en parallèle qu'un CPU. L'inférence, par contre, est très rapide, puisqu'il s'agit majoritairement de multiplications matricielles, ce qui rend possible l'utilisation de réseaux de neurones en temps réel, dans des véhicules autonomes par exemple.

Dans cette partie, nous allons voir comment implémenter un MLP. Nous nous limiterons ici au langage Python, qui dispose de nombreuses bibliothèques facilitant la mise en œuvre de réseaux de neurones. À vrai dire, la quasi-totalité des implémentations de réseaux de neurones se fait avec ce langage. Les bibliothèques les plus couramment utilisées pour l'implémentation de réseaux de neurones sont PyTorch (maintenu par Meta), Tensorflow

(maintenu par Google). Ces deux bibliothèques diffèrent quelque peu dans leur manière de définir un réseau de neurones, mais s'appuient sur la même fonction de base : *autograd*. Il s'agit d'une bibliothèque permettant de dériver n'importe quelle fonction codée en Python (pour peu que l'on respecte certaines règles). Sans *autograd*, le deep learning ne connaîtrait pas le succès qu'on lui connaît. Ces deux bibliothèques sont gratuites et libres, et extrêmement bien documentées en ligne.

Pour cet exemple, nous allons plutôt utiliser Scikit-learn, une bibliothèque qui regroupe des centaines de fonctions liées au machine learning. On y retrouve bien sûr le MLP, que nous allons utiliser, mais aussi l'algorithme des k-plus-proches voisins, random forest, etc... Son utilisation est extrêmement simplifiée et elle est donc idéale pour comprendre le fonctionnement des algorithmes avec de petits modèles. Cependant, pour des problèmes plus complexes, l'utilisation de bibliothèques spécialisées et optimisées comme PyTorch ou Tensorflow est obligatoire.

Dans un premier temps, on cherche à importer les données qui serviront à entraîner le réseau de neurones. Nous allons garder les chiffres manuscrits de la base de données MNIST mentionnées précédemment.

```
from sklearn.datasets import fetch_openml
import numpy as np
X, y = fetch_openml('mnist_784',
                    version=1, return_X_y=True)
```

Ces quelques lignes téléchargent automatiquement les données de MNIST et stockent les entrées X et les sorties y dans les variables correspondantes. On peut ensuite visualiser une image. Notons que les images des chiffres manuscrits sont redimensionnées en 28×28 pixels:

```
import matplotlib.pyplot as plt
plt.gray()
plt.imshow(np.reshape(np.array(X)[0,:],(28,28)))
plt.show()
```

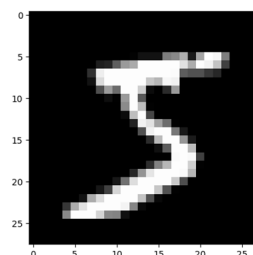


Figure 9 : Image extraite de la base de données MNIST. Il s'agit d'une image 28*28 pixels, pouvant également être représentée comme un vecteur "plat" de 784 éléments.

On doit maintenant séparer les données de notre dataset en deux sous-ensembles : *test* et *train*, pour éviter tout risque de sur-apprentissage. Scikit-learn propose une fonction pour automatiser ce découpage. Notez que nous n'utilisons pas d'ensemble de validation dans cet exemple, ce n'est pas nécessaire dans la mesure où le problème est très simple. On sélectionne ici 20% des données pour constituer l'ensemble de *test*.

L'argument "shuffle" permet de mélanger les données afin de bien répartir équitablement chaque classe dans les deux ensembles. Cela permet de limiter le biais lié à la structure de la base de données (Par exemple, tous les 9 pourraient être à la suite de tous les 8, eux-mêmes à la suite de tous les 7 etc...)

```
from sklearn.model_selection import
train_test_split
X = X / 255.
X_train, X_test, y_train, y_test =
train_test_split(X, y, shuffle=True,
test_size=0.2)
```

Une bonne pratique consiste à normaliser les entrées du réseau de neurones, cela a tendance à accélérer un peu l'apprentissage. On divise donc les images par 255. Les données sont maintenant prêtes, on peut créer le MLP et l'entraîner en lui fournissant les données connues. Plusieurs hyperparamètres sont précisés ici :

- Il y a 1 couche cachée de 50 neurones. On observe expérimentalement que cela suffit pour la reconnaissance de chiffres manuscrits.
- On réalise au maximum 10 passages sur l'ensemble de la base de données (*epoch*) d'entraînement. Cela est suffisant pour obtenir de bons résultats dans notre cas mais peut être bien plus élevé pour des applications plus complexes ou si plus de paramètres sont à déterminer
- L'algorithme d'apprentissage utilisé est une descente de gradient stochastique.

```
from sklearn.neural_network import MLPClassifier

# Définition du modèle
mlp = MLPClassifier(hidden_layer_sizes=(50, ),
max_iter=10, solver='sgd', verbose=10,
learning_rate_init=.1)

# Entraînement
model_final = mlp.fit(X_train, y_train)

# Calcul des précisions
print(f"Précision sur les données d'entraînement :
{mlp.score(X_train, y_train):.3f}")
print(f"Précision sur les données de test :
{mlp.score(X_test, y_test):.3f}")
```

Enfin, on peut visualiser une des prédictions :

```
exemple_test_x = np.array(X_test[:1])
exemple_test_y = np.array(y_test[:1])
plt.gray()
plt.imshow(exemple_test_x.reshape((28,28)))
plt.show()
print(f"Prédiction :
{mlp.predict(exemple_test_x)}")
print(f"Sortie : {exemple_test_y}")
```

Le code de cet exemple de classification des chiffres manuscrits avec la bibliothèque Scikit-learn, ainsi que le code complet de l'apprentissage sans bibliothèque, sont disponibles sur le dépôt :

https://github.com/LudovicDeMatteis/Revue3EI_IA

VII/ Conclusion

Le deep learning est un domaine en pleine expansion qui permet d'obtenir des performances très élevées dans de nombreuses tâches, notamment la reconnaissance d'images et de parole. Il repose sur des architectures de réseaux de neurones particulièrement complexes, appelés réseaux profonds. Bien qu'il soit extrêmement performant, le deep learning n'est pas exempt de difficultés et il est important de bien comprendre les principes de base avant d'en faire un usage aveugle. En particulier, il est important de disposer d'un jeu de données suffisamment important et représentatif pour pouvoir entraîner un réseau de neurones de manière efficace. Son apparition dans les programmes de l'éducation nationale permettra sans doute aux élèves et aux étudiants de mieux comprendre les principes fondamentaux du deep learning et plus généralement du machine learning, un sujet porteur qui interviendra nécessairement dans la carrière des futurs ingénieurs et techniciens. Il s'agit néanmoins d'une discipline jeune et en plein développement, il est important de suivre les évolutions en cours pour pouvoir tirer le meilleur parti des outils et des méthodes qui sont mis à disposition. Cet article est une brève introduction aux réseaux de neurones, mais leur champ d'application grandit un peu plus chaque jour, offrant son lot de nouvelles applications et de nouveaux challenges à relever !

Pour être honnête, ces derniers mots ne sont pas les nôtres, mais en réalité ceux d'une IA, qui a proposé ce paragraphe à partir de quelques extraits de l'article que vous venez de lire. Il s'agit de GPT3, un modèle basé sur les *transformers* pour le traitement du langage naturel.

Références bibliographiques

- [1] Figure 10 issue de "Learning to simulate complex physics with graph networks", de Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec et Peter Battaglia.
- [2] "L'apprentissage profond", Cours de Yann Le Cun au Collège de France, <https://www.college-de-france.fr/site/yann-lecun/course-2015-2016.htm>
- [3] "The bitter lesson", de Richard Sutton, <http://www.incompleteideas.net/Incldeas/BitterLesson.html>
- [4] Livre : "Deep Learning" de Ian Goodfellow, Yoshua Bengio et Aaron Courville, éditions MIT press.
- [5] TensorFlow Playground : <https://playground.tensorflow.org/>

APPRENTISSAGE SUPERVISÉ – COMPORTEMENT AÉRIEN D'UN DRONE

OLIVIER TOURVIEILLE

Professeur de Sciences de l'Ingénieur

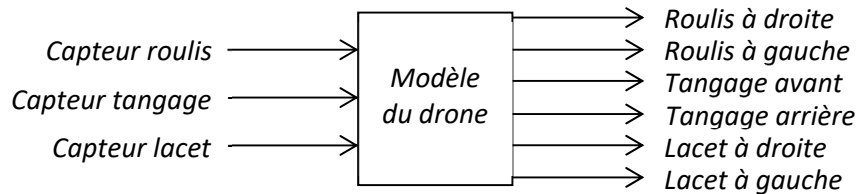
Lycée Etienne Mimard à SAINT-ETIENNE

olivier.tourvieille@ac-lyon.fr

Résumé : L'activité pédagogique suivante propose d'utiliser des mesures d'apprentissage pour produire un programme capable de déterminer automatiquement le comportement aérien d'un drone. Le système est fortement non linéaire donc difficile à modéliser avec des équations.

Les angles de roulis, de tangage et de lacet sont obtenus expérimentalement à l'aide d'un smartphone fixé sur le châssis de l'appareil. L'apprentissage automatique consiste à fournir à l'algorithme ces signaux émis par les capteurs en lui précisant à quels comportements ils correspondent. Ce procédé est appelé l'entraînement. Le résultat de l'entraînement est l'obtention d'un modèle qui permet de prédire le comportement du drone à partir de nouvelles données émises par les capteurs.

Le problème de classification des données d'apprentissage étant complexe, l'utilisation d'un réseau de neurones artificiel est bien adapté.



Les informations ressources sont surlignées en couleur grise et les réponses aux questions sont surlignées en couleur verte.

I/ Mise en situation

Le support de l'activité est un drone quadrirotor industriel remplissant des missions de sécurité publique, de secours ou d'inspection d'infrastructures.



Figure 1 : Drone DJI Matrice 300 RTK

Le pilotage de l'appareil est effectué à partir d'une télécommande ergonomique et configurable à la convenance du pilote.

En fonction des commandes de vol, le comportement aérien du drone est caractérisé suivant trois axes : tangage, roulis et lacet.

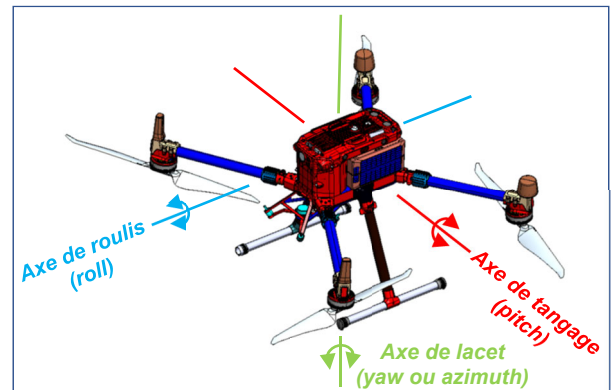


Figure 2 : Axes de vol

En vol stationnaire, les quatre rotors du drone tournent à la même vitesse et délivrent donc la même puissance. La puissance globale est suffisante pour le maintenir à une certaine hauteur sans qu'il ne descende et pas assez pour l'élever en altitude.

L'augmentation simultanée de la vitesse des quatre rotors entraîne l'élévation du drone. Inversement, la réduction simultanée de la vitesse des quatre rotors entraîne la descente de l'appareil vers la terre sous l'effet de la pesanteur.

Les quatre rotors R1, R2, R3 et R4 ne tournent pas dans le même sens. R1 et R3 tournent dans le sens horaire alors que R2 et R4 tournent dans le sens antihoraire pour annuler le couple induit par les hélices.

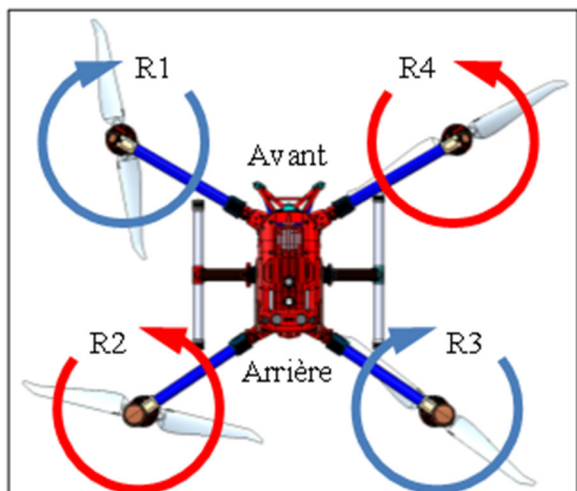


Figure 3 : Drone en vol stationnaire et désignation des rotors

Lorsque le drone avance, la puissance des deux moteurs avant se réduit et celle des deux moteurs arrière augmente (le drone s'incline légèrement vers l'avant autour de l'axe de tangage). Lorsque le drone recule, c'est exactement l'inverse.

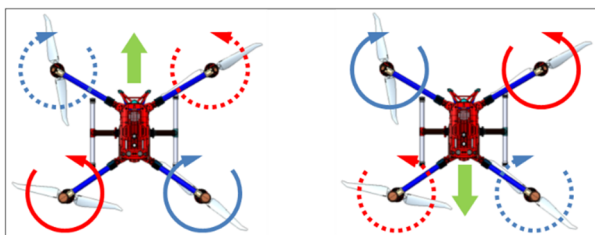


Figure 4 : Drone en vol vers l'avant et vers l'arrière

Lorsque le drone se déplace vers la gauche, la puissance des deux moteurs de gauche se réduit et celle des deux moteurs de droite augmente (le drone s'incline légèrement vers la gauche autour de l'axe de roulis). Lorsque le drone se déplace vers la droite, c'est l'inverse.

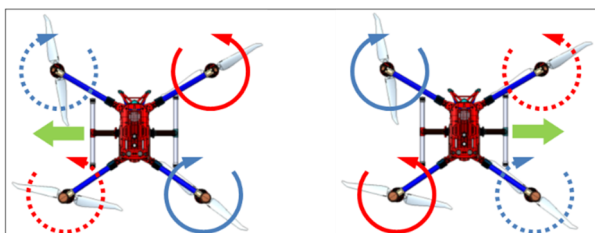


Figure 5 : Drone en vol vers la gauche et vers la droite

Lorsque le drone effectue une rotation à droite, la puissance des deux rotors 2 et 4 augmente et celle des rotors 1 et 3 se réduit (le drone pivote vers la droite autour de l'axe de lacet).

Lorsque le drone effectue une rotation à gauche, c'est exactement l'inverse. Une hélice tournant vers la gauche induit un effet de couple inverse à droite.

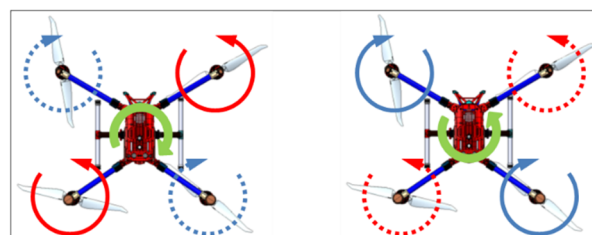


Figure 6 : Drone en vol en rotation à droite et à gauche

II/ Apprentissage automatique

II.1/ Problème à résoudre

L'activité suivante propose d'utiliser les données en provenance d'un smartphone implanté sur un drone pour produire un programme capable de déterminer automatiquement le comportement aérien de l'appareil.

II.2/ Apprentissage supervisé

L'apprentissage automatique consiste à fournir à l'algorithme un certain nombre de signaux émis par les capteurs en lui précisant à quels comportements ils correspondent. Ce procédé est appelé l'entraînement. Le résultat de l'entraînement est l'obtention d'un modèle qui permet de prédire le comportement du drone à partir de nouvelles données émises par les capteurs.

Il existe un grand nombre d'algorithmes d'apprentissage, chacun ayant ses avantages et ses inconvénients et pouvant être utilisés de manières plus ou moins satisfaisantes sur le problème donné.

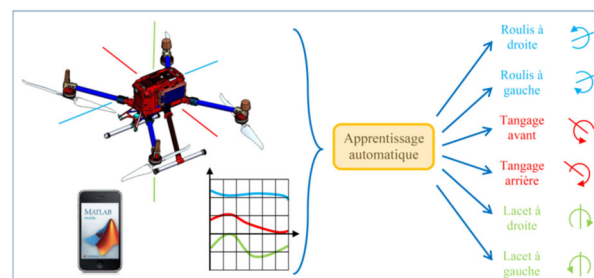


Figure 7 : Principe de l'apprentissage automatique

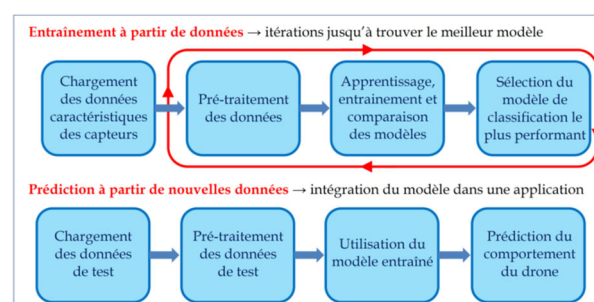


Figure 8 : Apprentissage automatique : entraînement et prédiction

III/ Données disponibles pour l'apprentissage

Les angles de roulis, de tangage et de lacet sont obtenus expérimentalement à l'aide d'un smartphone fixé sur le châssis du drone. L'application *Matlab Mobile* installée sur le smartphone permet d'acquérir les mesures issues des capteurs du téléphone et de les

déposer sur *Matlab Drive*. Ces fichiers de mesure, fournis dans l'activité proposée, seront utilisés sur un ordinateur équipé du logiciel *Matlab*.

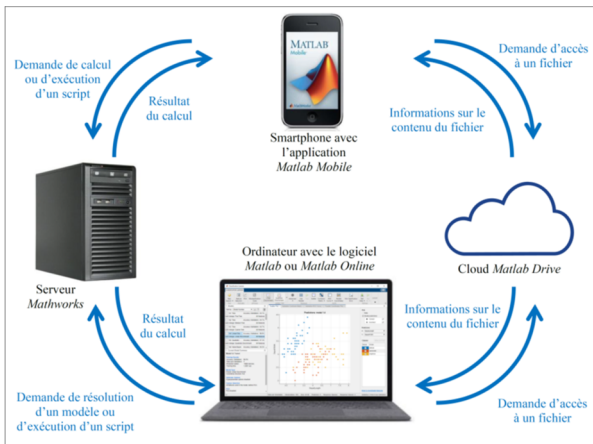


Figure 9 : Matlab Mobile et Matlab Drive

Les mesures expérimentales sont mises en forme et regroupées dans deux fichiers :

Temps seconde	Degré lacet	Degré tangage	Degré roulis	Mouvement
0	167,22252	0,19109	-0,216484	Lacet à droite
0,1	167,22252	0,191115	-0,215674	Lacet à droite
...
3,9	-154,290572	0,036468	-0,635974	Lacet à droite
6	-153,113006	0,023314	-0,638857	Lacet à droite
0	78,391707	-0,561339	0,680284	Lacet à droite
...
6	152,414346	-0,011	-0,220989	Lacet à droite
0	13,661357	-1,153675	0,15505	Lacet à droite
...
6	113,926613	-0,5125	0,088134	Lacet à droite
0	158,00573	0,094242	-0,101106	Lacet à gauche
...
6	95,036106	-0,390107	0,718008	Lacet à gauche
0	78,880229	-0,627174	0,703011	Lacet à gauche
...
6	-5,632954	-1,363217	0,246396	Lacet à gauche
0	-15,928474	-1,236114	-0,221776	Lacet à gauche
...
6	-113,463814	-0,354776	-0,426146	Lacet à gauche
0	-169,47967	0,016539	-0,052847	Roulis à droite
...
6	-169,508302	-0,078447	21,632203	Roulis à droite
0	149,105557	0,053171	0,517464	Roulis à droite
...
6	147,820907	-0,44285	16,965758	Roulis à droite
0	72,204737	-0,973995	0,944197	Roulis à droite
...
6	70,674863	-0,697175	19,779512	Roulis à droite
0	-176,185768	0,44909	0,058082	Roulis à gauche
...
6	-177,69666	-0,140202	-10,614001	Roulis à gauche
0	133,013822	-0,072082	0,779843	Roulis à gauche
...
6	131,743926	-0,405389	-21,747194	Roulis à gauche
0	66,288229	-0,685469	0,562022	Roulis à gauche
...
6	64,989277	-0,10624	-21,539632	Roulis à gauche
0	-37,251524	-0,563147	-0,157879	Tangage arrière
...
6	-36,848341	-18,444889	0,439592	Tangage arrière
0	-98,94225	0,017829	0,418502	Tangage arrière
...
6	-98,129451	-19,183582	1,073569	Tangage arrière
0	172,949296	-2,301855	-0,327971	Tangage arrière
...
6	172,90893	-19,294041	-1,065706	Tangage arrière
0	-50,589866	1,04445	0,114378	Tangage avant
...
6	-50,54062	21,681684	-0,067467	Tangage avant
0	-109,131909	-0,604242	0,81489	Tangage avant
...
6	-109,06205	21,889725	0,358349	Tangage avant
0	-160,652934	-0,868005	0,437645	Tangage avant
...
6	-160,550673	20,458146	0,258575	Tangage avant

Figure 10 : Extrait du fichier de mesures *Donneesorientations123.xlsx*

- *Donneesorientations123.xlsx* (acquisition de 3 mesures pour le tangage avant, 3 mesures pour le tangage arrière, 3 mesures pour le roulis droite, 3 mesures pour le roulis gauche, 3 mesures pour le lacet droite et 3 mesures pour le lacet gauche à la fréquence de 10 Hz et pendant une durée de 6 secondes).
- *Donneesorientations12345.xlsx* (acquisition de 5 mesures pour le tangage avant, 5 mesures pour le tangage arrière, 5 mesures pour le roulis droite, 5 mesures pour le roulis gauche, 5 mesures pour le lacet droite et 5 mesures pour le lacet gauche à la fréquence de 10 Hz et pendant une durée de 6 secondes).

IV/ Création du modèle d'apprentissage N°1

IV.1/ Utilisation de l'application *Classification Learner*

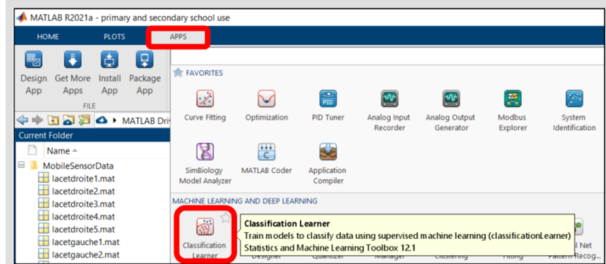
La résolution du problème s'appuiera sur l'utilisation de modèles d'apprentissages disponibles dans l'application *Matlab Classification Learner*.

L'application *Classification Learner* est un outil interactif présent dans le logiciel *Matlab* permettant d'effectuer des tâches d'apprentissage comme l'exploration de données, l'extraction de caractéristiques, l'entraînement et la validation de modèles ou l'évaluation de résultats.

L'un des principaux avantages de l'application par rapport à l'écriture de lignes de code est que celle-ci permet de comparer facilement et rapidement plusieurs modèles d'entraînement.

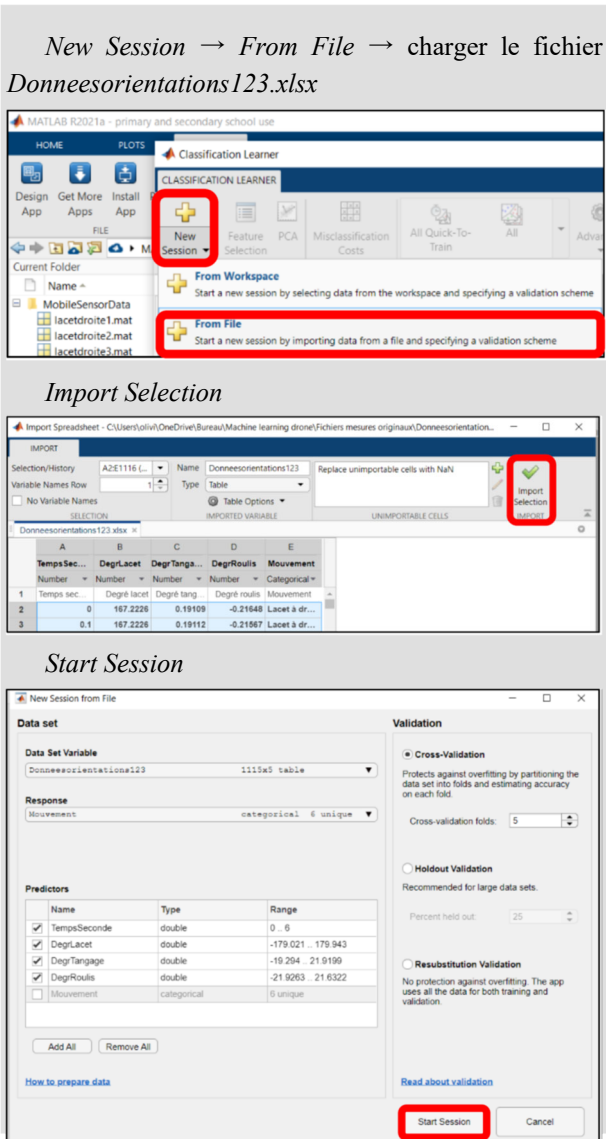
En plus, cette application génère automatiquement les lignes de code *Matlab* correspondant au modèle sélectionné. Une fois le code généré, il peut être modifié et intégré dans une application existante.

Chargement de l'application : Lancement du logiciel *Matlab* → *Apps* → *Classification Learner*



IV.2/ Importation des données d'apprentissage

Importer le premier fichier *Donneesorientations123.xlsx*



IV.3/ Choix d'un modèle d'apprentissage

Les deux modèles suivants sont sélectionnés :

- classificateurs des plus proches voisins (*Nearest Neighbor Classifiers – All KNNs*),
- classificateurs des réseaux de neurones (*Neural Network Classifiers – All NN*).

Question 1. Expliquer brièvement ces deux méthodes d'apprentissage automatique.

Classificateurs des plus proches voisins :

Un ensemble de données est utilisé pour entraîner l'algorithme dans un premier temps. Ensuite, la distance qui sépare les données d'entraînement des nouvelles données est évaluée afin de ranger ces nouvelles données dans une catégorie. Par exemple, pour la figure ci-dessous, la nouvelle donnée (rond vert) peut être rangée dans le groupe des carrés bleus (K = 1 seul proche voisin) ou dans le groupe des triangles rouges (3 plus proches voisins).

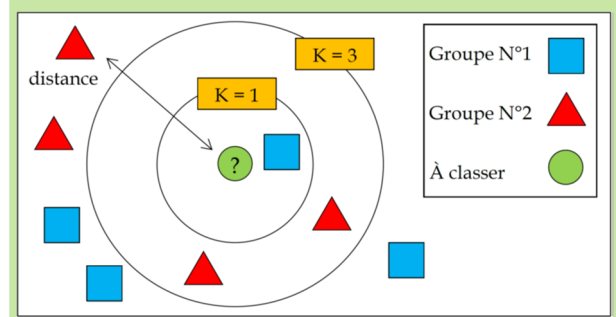


Figure 11 : Illustration du classificateur des plus proches voisins

Classificateurs des réseaux de neurones :

Un réseau de neurones s'appuie sur plusieurs couches organisées en parallèle et contenant chacune un grand nombre de neurones artificiels. La première couche reçoit les entrées d'informations brutes. La couche suivante reçoit les sorties d'informations de la couche précédente. Le dernier niveau produit les résultats de l'algorithme.

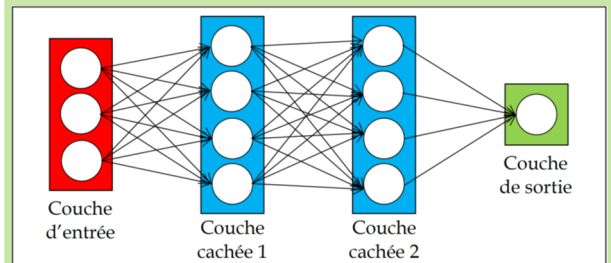


Figure 12 : Illustration du classificateur des réseaux de neurones

Question 2. Avec l'application *Classification Learner*, rechercher des schémas sur lesquels les données relatives aux angles de roulis, de tangage et de lacet sont regroupées. Après leur entraînement, relever la précision de chaque modèle (*Accuracy*).

Pour entraîner un modèle d'apprentissage, il suffit d'aller dans la galerie des modèles de classification (*Model Type*), d'en choisir un et de cliquer sur *Train*.

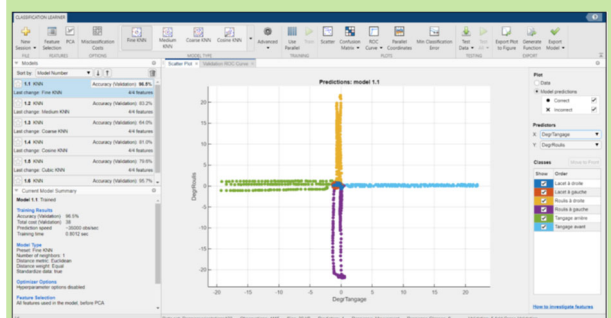


Figure 13 : Prédiction du classificateur des plus proches voisins

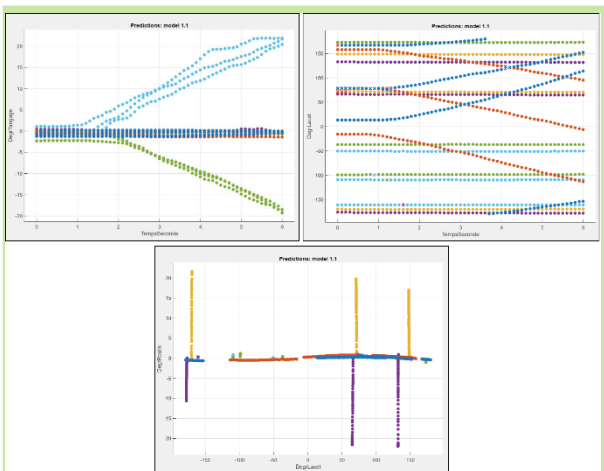


Figure 13bis : Autres représentations du classificateur des plus proches voisins

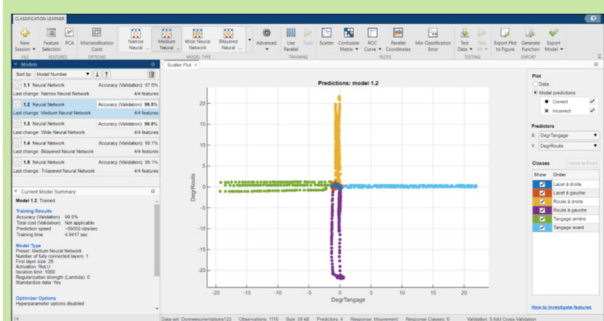


Figure 14 : Prédiction du classificateur des réseaux de neurones

Résultats classificateurs :

- plus proches voisins (*Fine KNN*) : 96,5 %
- réseaux de neurones (*Médium Neural Network*) : 99,5 %

L'information d'appartenance à un comportement du drone est exprimée par des nuages de points de couleurs différentes. Les graphiques permettent d'identifier des critères de séparations de variables.

Sur la figure 13, le comportement de tangage arrière ou de tangage avant est clairement séparé des autres. En revanche, sur la figure 13bis, certains comportements se recouvrent et sont donc moins bien séparés.

Une fois l'entraînement terminé sur des données en provenance des capteurs, le modèle apparaît à gauche, dans l'historique, avec l'indication de son niveau de précision (voir figure 13). Plus ce pourcentage est élevé, plus le modèle sera précis dans la classification de nouvelles données. Pour faciliter la comparaison de plusieurs modèles, le plus performant est surligné.

Dans le domaine de la classification, il n'existe pas un modèle qui soit plus performant que tous les autres. En fonction du problème traité, l'un ou l'autre de ces modèles sera plus adapté : arbres de décision, machines à vecteur de support, méthode des plus proches voisins, méthodes ensemblistes ou réseaux de neurones... Des paramètres définis par défaut devraient convenir à de nombreux problèmes.

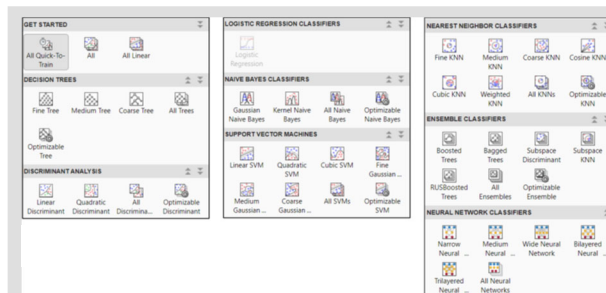


Figure 15 : Galerie des modèles de classification disponibles

IV.4/ Matrice de confusion

Question 3. Relever la matrice de confusion de chaque modèle. Comparer les résultats et rédiger quelques commentaires.

Les matrices de confusion sont des outils très utiles pour déterminer rapidement si un modèle de classification de données est efficace. Les données qui sont placées sur la diagonale sont correctement classées. Toutes celles qui sont en dehors de la diagonale ont été mal classées par le modèle. Un modèle parfait aurait un score de 100% sur la diagonale et un score de 0% partout ailleurs.

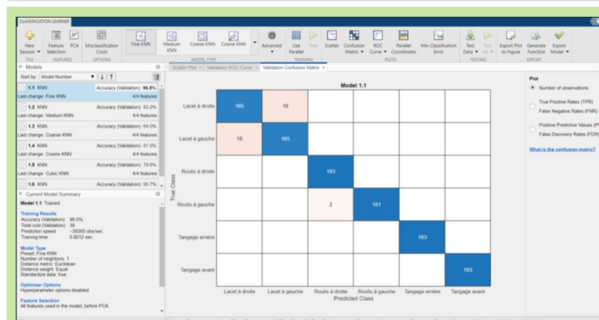
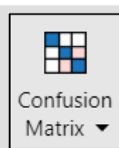


Figure 16 : Matrice de confusion du classificateur des plus proches voisins

Modèle des plus proches voisins : pour le comportement *Lacet à droite*, pour 165 cas (90 %) le modèle a correctement prédit l'activité, en revanche dans 18 cas (10 %), c'est-à-dire dans tous les autres cas, le modèle a prédit le comportement *Lacet à gauche* au lieu du comportement *Lacet à droite*. Le modèle a donc confondu ces deux comportements.

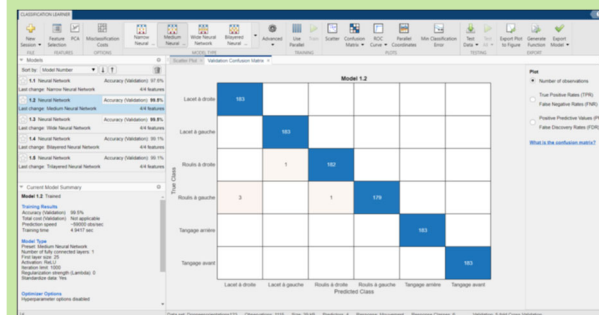


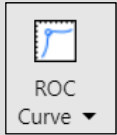
Figure 17 : Matrice de confusion du classificateur des réseaux de neurones

Modèle de réseaux de neurones : pour le comportement *Roulis à gauche*, pour 179 cas (97,8 %) le modèle a correctement prédit l'activité, en revanche dans 3 cas (1,6 %), le modèle a prédit le comportement *Lacet à droite* au lieu du comportement *Roulis à gauche*.

IV.5/ Courbe ROC

Question 4. En relevant la surface minimale sous la courbe *ROC (AUC)*, identifier la qualité des classificateurs étudiés. Comparer les performances puis rédiger quelques commentaires sur les valeurs relatives aux sensibilités et aux spécificités proposées.

Un autre type de diagnostic proposé par l'application *Classification Learner* est la courbe *ROC*. Elle utilisée pour représenter la sensibilité des classificateurs binaires. L'allure de la courbe illustre le compromis réalisé entre sensibilité et spécificité. Elle représente le nombre de vrais positifs en fonction du nombre de faux positifs



Le marqueur (point rouge) sur le tracé montre les performances du classificateur sélectionné. Le marqueur affiche les valeurs du taux de faux positifs (*FPR*) et du taux de vrais positifs (*TPR*).

Un résultat parfait sans point mal classé est un angle droit en haut à gauche du tracé. La surface sous la courbe (*AUC*) est une mesure de la qualité globale du classificateur. Des valeurs de zone sous la courbe plus grandes indiquent de meilleures performances du classificateur.

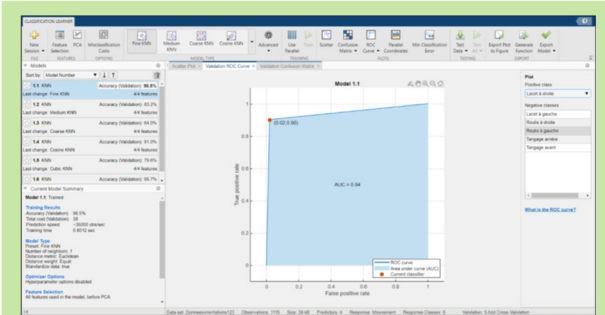


Figure 18 : Courbe ROC du classificateur des plus proches voisins (*lacet à droite*)

Pour le classificateur des plus proches voisins, un taux de faux positifs (*FPR*) de 0,02 indique que le classificateur actuel attribue de manière incorrecte 2 % des observations à la classe positive. Un vrai taux positif de 0,9 indique que le classificateur actuel attribue correctement 90 % des observations à la classe positive.

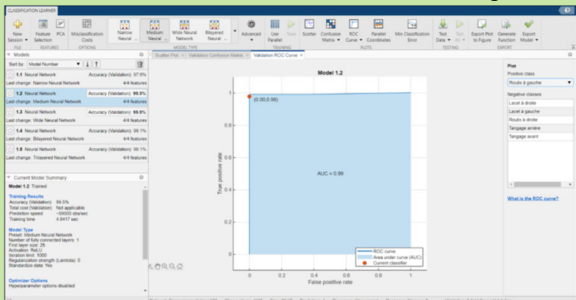


Figure 19 : Courbe ROC du classificateur des réseaux de neurones (*roulis à gauche*)

La surface sous la courbe (*AUC*) mesure la qualité globale du classificateur.

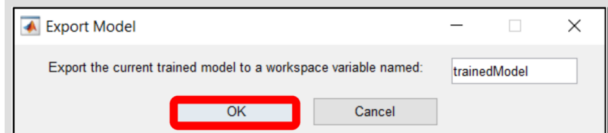
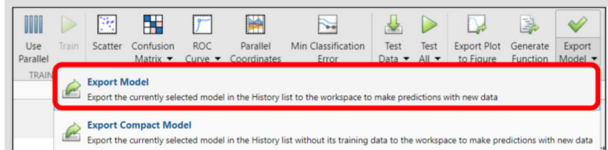
La valeur de zone sous la courbe la plus grande indique que les meilleures performances sont celle du classificateur des réseaux de neurones.

IV.6/ Exportation du modèle

Question 5. Après avoir complété le tableau de synthèse ci-dessous, sélectionner le modèle d'apprentissage le plus performant puis l'exporter pour continuer à l'exploiter sous *Matlab*.

Modèle d'apprentissage fichier N°1	Caractéristiques	
Plus proches voisins (<i>KNN</i>)	Précision (<i>accuracy</i>)	
	Qualité minimale (<i>AUC</i>)	
	Taux de faux positifs (<i>FPR</i>)	
Réseaux de neurones (<i>NN</i>)	Précision (<i>accuracy</i>)	
	Qualité minimale (<i>AUC</i>)	
	Taux de faux positifs (<i>FPR</i>)	
	Taux de vrais positifs (<i>TPR</i>)	

Après avoir sélectionné le modèle d'apprentissage le plus performant de la liste, il faut l'exporter pour continuer à l'exploiter sous *Matlab*.



Le modèle exporté apparaît dans le *Workspace* avec la désignation *trainedModel*.

Modèle d'apprentissage fichier N°1	Caractéristiques	
Plus proches voisins (<i>KNN</i>)	Précision (<i>accuracy</i>)	96,5 %
	Qualité minimale (<i>AUC</i>)	0,94
	Taux de faux positifs (<i>FPR</i>)	0,02
Réseaux de neurones (<i>NN</i>)	Précision (<i>accuracy</i>)	99,5 %
	Qualité minimale (<i>AUC</i>)	0,99
	Taux de faux positifs (<i>FPR</i>)	0
	Taux de vrais positifs (<i>TPR</i>)	0,98

Le modèle d'apprentissage le plus performant est le réseau de neurones (*Medium Neural Network*).

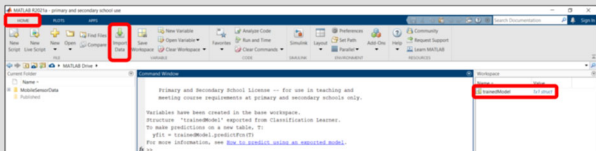
Remarque : il peut arriver que la précision du modèle soit différente entre deux séances d'apprentissage pour une même série de données.

IV.7/ Test du modèle d'apprentissage

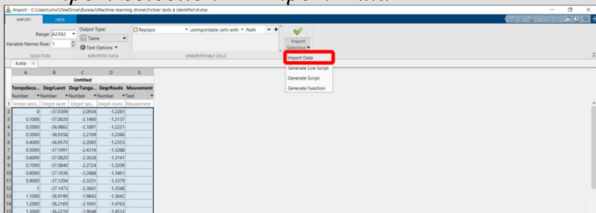
Question 6. Tester le modèle d'apprentissage sélectionné. Afficher la prédiction du modèle entraîné sous forme d'histogramme. Comparer la proposition de comportement du drone en s'appuyant sur un fichier test.

Le modèle d'entraînement *trainedModel* est utilisé pour classer des nouvelles données de test et visualiser les résultats obtenus.

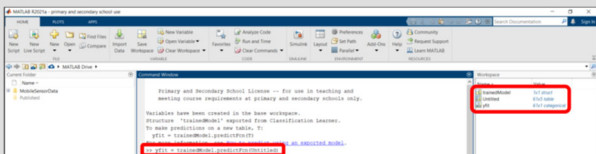
Importation des nouvelles données → *Import Data* → charger le fichier *4.xlsx*



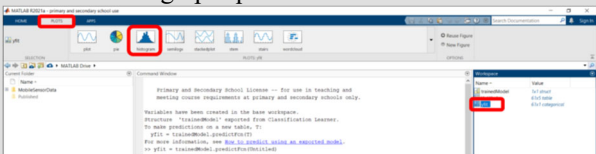
Import Selection → *Import Data*



Le fichier de données de test importé apparaît avec la désignation *Untitled*. Le modèle d'apprentissage sélectionné précédemment peut être testé avec la ligne de commande *yfit*.



La prédiction du modèle entraîné est visualisable sous forme de graphique.



Pour le tracé des courbes relatives à l'évolution des angles du drone (mesures réelles) :

- cliquer sur le fichier de données de test importé (*Untitled*) dans le *Workspace*,
- cliquer sur l'onglet *Stackeplot*.

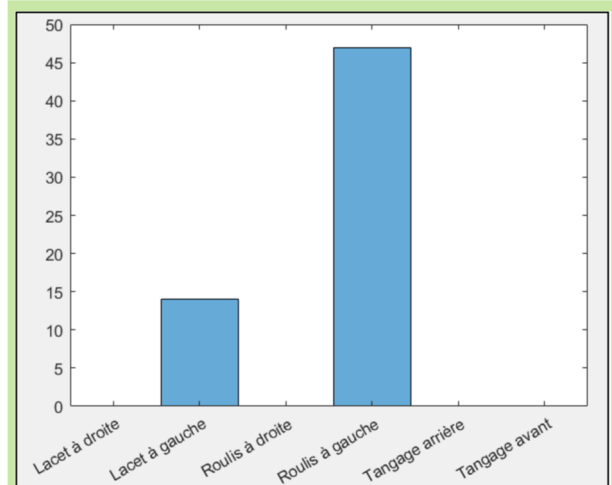
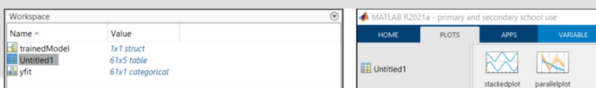


Figure 20 : Prédiction du modèle d'apprentissage à partir d'un fichier de données test 4

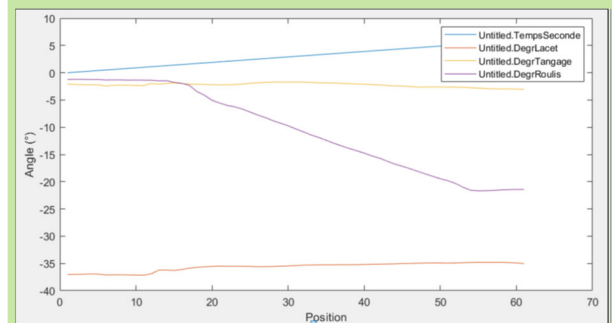


Figure 21 : Représentation graphique des données du test 4 attestant du roulis à gauche

Le modèle d'apprentissage par réseau de neurones indique que le fichier de données test est plutôt relatif à un comportement de roulis à gauche. En comparant cette prédiction à la représentation graphique des données de test, la réponse est exacte.

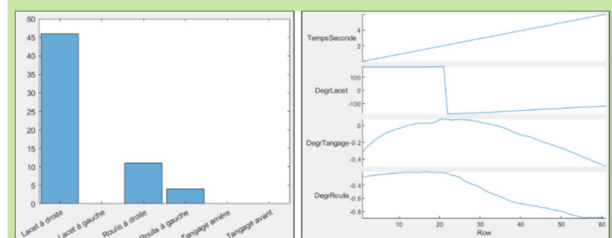


Figure 22 : Prédiction du modèle // Données de test 1 (lacet à droite)

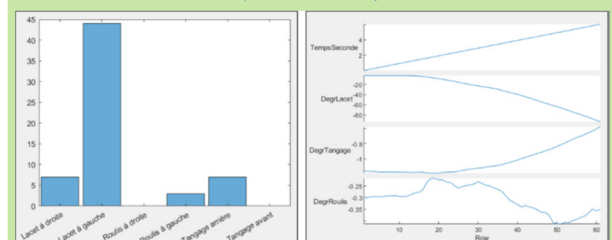


Figure 23 : Prédiction du modèle // Données de test 2 (lacet à gauche)

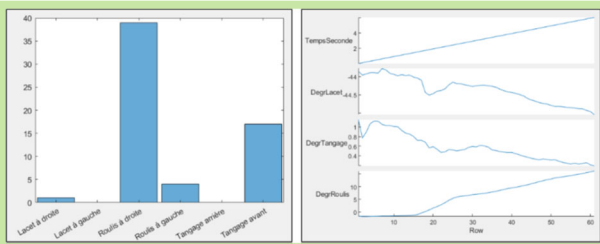


Figure 24 : Prédiction du modèle // Données de test 3 (roulis à droite)

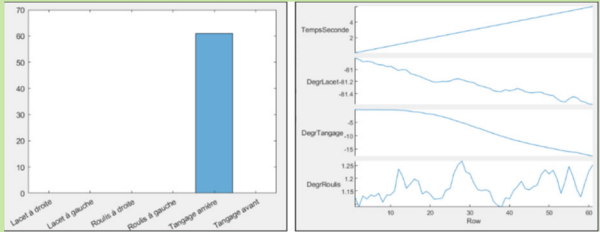


Figure 25 : Prédiction du modèle // Données de test 5 (tangage arrière)

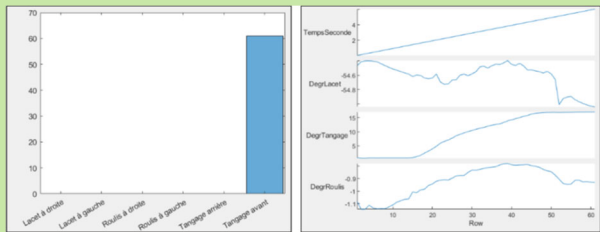


Figure 26 : Prédiction du modèle // Données de test 6 (tangage avant)

v/ Création du modèle d'apprentissage N°2

Le nouveau modèle d'apprentissage est construit à partir de l'importation du deuxième fichier *Donneesorientations12345.xlsx*

Question 7. Suivre la procédure d'analyse utilisée dans la partie précédente pour ce nouveau fichier d'apprentissage. Compléter le tableau et comparer les résultats obtenus. Rédiger quelques commentaires.

Modèle d'apprentissage fichier N°2	Caractéristiques	
Plus proches voisins (KNN)	Précision (accuracy)	
	Qualité minimale (AUC)	
	Taux de faux positifs (FPR)	
	Taux de vrais positifs (TPR)	
Réseaux de neurones (NN)	Précision (accuracy)	
	Qualité minimale (AUC)	
	Taux de faux positifs (FPR)	
	Taux de vrais positifs (TPR)	

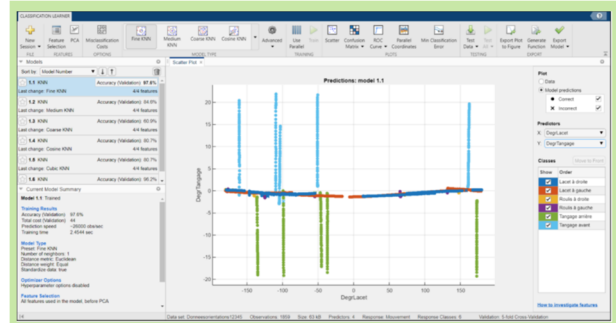


Figure 27 : Prédiction du classificateur des plus proches voisins

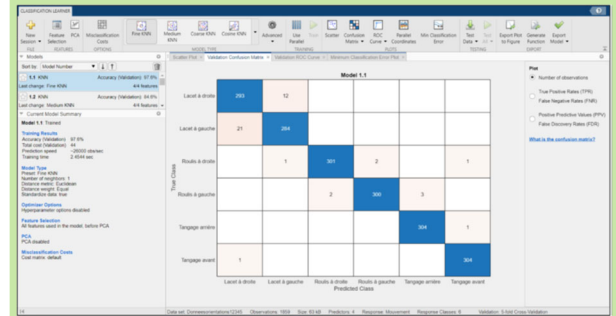


Figure 28 : Matrice de confusion du classificateur des plus proches voisins

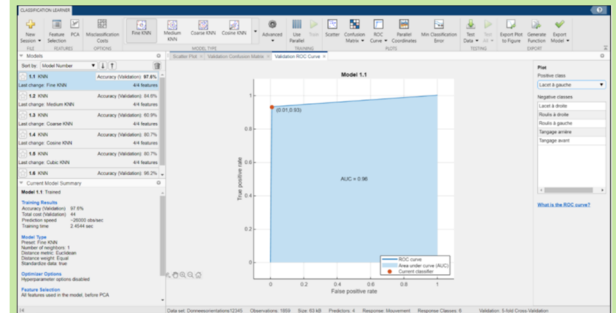


Figure 29 : Courbe ROC du classificateur des plus proches voisins (lacet à gauche)

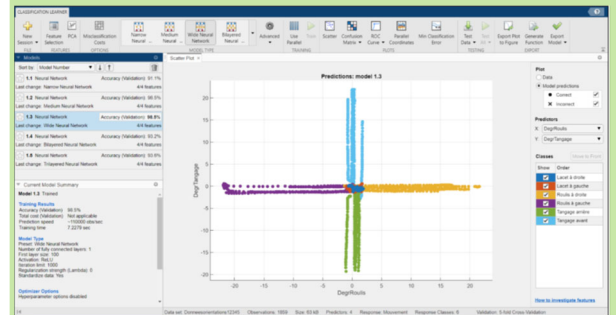


Figure 30 : Prédiction du classificateur des réseaux de neurones

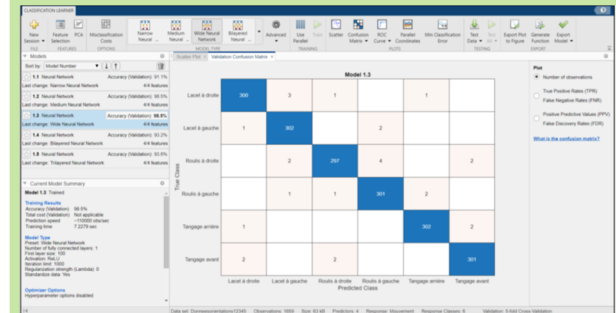


Figure 31 : Matrice de confusion du classificateur des réseaux de neurones

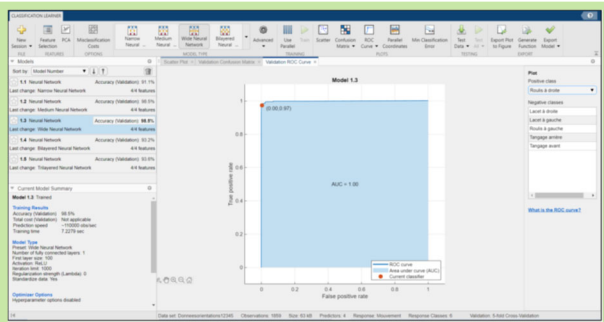


Figure 32 : Courbe ROC du classificateur des réseaux de neurones (roulés à gauche)

Modèle d'apprentissage fichier N°2	Caractéristiques	
Plus proches voisins (KNN)	Précision (accuracy)	97,6 %
	Qualité minimale (AUC)	0,96
	Taux de faux positifs (FPR)	0,01
	Taux de vrais positifs (TPR)	0,93
Réseaux de neurones (NN)	Précision (accuracy)	98,5 %
	Qualité minimale (AUC)	1
	Taux de faux positifs (FPR)	0
	Taux de vrais positifs (TPR)	0,97

Avec les mêmes données d'apprentissage, le classificateur des réseaux de neurones est toujours plus précis que le classificateur des plus proches voisins.

Question 8. Le modèle d'apprentissage des plus proches voisins est optimisable en sélectionnant l'onglet *Optimizable KNN (Nearest Neighbor Classifiers)*.

Les paramètres par défaut sont modifiables en cliquant sur le bouton *Advanced* (par exemple l'augmentation ou la réduction du nombre de voisins). Les changements effectués apparaissent dans l'historique afin de pouvoir comparer un seul et même modèle avec différents paramètres.

Lancer l'entraînement de ce nouveau modèle et relever ses caractéristiques. Rédiger quelques commentaires.

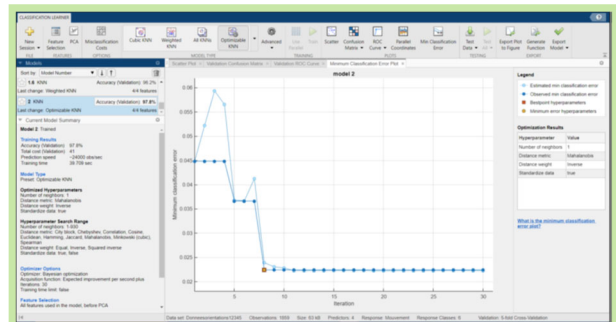
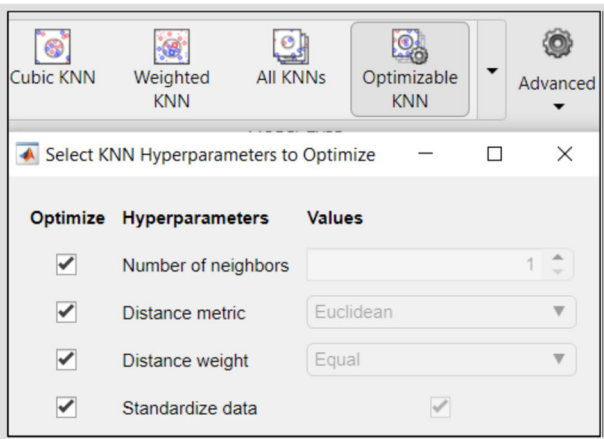


Figure 33 : Courbe d'erreur de classification minimale

Les caractéristiques d'optimisation de ce nouveau modèle sont indiquées sur la partie gauche de la courbe (la précision est de 97,8 % et l'optimisation ne révèle qu'un seul voisin (*neighbors*)). Ces caractéristiques sont ajustables à partir de l'onglet *Advanced*.

VI/ Sélection du modèle d'apprentissage optimal

Question 9. Compléter le tableau de synthèse des résultats ci-dessous et sélectionner le modèle optimal en rédigeant quelques commentaires :

Modèle d'apprentissage	Précision
Plus proches voisins N°1 (3 données d'apprentissage)	
Réseaux de neurones N°1 (3 données d'apprentissage)	
Plus proches voisins N°2 (5 données d'apprentissage)	
Réseaux de neurones N°2 (5 données d'apprentissage)	
Plus proches voisins N°3 (5 données d'apprentissage + optimisation)	

Modèle d'apprentissage	Précision
Plus proches voisins N°1 (3 données d'apprentissage)	96,5 %
Réseaux de neurones N°1 (3 données d'apprentissage)	99,5 %
Plus proches voisins N°2 (5 données d'apprentissage)	97,6 %
Réseaux de neurones N°2 (5 données d'apprentissage)	98,5 %
Plus proches voisins N°3 (5 données d'apprentissage + optimisation)	97,8 %

Le modèle optimisable des plus proches voisins est le plus précis des trois modèles *KNN* étudiés.

L'augmentation du nombre de données d'apprentissage est favorable au classificateur des plus proches voisins. En revanche, cette augmentation n'est pas à l'avantage du réseau de neurones.

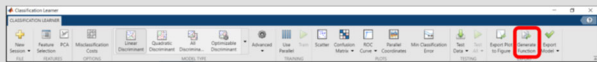
En conclusion, il n'existe pas d'emblée un modèle plus performant que tous les autres. En fonction du problème traité et des données d'apprentissage à disposition, un modèle sera plus ou moins adapté.

Pour la suite de l'étude, le modèle N°1 des réseaux de neurones sera retenu (*Medium Neural Network*).

VII/ Génération du code *Matlab* et synthèse

Question 10. Générer le code *Matlab* correspondant au modèle d'apprentissage sélectionné. Commenter les différentes parties du programme (données d'apprentissage, modèle d'entraînement et affichage des résultats).

L'application *Classification Learner* permet de générer le code *Matlab* correspondant au modèle d'apprentissage afin d'automatiser ultérieurement l'ensemble des étapes effectuées lors de l'entraînement.



Le code complet et documenté est généré de manière totalement automatique. On retrouve les différentes étapes : la sélection des prédicteurs, l'entraînement du type de modèle, le choix de la méthode de validation.... Une fois le code généré, il peut être modifié et intégré dans une application existante.

```
function [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% Returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
% Input:
%   trainingData: A table containing the same predictor and response
%   columns as those imported into the app.
%
% Output:
%   trainedClassifier: A struct containing the trained classifier. The
%   struct contains various fields with information about the trained
%   classifier.
%   trainedClassifier.predictFcn: A function to make predictions on new
%   data.
%   validationAccuracy: A double containing the accuracy in percent. In
%   the app, the History list displays this overall accuracy score for
%   each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T1, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T1)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use:
%   yf1 = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict
%
% Auto-generated by MATLAB on 01-Oct-2021 16:48:15

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'TempsSeconde', 'DegrLacet', 'DegrTangage', 'DegrRoulis'};
predictors = inputTable(:, predictorNames);
response = inputTable.Mouvement;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
```

Intégration des données d'apprentissage

```
classificationNeuralNetwork = fitnetot(...
    predictors, ...
    response, ...
    'LayerSizes', 25, ...
    'Activations', 'relu', ...
    'Units', 0, ...
    'IterationLimit', 1000, ...
    'Standardize', true, ...
    'ClassNames', categorical({'Lacet à droite'; 'Lacet à gauche'; 'Roulis à
droite'; 'Roulis à gauche'; 'Tangage arrière'; 'Tangage avant'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
neuralNetworkPredictFcn = @(x) predict(classificationNeuralNetwork, x);
trainedClassifier.predictFcn = @(x) neuralNetworkPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'DegrLacet', 'DegrRoulis', 'DegrTangage',
'TempsSeconde'};
trainedClassifier.ClassificationNeuralNetwork = classificationNeuralNetwork;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner app.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new table, T,
use: An yfit = c.predictFcn(T) (replacing 'c' with the name of the variable
that is this struct, e.g. 'trainedModel'). An NxM table, T, must contain the
variables returned by: An c.RequiredVariables. AnVariable format (e.g.
matrix/vector, datatype) must match the original training data. Additional
variables are ignored. An NxM or more information, see c.Help='mdl:helpview
(fullfile(cocroot, 'stats', 'stats.map'))';
'appClassificationExportToCodeToWorkspace')>>'How to predict using an exported
model</>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'TempsSeconde', 'DegrLacet', 'DegrTangage', 'DegrRoulis'};
predictors = inputTable(:, predictorNames);
response = inputTable.Mouvement;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationNeuralNetwork, 'kfold',
5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'Classification');
```

Entraînement du classificateur par réseaux de neurones

Extraction des prédictions et des performances du modèle

Question 11. Commenter l'étude d'apprentissage automatique réalisée. Proposer une démarche ou un protocole permettant de qualifier plus complètement le comportement aérien du drone.

L'activité d'apprentissage automatique réalisée est traitée avec les angles d'orientation du drone. Ces angles sont facilement vérifiables dans les tableaux des valeurs tests. L'utilisation de l'application *Classification Learner* permet une rapide appréciation du comportement.

En complément de l'application *Matlab Mobile* utilisée dans le TP, il existe d'autres applications de mesures disponibles sur smartphone comme *Physics Toolbox Sensor Suite* ou *Phyphox*.

Les techniques d'apprentissage automatique sont d'autant plus efficaces lorsqu'il est difficile de formaliser la relation liant les données d'entrée et de sortie.

Ainsi, par exemple, en mesurant les accélérations a_x , a_y et a_z (m/s^2), les vitesses angulaires ω_x , ω_y et ω_z (rad/s), les positions en latitude ($^\circ$), longitude ($^\circ$), la vitesse (m/s), ... il serait possible de caractériser plus complètement le comportement aérien du drone : déplacement vers l'avant en montée à droite, déplacement vers la gauche en descente accélérée... mais l'apprentissage serait plus long.

VIII/ Bibliographie

- [1] Machine Learning avec Matlab :
<https://fr.mathworks.com/campaigns/offers/machine-learning-with-matlab.html>
- [2] Introduction à l'apprentissage automatique :
<https://fr.mathworks.com/videos/series/introduction-to-machine-learning.html>
- [3] Modèles d'apprentissage automatique :
https://fr.mathworks.com/discovery/machine-learning-models.html?source=15576&s_eid=psm_15576

STABILISATION D'UN PENDULE INVERSÉ À L'AIDE D'UN APPRENTISSAGE PAR RENFORCEMENT

CHÉROT GUÉNOLÉ, GALLOIS MAEL

guenole.cherot@ens-rennes.fr, gallois.mael@ens-rennes.fr

Résumé : Cet article propose d'illustrer à travers une séance de travaux pratiques, les principes de base de l'apprentissage par renforcement.

I/ Déroulement de la séance de travaux pratiques

Nous commencerons tout d'abord par rappeler les concepts de base de l'apprentissage par renforcement. Cette partie théorique présentera les notions d'action, d'observation, d'état, de récompense et de politique. Puis nous montrerons comment formaliser le redressement et la stabilisation d'un pendule simple sous la forme d'un problème d'apprentissage par renforcement. Enfin, une séance de travaux pratiques mêlant simulations et expérimentations sur un système réel permettra d'illustrer la pertinence de la méthode proposée. Nous discuterons en particulier de l'importance du modèle utilisé pour modéliser l'environnement. L'activité pratique permettra également de mettre en évidence la différence entre phases d'apprentissage et d'inférence.

La phase d'expérimentation se base sur Control'X et Matlab - fichiers disponibles sur Gitlab¹. Cependant, la partie simulation peut également se faire sur python via google colab² - dirigée principalement vers l'utilisation de la librairie Gym.

Cet article est à destination d'enseignants ou de chercheurs désirant se familiariser avec les concepts de l'apprentissage par renforcement. Il ne nécessite pas de connaissances préalables. Le lecteur curieux pourra approfondir grâce à l'excellent ouvrage de Sutton & Barto [1]. Les activités pratiques peuvent être effectuées par des élèves de lycée ou de classe préparatoire même si l'apprentissage par renforcement n'est pas explicitement au programme. Les points du programme de classe préparatoire abordés sont listés dans le tableau à la fin de cet article.

II/ Contextualisation

Handle (*figure 1*) est un robot à deux roues développé par l'entreprise Boston Dynamics. Dans sa dernière version, il est stabilisé grâce à des asservissements complexes. Nous étudierons ici la faisabilité de remplacer tout ou partie de ces algorithmes par une intelligence artificielle.

Pour cette activité pratique, nous étudierons uniquement la stabilisation du robot sur l'axe avant/arrière. Le problème est donc plan. Le robot

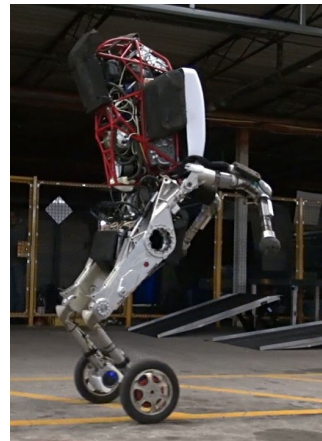


Figure 1 : Photo du robot Handle de Boston Dynamics

Handle n'étant pas disponible dans notre laboratoire, nous illustrerons la démarche à l'aide d'un système plus simple : un pendule inversé monté sur un chariot mobile. Nous utiliserons le système Control'X dont une illustration est donnée *figure 2*.



Figure 2 : Control'X + Pendule

III/ Formalisme général

III.1/ Théorie

L'apprentissage par renforcement est la science de la **prise de décisions séquentielles**. Un agent autonome (robot, joueur...) cherche à **maximiser la somme des récompenses** qu'il obtient en interagissant avec un environnement. A chaque itération, l'agent reçoit une **observation** et une **récompense** (scalaire) de la part de l'environnement. Il doit ensuite choisir une **action** qui lui permet de maximiser la somme des récompenses perçues. L'interaction entre un agent et son environnement est décrite *figure 3*.

¹ https://gitlab.com/Guenole.cherot/controlx_rl

² https://colab.research.google.com/drive/1AZh_FSSlmncpOuy_vdGbewK6OyqksAme?usp=sharing

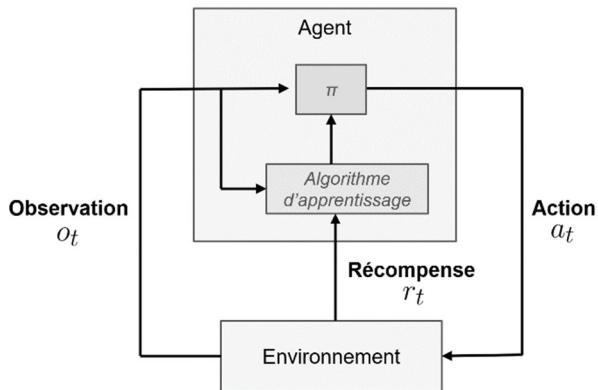


Figure 3 : Interaction entre l'agent et l'environnement

Pour prendre ses décisions, l'agent a besoin d'une stratégie de commande. Dans le formalisme de l'apprentissage par renforcement, on la nomme **politique**, noté π . La politique est une fonction qui à une observation O associe une action $\pi(o_t) = a_t$. On pourrait aussi utiliser les termes loi de contrôle. N'importe quelle fonction paramétrable peut être utilisée pour représenter la politique, mais en pratique un **réseau de neurones artificiels** est souvent utilisé.

Pour améliorer sa politique, c'est-à-dire la façon dont il prend ses décisions, l'agent a besoin d'un **algorithme d'apprentissage**. Il existe une très grande diversité d'algorithmes [2]. Nous ne les étudierons pas ici car ils sont souvent complexes et nécessitent un grand nombre de connaissances théoriques.

Le problème mathématique s'exprime donc de la manière suivante :

$$\pi^* = \operatorname{argmax}_{\pi} \sum_t^T r_{\pi}(t)$$

Où T représente le temps de fin d'un épisode. C'est-à-dire le moment auquel l'expérience s'interrompt soit parce que le temps est écoulé soit parce qu'il n'y a plus d'action faisable. Si l'expérience n'a pas de fin alors $T = \infty$.

Cette formulation très générale permet de traiter une grande variété de problèmes. Nous l'appliquerons dans la suite à la stabilisation d'un pendule inverse.

III.2/ Importance de l'observation

L'agent ne peut observer l'environnement que par le biais des observations. Dans une approche système, les observations peuvent être vues comme des sorties de capteurs. Pour que l'agent converge vers la politique optimale (celle qui maximise la somme des récompenses), il est nécessaire de lui donner les bonnes observations. C'est-à-dire, celles qui permettent de décrire le fonctionnement de l'environnement de manière assez précise. Par exemple, il est impossible de contrôler un pendule si on ne connaît pas sa position angulaire.

Nous verrons plus en détail l'importance de l'observation dans la suite de l'article.

III.3/ Parallèle avec le formalisme habituel du contrôle-commande

Le lecteur familier avec l'automatique aura sûrement fait le rapprochement entre la *figure 3* et les systèmes asservis. Habituellement, un système est dit asservi s'il répond aux trois conditions suivantes : Un opérateur impose une **consigne** au système. Des **capteurs** mesurent la valeur de la grandeur asservie. Une forme d'intelligence cherche à **annuler l'écart entre la consigne et la grandeur mesurée**. Cette section a donc pour but de comparer trois types d'asservissements (illustrée *figure 6*, annexe) :

- Système en boucle fermée « classique »
- Retour d'état - Placement de pôles
- Apprentissage par renforcement

III.3.1/ Système en boucle fermée

Ce type de structure est illustré à gauche de la *figure 6*. C'est le contexte classique des systèmes asservis. Un opérateur applique une **consigne** au système. Des **capteurs** mesurent les grandeurs d'intérêt. La régulation est effectuée par le couple {soustracteur + correcteur}. **L'intelligence est placée dans le correcteur**. Si le but est d'asservir un pendule en position, alors il faut mesurer l'écart entre la position souhaitée et la position réelle. Le correcteur va ensuite calculer la commande telle que l'erreur soit nulle. C'est adapté pour l'asservissement de grandeurs scalaires, pour des systèmes linéaires et invariants.

III.3.2/ Retour d'état - Placement de pôles

Ce type de commande est illustré au centre de la *figure 6*. L'idée générale est de changer les propriétés du système en changeant l'endroit de ses pôles. Il peut alors être plus rapide, ou voir un de ses équilibres instables devenir stable.

L'opérateur **n'applique pas directement une consigne** sur le système. Il a plutôt un cahier des charges qui spécifie des exigences de performances. Des **capteurs** mesurent l'état du système. L'état est souvent un vecteur de dimension supérieure à 1, par exemple dans le cas du pendule ce pourrait être la position et la vitesse du pendule. **L'intelligence est placée dans le calcul du vecteur K** , qui réalise le retour d'état. Dans le cas du pendule, une fois le vecteur K calculé, le pendule est stable en position haute mais on ne peut pas changer la position d'équilibre dynamiquement. Il faudrait recalculer K pour obtenir un nouveau comportement.

III.3.3/ Apprentissage par renforcement

Ce type de commande est illustré à droite de la *figure 6*. Comme nous allons le voir, le fonctionnement est très proche de celui d'un contrôle par retour d'état. Une comparaison plus complète est faite dans une série d'articles Matlab [3].

L'opérateur **n'applique pas directement une consigne** sur le système. Il doit spécifier une fonction récompense dont la maximisation fait émerger le comportement souhaité. Certains problèmes s'y prêtent mieux que d'autres. Pour les échecs par exemple, il est

très facile de donner l'objectif sous forme de récompense : +1 si l'agent gagne une partie, -1 s'il perd, 0 si égalité. En revanche, il est difficile de formaliser ce même problème à l'aide d'une consigne.

Des **capteurs** permettent d'observer l'état du système. On nomme observation, l'ensemble des données issues des capteurs.

L'**intelligence est placée dans la politique et sa mise à jour**. C'est grâce à la politique que l'agent décide des actions à appliquer en fonction de l'observation reçue : $\pi(o_t) = a_t$. Comme pour le contrôle par retour d'état, nous pouvons considérer que le système est asservi car l'agent n'est pas en boucle ouverte, il sait -au moins en partie- ce qu'il se passe dans l'environnement.

IV/ Formalisation du redressement et de la stabilisation du pendule inverse

Nous allons maintenant formaliser le redressement et la stabilisation d'un pendule inversé sous la forme d'un problème d'apprentissage par renforcement.

Le schéma cinématique du pendule et du chariot mobile est donné *figure 4*.

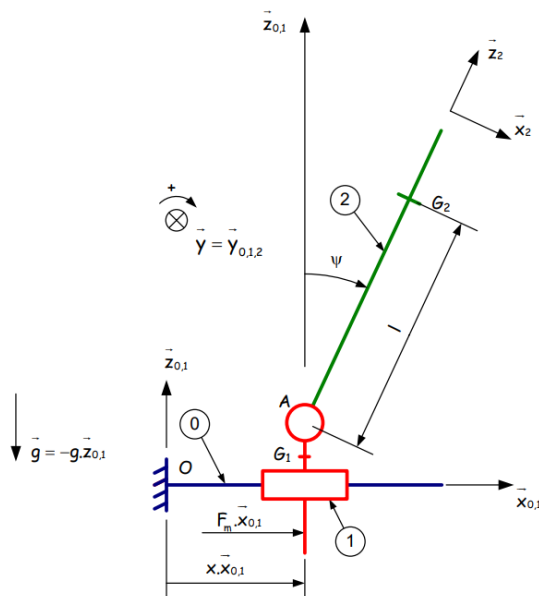


Figure 4 : Schéma cinématique de l'ensemble {chariot + pendule}

IV.1/ Action

L'agent interagit avec le Control'X via la tension appliquée sur la machine à courant continu. Cette tension est comprise entre +/- 10V. L'espace d'action est donc : $\mathcal{A} = [-10, 10]$. C'est la seule manière pour l'agent d'interagir avec le Control'X.

IV.2/ Récompense

La récompense joue un rôle central dans l'apprentissage par renforcement. Pour rappel, l'agent ne reçoit pas de consigne explicite. Son comportement émerge de la maximisation de la somme des récompenses qu'il reçoit.

La récompense est un signal **scalaire** qui doit être conçu par l'ingénieur ou le chercheur. Pour concevoir un bon signal récompense, il faut avoir l'idée suivante en tête :

La récompense doit récompenser le but et non la manière d'atteindre ce but [4]

Par exemple, pour entraîner l'agent à jouer aux échecs, il faut le récompenser quand il gagne une partie et le pénaliser quand il perd (**But** = gagner des parties). Il peut être tentant de l'inciter à occuper le centre, ou à échanger des pions contre des pièces de plus grandes valeurs (**stratégies** pour gagner), mais cela pourrait conduire à des comportements contre-productifs comme occuper le centre mais perdre systématiquement.

Pour ce TP, nous cherchons à stabiliser le pendule en position haute, nous étudierons deux types de fonctions récompenses selon si nous souhaitons relever et stabiliser le pendule ou uniquement le stabiliser.

IV.2.1/ Initialisation du pendule en position haute

Commençons par la version simple du problème : stabiliser le pendule lorsqu'il est déjà en position haute. Dans ce cadre, la simulation débute avec le pendule en position haute, $\psi = 0$, et termine quand l'angle du pendule dépasse 15° .

Question :

Proposer une fonction récompense à maximiser. La récompense peut (ou pas) dépendre de l'état du système.

Correction :

Une récompense constante peut fonctionner. L'agent reçoit $r = 1$ à chaque fois qu'il choisit une action, quelque soit l'action ou l'état du système.

Pour maximiser la somme des récompenses sur un épisode (une simulation) l'agent doit faire durer la simulation le plus longtemps possible, donc trouver la stratégie permettant de garder l'angle du pendule entre +/- 15° . Cela revient à stabiliser le pendule en position haute.

IV.2.2/ Initialisation du pendule en position basse

Prenons maintenant le problème complet : **relever** puis **stabiliser** le pendule. Chaque épisode commence avec le pendule en position basse. Cette configuration est beaucoup plus pratique si on réalise tout l'apprentissage sur le système réel (sans modèle) car redresser le pendule au début de chaque épisode est fastidieux. L'épisode termine quand le chariot arrive en fin de course ($|x| > x_{max}$) ou que la simulation dépasse 4 secondes ($t > T_{max}$).

Question :

Proposer une fonction récompense à maximiser. La récompense peut (ou pas) dépendre de l'état du système.

Correction :

Il est tout d'abord intéressant de noter que la fonction récompense proposée à la question précédente ne fonctionnerait pas. Rester sur place conduirait à la maximisation de la récompense. L'agent n'est pas incité à stabiliser le pendule en position haute.

Une solution est de donner une récompense proportionnelle à l'énergie potentielle du pendule. Elle vaut $E_c = \frac{1}{2} m l \cos(\psi)$. Il est souvent judicieux de normaliser la récompense (le formalisme utilisé pour améliorer la politique utilise des lois de probabilités centrées réduites). On peut choisir :

$$r = (\cos(\psi) + 1)/2.$$

La récompense est donc comprise entre 0 et 1. De manière générale, l'apprentissage est assez robuste aux choix de la fonction récompense. Dans le cas considéré ici, toute fonction paire, concave, et ayant un maximum en $\theta = 0^\circ$ fonctionne en théorie. Nous invitons le lecteur intéressé à regarder la publication [5]. Cependant, le choix de la récompense peut grandement influencer la vitesse d'apprentissage. Par exemple, choisir la fonction $r = \cos(\psi)$ n'est pas judicieux car en position initiale ($\psi = \pi$), la récompense vaut $r = -1$. Si l'agent cherche à relever le pendule sans sortir de la course ($|x| < x_{max}$) alors il va dans un premier temps accumuler beaucoup de récompenses négatives. Pour maximiser la somme des récompenses, il peut être tenté de chercher à terminer la simulation le plus rapidement possible en sortant des limites. Il y a donc un fort risque qu'il reste bloqué dans une stratégie sous-optimale, sans jamais se rendre compte qu'il est possible d'obtenir des récompenses positives en relevant le pendule.

Il est possible de modifier la fonction récompense pour pénaliser les changements de consigne trop brusques : $r = (\cos(\psi) + 1)/2 + r_{pénalité}$. En effet les à-coups peuvent endommager les machines, la courroie ou toutes autres parties mécaniques du système. Cette pénalisation peut prendre la forme suivante : $r_{pénalité} = -\left(\frac{a_t - a_{t-1}}{2 * a_{max}}\right)^2$, avec a la consigne appliquée sur la machine.

IV.3/ Algorithme d'apprentissage

Nous utiliserons l'algorithme « Soft Actor Critic » [6]. Il n'est pas nécessaire de comprendre le fonctionnement de cet algorithme pour la suite.

IV.4/ Observation

La position du chariot est repérée par la variable x , sa vitesse par \dot{x} . La position angulaire du pendule est donnée par ψ , sa vitesse angulaire par $\dot{\psi}$. L'espace d'observation est définie par la position linéaire et angulaire du pendule et sa vitesse linéaire et angulaire. L'observation prend donc la forme suivante :

$$o = (x, \dot{x}, \psi, \dot{\psi})$$

Si l'agent est pénalisé en cas d'écarts de consigne

³ Pour rappel, l'agent prend des décisions uniquement en fonction de la dernière observation qu'il a reçu. S'il ne sait pas

trop grands ($r = \cos(\psi) + r_{pénalité}$), alors il faut ajouter la dernière action à l'observation. Sinon, l'agent ne pourra pas être en mesure de comprendre pourquoi il est pénalisé³ :

$$o = (x, \dot{x}, \psi, \dot{\psi}, a)$$

Cette formulation pose cependant un léger problème. La position du pendule est définie modulo 2π . Ainsi, lorsque le pendule est en position basse, ψ passe de $+\pi$ à $-\pi$ alors que ces états sont très proches. Cela rend l'apprentissage plus long car l'agent doit *apprendre* que ces états sont très proches physiquement. Afin de pallier ce problème, nous remplaçons la variable d'état ψ par le couple $(\sin \psi, \cos \psi)$. Finalement l'espace d'observation est :

$$o = (x, \dot{x}, \sin(\psi), \cos(\psi), \dot{\psi}, a) \in \mathcal{O}$$

V/ Préparer l'apprentissage

V.1/ Se mettre à la place de l'agent

Dans cette partie, nous vous proposons de prendre un instant la place de l'agent.

1. Ouvrir le fichier *jeux.exe* (figure 7).
2. Les informations auxquelles a accès l'agent sont affichées en bas à gauche de l'écran. Une version graphique de ces informations est affichée en haut à gauche. La récompense obtenue au dernier pas de temps ainsi que la récompense cumulée au cours d'un épisode sont affichées en bas à droite de l'écran.
3. Il est possible d'interagir avec le système avec le panneau de droite. A chaque pas de temps, il faut choisir une action, ici la tension appliquée sur la machine à courant continu.

Questions :

1. Cliquez sur le bouton « Mission 1 ».
2. Votre but et celui de l'agent est **maximiser la somme des récompenses**. Jouez deux ou trois épisodes en essayant d'obtenir le meilleur score possible.
3. Comment avez-vous procédé pour maximiser votre score ?
4. Maximiser la récompense sur un épisode a-t-il fait émerger un comportement particulier du pendule ?
5. Après avoir cliqué sur le bouton « Mission 2 », répondre à nouveau aux questions 2. 3. et 4.
6. A votre avis, comment sont obtenues les observations sur le système réel ?

Correction :

Question 3

Le but de cette partie est d'illustrer la **prise de décision séquentielle** ainsi que le rôle de l'**observation** et de l'**action**. Les élèves doivent être capables de formuler la politique comme une **correspondance entre une observation et une action**.

la dernière action qu'il a prise, il ne peut pas en choisir une proche pour éviter d'être pénalisé.

Si les élèves ne font pas directement le lien, leur montrer un état (image + vecteur) du pendule hors contexte. Ils devraient être capables de donner l'action à effectuer pour maximiser la récompense. Si c'est le cas, ils ont appris le lien entre une observation et une action : c'est ce que doit faire la politique. La politique n'est qu'une fonction qui prend en entrée un vecteur de dimension 6 et qui sort un scalaire.

Question 4

Il faut s'assurer que les élèves aient bien compris le but de l'agent : **maximiser la récompense sur un épisode**. La récompense de la « **Mission 1** » est proportionnelle à l'énergie potentielle de pesanteur du pendule. Maximiser cette récompense a pour effet de maintenir le pendule en position haute. Le fait que le pendule soit en équilibre en position haute est une **conséquence** de la maximisation de la récompense. La « **Mission 2** » récompense la vitesse du pendule. Maximiser la récompense revient à faire le plus de tours possibles dans le temps imparti.

V.2/ Nécessité d'un modèle

Nous avons montré dans les sections précédentes comment formaliser le redressement et la stabilisation du pendule. Il ne reste qu'à entraîner l'agent. Trois possibilités s'offrent à nous :

1. Faire l'**apprentissage** puis l'**inférence** sur le système réel.
2. Faire l'**apprentissage** avec un **modèle** puis l'**inférence** sur le système réel.
3. Faire l'**apprentissage** sur un modèle simple, puis améliorer les performances en continuant d'apprendre sur le système réel. Et enfin faire l'**inférence** sur le système réel.

Questions :

1. Définir les termes **apprentissage** et **inférence**.
2. Que se passerait-il si l'agent commençait par la phase d'inférence ?
3. Quelle est l'intérêt de chacune des approches par rapport aux autres ?

Correction :

1. Un agent est **capable d'apprendre**, si la somme des récompenses qu'il perçoit augmente avec le nombre d'expériences. Pour l'apprentissage par renforcement, la base de données **se construit au fur et à mesure** grâce à l'interaction entre l'agent et l'environnement.

La phase d'**inférence** est la phase durant laquelle l'agent utilise la politique pour choisir les actions à effectuer. Les paramètres de la politique sont alors fixés. Le comportement de l'agent n'évolue plus.

2. Si l'agent commençait par l'inférence sans s'entraîner alors il ne serait pas capable de bien remplir la tâche pour laquelle il est utilisé. Dans le cas de la stabilisation du pendule, le chariot bougerait de façon

quasi aléatoire sans parvenir à maintenir le pendule en position haute.

3. La première approche – souvent nommée 'hardware in the loop' - a l'avantage de ne pas nécessiter de modèle de l'environnement. Elle n'est cependant que rarement privilégiée. D'une part, l'agent a souvent besoin de milliers d'interactions avant d'avoir des performances satisfaisantes. C'est problématique pour les applications de conduite autonome par exemple. Il n'est pas possible de laisser un agent appliquer une politique aléatoire au milieu du trafic. D'autre part, l'interaction avec le système réel est souvent beaucoup plus lente qu'avec une simulation. L'apprentissage est donc plus rapide en simulation.

La seconde approche nécessite un modèle de l'environnement. Cela suppose être capable de créer ce modèle sur la base d'une compréhension des phénomènes physiques ou de mesures effectuées sur l'environnement. De plus, un modèle est toujours imparfait, il faut donc s'assurer que les hypothèses réalisées lors de son élaboration sont raisonnables (linéarisation ou pas, prise en compte des frottements, ...). Si le modèle et l'environnement se comportent différemment, il est possible que la politique apprise sur le modèle ne fonctionne pas sur l'environnement.

La troisième approche est à privilégier car elle combine les avantages des deux précédentes. Dans un premier temps, l'agent apprend rapidement une politique convenable sur un modèle imparfait. Dans un deuxième temps, elle s'améliore en interagissant directement avec l'environnement. Comme l'agent continue d'apprendre sur l'environnement, les erreurs de modélisations sont progressivement effacées à mesure que de nouvelles expériences sont collectées.

Pour la version 2021b et les précédentes, Matlab n'offre pas la possibilité d'apprendre en interagissant avec l'environnement. Nous appliquerons donc la deuxième méthode.

VI/ Application sur le Control'X

Il ne reste qu'à appliquer les résultats des parties précédentes. L'objectif est de réaliser l'apprentissage sur un modèle du Control'X puis de déployer la politique pour vérifier que l'agent est bien capable de relever et stabiliser le pendule.

Nous considérons que la modélisation du Control'X est déjà effectuée. Si ce n'est pas le cas, se référer à la documentation fournie avec le système.

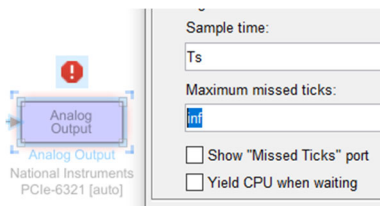
VI.1/ Installation

Les fichiers Matlab sont disponibles sur Gitlab⁴ - version 2021b et plus. Bugs possibles :

- Les blocs faisant l'interface entre le PC et le Control'X peuvent ne pas fonctionner

⁴ https://gitlab.com/Guenole.cherot/controlx_rl

correctement : régler « maximum missed ticks : inf »



- Pour tout autre problème lié à des bibliothèques, se reporter à la documentation du Control'X.

VI.2/ Entraînement

ATTENTION

1. Le temps de réponse des boutons est parfois long (Ne pas cliquer plusieurs fois)
2. Il arrive que Simulink se bloque lors de l'apprentissage (il se lance mais rien ne se passe). Dans ce cas redémarrer Matlab (*Ctrl+Alt+Suppr* puis *Fin de tâche*).

1. Ouvrir le projet « Control'X » dans Matlab
2. Ouvrir le fichier Simulink « model_simulink.slx », il doit correspondre avec la *figure 8*.
3. Initialiser les constantes du modèle avec le bouton **Position haute**. Le pendule commencera chaque simulation avec $\psi = 0$.
4. Tester les performances de l'agent par renforcement non entraîné en cliquant sur **Inférer sur le modèle**. Une fenêtre illustre le comportement simulé du pendule contrôlé par l'agent.
5. Choisir un type de récompense : Pour la stabilisation, les trois types de récompenses fonctionnent.
6. Choisir un type de pénalisation : pénaliser les variations de consigne trop brusques permet un contrôle plus en douceur mais l'apprentissage est plus lent.
7. Lancer l'apprentissage en cliquant sur **Entraîner**. La fenêtre « Reinforcement Learning Episode Manager » apparaît. La courbe bleue représente la récompense cumulée à chaque épisode. Celle orange représente la récompense que l'agent espère obtenir.
8. Une fois l'entraînement terminé, **Inférer sur le modèle**. L'agent doit être capable de maintenir le pendule en position haute sur la simulation.
9. Vérifier que le Control'X est branché et qu'aucun objet n'obstrue le passage du chariot puis **Inférer sur le système**. Le pendule doit être stabilisé sur le Control'X.

Remarque : Dans la version actuelle, seule la stabilisation du pendule fonctionne. L'agent ne parvient

pas à apprendre la phase de redressement. Une illustration du redressement est disponible (uniquement en simulation) sur python et Google Colab⁵.

VII/ Conclusion

Pour conclure, réaliser une courte présentation de 10 minutes en vous aidant des questions suivantes :

Questions :

1. Dans quel contexte est-il possible d'utiliser l'apprentissage par renforcement ?
2. A quoi sert la récompense ?
3. Qu'est-ce que la politique ?
4. Le signal récompense est-il toujours utile lors de la phase d'inférence ?
5. Quels sont les intérêts de l'apprentissage par renforcement par rapport à l'approche classique de contrôle-commande ?
6. Quels sont les inconvénients ?

Correction :

1. Prise de décision séquentielle
2. La récompense est l'équivalent de la commande. L'agent cherche à la maximiser. Le comportement global de l'agent est émergent : les stratégies gagnantes aux échecs sont inconnues, cependant il est facile de récompenser un agent qui gagne une partie.
3. La politique est la stratégie de contrôle de l'agent. C'est une correspondance entre l'observation de l'environnement et l'action à prendre. C'est ce que l'on cherche à apprendre durant l'entraînement.
4. Le signal récompense ne sert que lors de la phase d'apprentissage. Lors de la phase d'inférence, l'agent applique la politique qu'il a appris sans la modifier.
5. L'apprentissage par renforcement permet de traiter des problèmes qu'il serait difficile de formaliser autrement (ex : jeu d'échec). Il permet de se passer de modèle de l'environnement : il est possible d'apprendre directement sans connaissance a priori.
6. Les inconvénients de l'apprentissage par renforcement sont les suivants : Il faut pouvoir interagir avec le système et collecter des données. L'apprentissage est souvent long et parfois instable. Les algorithmes d'apprentissages sont complexes et pour l'instant hors de portée du grand public.

VIII/ Bibliographie

- [1] S. Sutton, A. Barto, « Reinforcement learning: An introduction », MIT press, 2018.
- [2] S. Levine, A. Kumar, G. Tucker, et al. « Offline reinforcement learning: Tutorial, review, and perspectives on open problems », arXiv preprint arXiv:2005.01643, 2020.

⁵https://colab.research.google.com/drive/1AZh_FSSImncpOuy_vdGbewK6OyqksAme?usp=sharing

- [3] Matlab, « Reinforcement Learning avec MATLAB et Simulink », <https://fr.mathworks.com/campaigns/offers/reinforcement-learning-with-matlab-ebook.html>
- [4] D. Silver, « RL Course by David Silver », Youtube, 2015
- [5] D. Silver, S. Singh, D. Precup, et al, « Reward is enough », Artificial Intelligence, 2021, vol. 299, p. 103535.
- [6] T. Haarnoja, A. Zhou, P. Abbeel, et al. « Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor », International conference on machine learning. PMLR, 2018. p. 1861-1870.

IX/ Annexe

	Compétences générales	Compétences développées	Connaissances associées
Analyser	A3 - Analyser l'organisation fonctionnelle et structurelle	Analyser les principes d'intelligence artificielle	- Phases d'apprentissage et d'inférence. - Réseaux de neurones (couches d'entrée, cachées et de sortie, neurones, biais, poids et fonction d'activation).
		Identifier la structure d'un système asservi.	- Grandeurs d'entrée et de sortie. - Capteurs
Modéliser	B2 – Proposer un modèle de connaissance et de comportement	Choisir un modèle adapté aux performances à prévoir ou à évaluer	Domaine de validité.
	B3 – Valider un modèle	Vérifier la cohérence du modèle choisi en confrontant les résultats analytiques et/ou numériques aux résultats expérimentaux.	Critères de performances
Résoudre	C3 – Mettre en œuvre une démarche de résolution numérique	Mener une simulation numérique	Influence des paramètres du modèle sur les performances.
Expérimenter	D1 – Mettre en œuvre un système	Mettre en œuvre un système en suivant un protocole	

Figure 5 : Compétences et connaissances du référentiel MPSI/PCSI/PC/PSI/MP

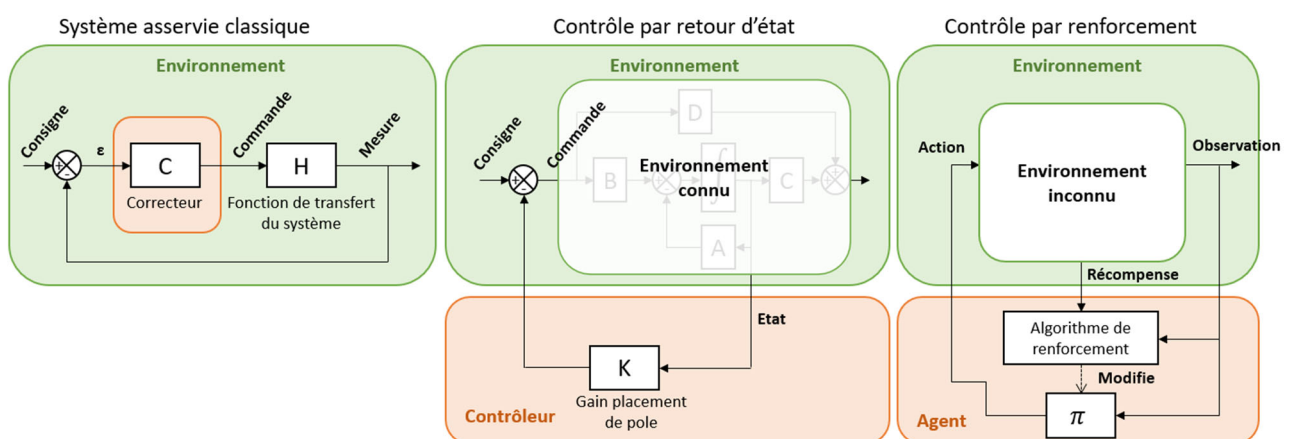


Figure 6 : Comparaison d'un contrôle par retour d'état et à l'aide d'un apprentissage par renforcement

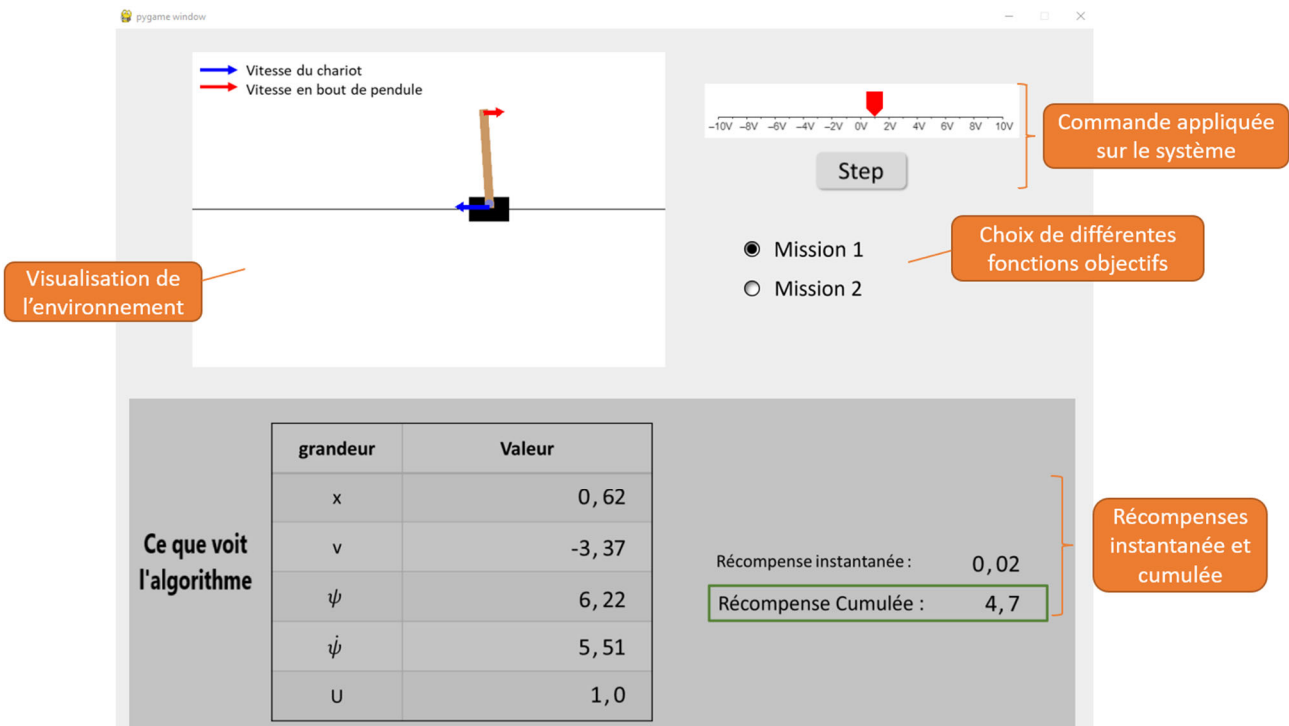


Figure 7 : Se mettre à la place de l'agent par renforcement

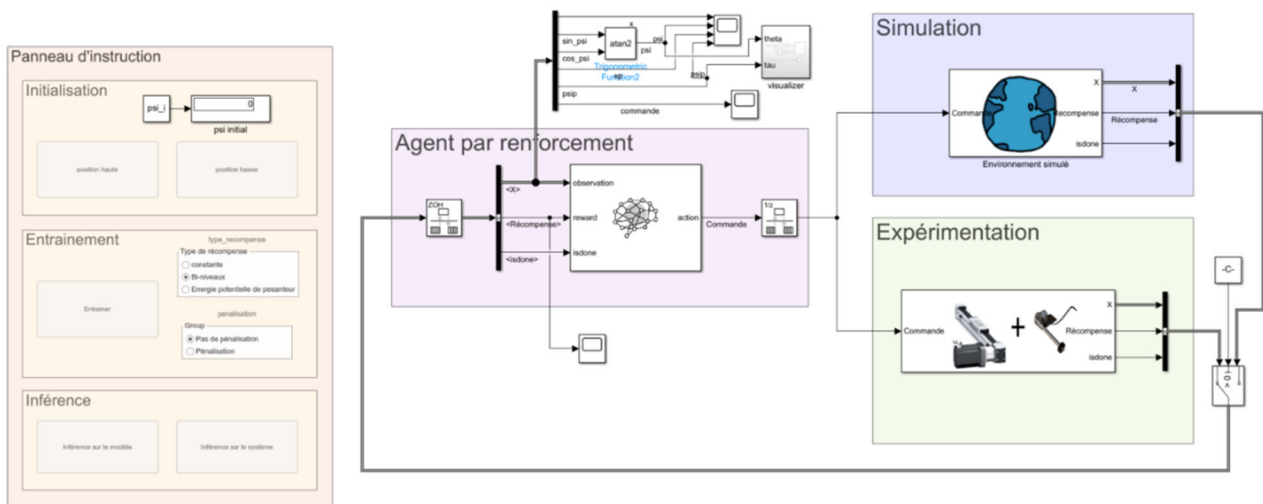


Figure 8 : Fichier Simulink

SUIVI CELLULAIRE PAR TRAITEMENT D'IMAGES ET APPRENTISSAGE

MÉLANIE PIETRI

melanie.pietri@ens-paris-saclay.fr

Résumé : L'étude de la migration de cellules cancéreuses s'est développée ces dernières années avec les progrès en intelligence artificielle pour obtenir plus d'informations sur la formation de métastases (tumeurs cancéreuses). Les expériences mises en œuvre pour effectuer des observations nécessitent de recourir à des algorithmes de traitement d'images pour en tirer des conclusions. La variabilité de la géométrie des cellules étudiées est une difficulté pour le suivi cellulaire qui peut être abordée à l'aide d'algorithmes de traitement d'images s'appuyant sur des réseaux de neurones convolutifs à segmentation d'instance. La structure complexe du réseau de neurones Mask R-CNN permet d'obtenir des résultats avec une meilleure exactitude que pour d'autres structures et un grand gain de temps sur le traitement des données.

I/ Contexte d'étude

I.1/ Étude de la migration de cellules cancéreuses

Le phénomène d'électrotaxie a été découvert il y a une centaine d'années : lorsque des cellules biologiques vivantes sont placées dans un champ électrostatique, celles-ci ont tendance à migrer en s'alignant avec la direction du champ. Depuis ces dix dernières années, ce phénomène a été une source d'intérêt pour la communauté scientifique et en particulier dans le domaine biomédical en cancérologie. En effet, la migration cellulaire n'est pas la même pour des cellules saines et des cellules cancéreuses qui vont chercher à se regrouper et créer des métastases (amas de cellules ou tumeurs).

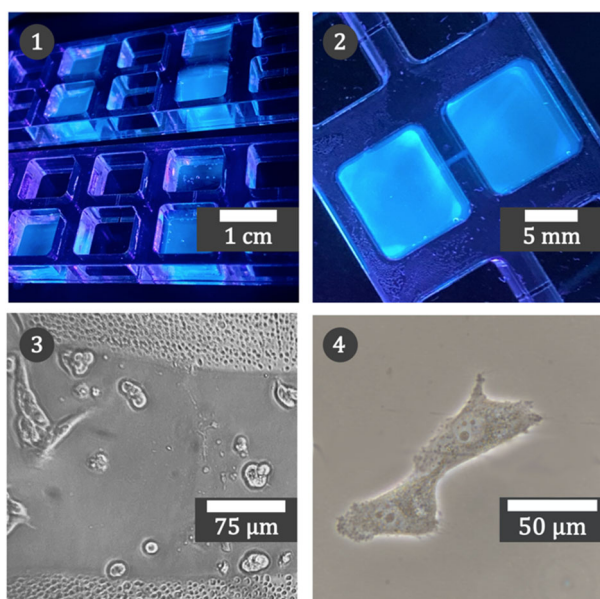


Figure 1 : 1) Dispositif microfluidique, les réservoirs cubiques sont reliés par un microcanal. 2) Zoom sur un microcanal reliant deux réservoirs. 3) Zoom sur le microcanal contenant les cellules. 4) Image de deux cellules cancéreuses.

Les données collectées sur ce type de comportements révèlent qu'il existe un lien entre le taux d'invasivité des cellules cancéreuses (leur capacité à se regrouper pour former des tumeurs) et leur capacité électrotactique (leur faculté à se déplacer en présence d'un champ électrostatique extérieur). C'est ce lien qui est étudié pour tenter de contrôler le mouvement des cellules cancéreuses sous champs électrique et réaliser par exemple de la chimiothérapie localisée.

Afin d'étudier ce phénomène, des microsystemes s'appuyant sur des dispositifs microfluidiques sont développés [1] pour observer les cellules dans des canaux d'épaisseur d'une centaine de micromètres (voir **Figure 1**). Les microscopes utilisés pour visualiser les cellules sont équipés de caméras afin d'enregistrer des vidéos des cellules se déplaçant dans le microcanal. On utilise un type de microscopie particulière : la microscopie à contraste de phase, pour avoir un meilleur contraste entre les cellules et le substrat en arrière-plan.

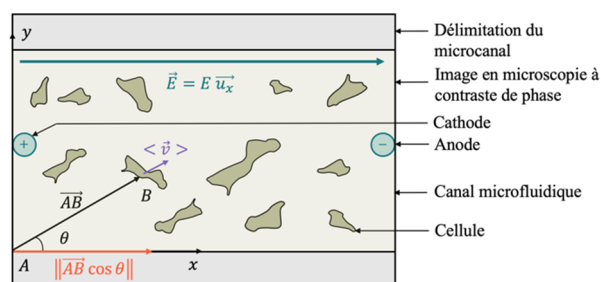


Figure 2 : Schéma des grandeurs d'intérêt pour l'étude de l'électrotaxie

Le traitement de données nécessite alors d'avoir recours à des algorithmes de traitement d'images. Les grandeurs d'intérêt pour étudier la migration cellulaire sont la vitesse moyenne de chaque cellule et la projection de la position de chaque cellule par rapport à l'horizontal. Ces paramètres sont décrits sur la **Figure 2**. Ces grandeurs sont mesurées sur toutes les cellules de l'image pour chaque image d'une vidéo. On étudie ensuite la relation entre la norme du champ électrique et ces deux paramètres.

1.2/ Traitement des données : intérêt du *deep learning*

Comme on peut le voir sur la **Figure 1**, les cellules cancéreuses qui adhèrent au microcanal ne sont pas de formes circulaires. Il est donc assez difficile de suivre ces cellules avec des algorithmes du type transformée de Hough (détection de cercles) [2]. D'autres algorithmes comme celui de Crocker-Grier [3] s'appuyant sur la physique du mouvement brownien permettent d'effectuer du suivi cellulaire plus précis mais les erreurs demeurent encore trop grandes pour des cellules avec ce type de géométrie. Ceux-ci sont mis en œuvre sur la **Figure 3** pour des cellules circulaires.

Une autre difficulté s'ajoute à la complexité de la forme géométrique des cellules : celles-ci changent de forme d'une image à une autre en se déplaçant sur le substrat du canal.

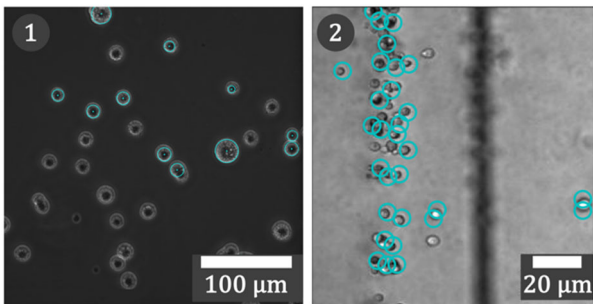


Figure 3 : 1) Détection de cellules en suspension par transformée de Hough. 2) Détection de cellules en suspension par l'algorithme de Crocker-Grier

De plus il faut un certain nombre d'images pour pouvoir obtenir des données exploitables, ce qui est un point à prendre en compte concernant la complexité en temps et en mémoire des algorithmes utilisés.

Face à ces difficultés, des approches utilisant des réseaux de neurones convolutifs avec apprentissage supervisé et des modèles pré-entraînés ont été développés. Nous allons par la suite présenter une structure utilisée pour traiter les images et mesurer les grandeurs d'intérêt.

II/ Réseau de neurones convolutif et segmentation d'instance

II.1/ Présentation générale

L'objectif derrière l'utilisation de la structure de réseau de neurones est d'effectuer de la segmentation d'instance c'est-à-dire classer et distinguer des objets de la même instance « cellule » afin de suivre leur mouvement au cours du temps sur les différentes images d'une vidéo.

L'algorithme Usiigaci utilisé [4] fait appel à trois modules distincts : un réseau de neurones Mask R-CNN (*Mask Regional Convolutional Neural Network*) pour la détection des cellules et la segmentation, le module Python Trackpy permettant de déterminer les trajectoires des cellules et un module de traitement de données pour automatiser le traitement et visualiser les

résultats. L'ensemble est codé en langage Python et accessible via Github [5].

Le réseau de neurones Mask R-CNN s'articule autour de deux structures combinées : un réseau à pyramide de caractéristiques (*FPN : Features Pyramid Network*) [6] et un réseau résiduel de 101 couches (Resnet-101) [7]. Nous allons ici montrer l'intérêt de cette structure assez complexe dans le suivi cellulaire.

II.2/ Données pour le pré-entraînement

Dans un premier temps il faut pré-entraîner le réseau de neurones pour la segmentation. Ici le réseau est pré-entraîné sur un problème de classification général pour ensuite en faire l'application sur un problème plus précis (*transfer learning*). Mask R-CNN a été pré-entraîné préalablement avec la base de données Microsoft COCO contenant de nombreuses images d'objets à détecter [8]. Le réseau pré-entraîné provenant de Matterport, Inc est documenté et disponible sur Github [9]. On ajoute ensuite de l'information par rapport au problème de suivi cellulaire avec notre pré-entraînement sur des images des cellules étudiées : un entraînement de 100 epochs pour le réseau de neurones à pyramide de caractéristiques et 500 epochs sur tout le réseau ResNet-101. Le principe de génération des données d'entraînement est décrit sur la **Figure 4**.

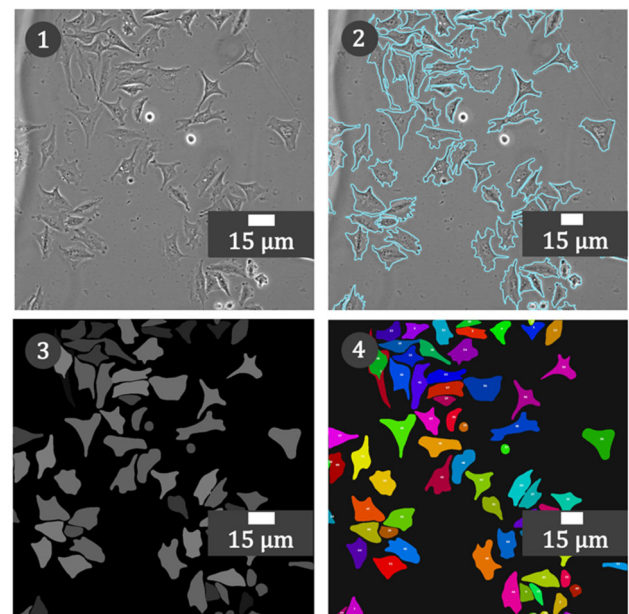


Figure 4 : Différentes étapes de la création des données d'entraînement avec la base de données de [5]. 1) Image en microscopie à contraste de phase. 2) Segmentation manuelle des cellules. 3) Création de masques de segmentation. 4) Numérotation des masques de segmentation.

Un jeu de 50 images des cellules est utilisé. On segmente à la main les cellules en effectuant un détourage puis on génère des masques de segmentation à l'aide du logiciel ImageJ. On peut alors identifier de manière unique (par la couleur ou la numérotation) chaque cellule. 45 images de ce jeu de données sont utilisées pour l'entraînement et les 5 restantes permettent d'effectuer des tests de validation.

II.3/ Structure du réseau de neurone

La **Figure 5** illustre la structure globale du réseau de neurones utilisé [10]. On peut dans un premier temps s'intéresser à la structure résiduelle. Il est notable que le nombre de couches utilisées ici (101) est assez élevé. Ceci permet d'avoir une meilleure précision sur les résultats obtenus.

Néanmoins cela mène à une structure de réseau complexe en couches qui, si elles sont empilées une à une successivement, peut conduire à une dégradation de la précision sur les résultats par rapport à des structures moins profondes.

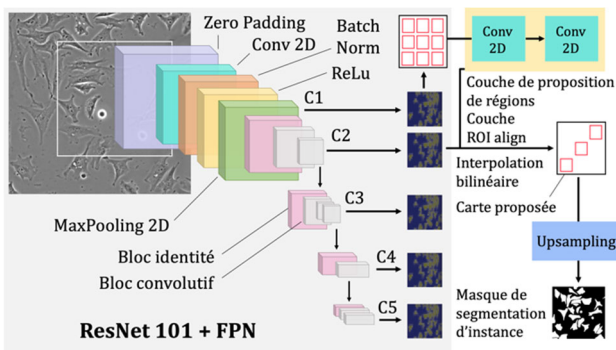


Figure 5 : Structure de Mask R-CNN, schéma adapté de [10]

Un problème connu pour les réseaux avec de nombreuses couches est la disparition du gradient (*gradient vanishing*) : lors de la rétropropagation du gradient de couches en couches ses valeurs diminuent rapidement ce qui entraîne son annulation totale et l'arrêt de l'apprentissage [11].

Pour pallier ce problème, les réseaux de neurones résiduels autorisent des sauts de connexions entre les couches. Avec cette structure l'apprentissage est plus rapide et moins vulnérable aux perturbations. Le réseau de neurones résiduel d'Usiigaci comprend deux types de blocs : des blocs identités et des blocs convolutifs décrits sur la **Figure 6**.

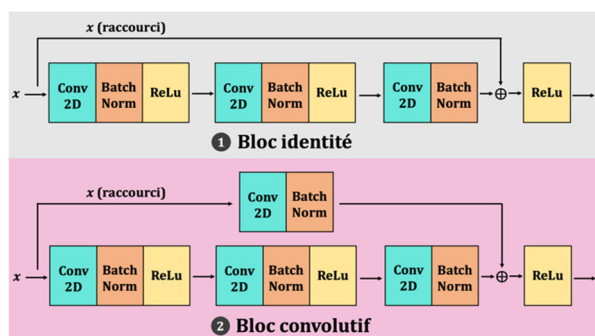


Figure 6 : Types de blocs du réseau de neurones résiduel

Le bloc identité est utilisé lorsque les dimensions de l'entrée et de la sortie sont identiques tandis que le bloc convolutif comprend une couche convolutive dans le chemin raccourci pour assurer une dimension égale entre l'entrée et la sortie. Ces deux types de blocs comprennent des fonctions identiques. Conv

2D effectue une convolution 2D. La fonction *Batch Normalization* rend le réseau plus rapide et plus stable en ajoutant des couches supplémentaires. Enfin la fonction ReLu (*Rectifier Linear Unit*) est une fonction d'activation qui a pour but de « casser » la linéarité de l'image après les opérations linéaires de convolution. En effaçant certaines valeurs de l'image, les calculs sont accélérés et les caractéristiques mises en évidence par la convolution sont d'avantage surlignées.

La structure FPN combinée à la structure résiduelle permet la détection de cellules à plusieurs échelles. Cette structure fonctionne de telle sorte qu'à partir d'une image unique d'une taille donnée, des cartes de caractéristiques de tailles proportionnelles à plusieurs niveaux sont générées de manière convolutive [12]. Les pyramides ainsi créées sont alors invariantes avec l'échelle : lors d'une réduction ou d'une augmentation de la taille d'une cellule, le changement d'échelle est compensé par le déplacement dans la pyramide. Cette structure permet donc de détecter des cellules sur une gamme d'échelles plus étendue. Cette structure est schématisée sur la **Figure 5** avec la succession de blocs roses et gris.

En sortie du réseau, une couche de propositions de régions génère des cadres qui délimitent les cellules potentiellement détectées. La couche *ROI (Region Of Interest) align* [13] permet ensuite d'aligner correctement les caractéristiques extraites des diverses opérations du réseau avec l'image d'entrée. Pour effectuer cela, une interpolation bilinéaire est utilisée, elle correspond à un redimensionnement de l'image [14]. La fonction *Upsampling* (suréchantillonnage) ramène enfin la résolution à celle de la couche précédente. On obtient finalement le masque de segmentation d'instance de la **Figure 5**.

III/ Détermination des paramètres d'intérêt

Une fois les masques de segmentation obtenus pour chaque image de la vidéo de la migration cellulaire, le module TrackPy calcule les paramètres d'intérêt décrits précédemment. Cela passe par l'utilisation d'un algorithme de recherche du plus proche voisin avec un arbre k-dimensionnel [15] qui ne sera pas détaillé ici.

On peut ainsi visualiser par exemple la trajectoire de chaque cellule de même que sur la **Figure 7**. Le module de traitement de données permet ensuite d'obtenir les graphes de la direction et la vitesse des cellules en fonction du champ électrique appliqué.

IV/ Discussion

IV.1/ Mise en œuvre du réseau de neurones

L'utilisation d'Usiigaci requiert l'accès à une carte graphique (GPU) pour effectuer des calculs parallélisés. Les différentes bibliothèques Python doivent être configurées en accord avec la configuration proposée dans [5] afin d'éviter les conflits entre bibliothèques et empêcher le fonctionnement de l'algorithme.

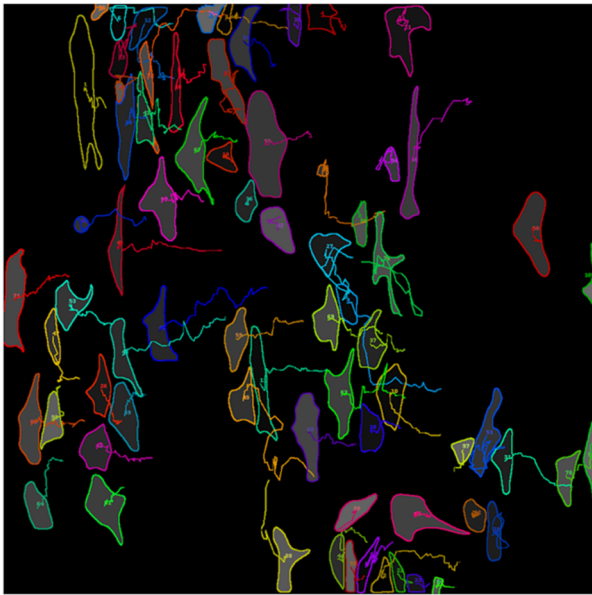


Figure 7 : Masques de segmentation et trajectoires de chaque cellule après traitement sous Trackpy, image issue de [5]

L'avantage d'Usiigaci est qu'il est basé sur un réseau Mask R-CNN pré-entraîné sur une base de données assez conséquente. De ce fait, l'entraînement requiert un jeu de données moins important que d'autres algorithmes comme Deepcell et cet entraînement prend moins de temps. Les *epochs* sont complétés plus rapidement mais un grand nombre est nécessaire pour obtenir une bonne précision. Des données issues de [10] concernant l'entraînement sont visibles sur la Figure 8.

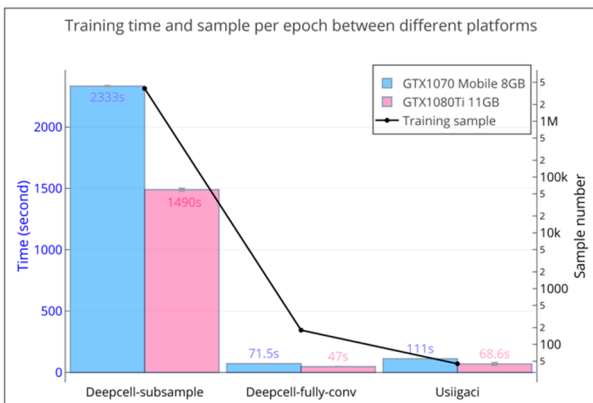


Figure 8 : Comparaison du temps d'entraînement et du nombre d'échantillons par epoch pour Usiigaci et Deepcell, avec deux modèles de carte graphique, graphe issu de [10]

En comparaison avec d'autres algorithmes, ce que nous verrons dans la partie suivante, la quantité de données utilisée (50 images) pour l'entraînement et les tests de validation est suffisante pour obtenir des résultats.

IV.2/ Comparaison avec d'autres méthodes

On peut maintenant comparer l'algorithme Usiigaci à d'autres structures développées pour traiter ce type de problèmes. Pour effectuer cette comparaison plusieurs indicateurs ont été définis. L'un d'entre eux est l'indice de Jaccard qui permet d'évaluer la similarité entre l'ensemble d'images de départ et celles en sortie de l'algorithme. Cet indice est évalué dans [4] et représenté sur la Figure 9 pour différents algorithmes de segmentation.

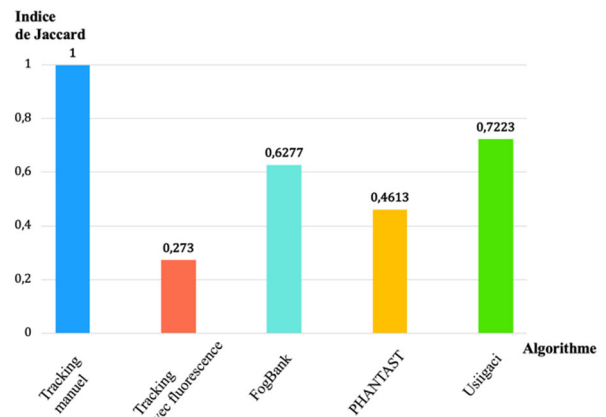


Figure 9 : Comparaison de l'indice de Jaccard pour différents algorithmes de segmentation, données issues de [4]

À première vue, une méthode qui semble simple à mettre en œuvre est le suivi manuel des cellules à l'aide d'un logiciel de *tracking* du type Tracker. Les résultats obtenus sont très satisfaisant du point de vue de l'indicateur considéré (indice de Jaccard de 1) mais le coût en temps peut être assez conséquent lorsqu'il faut traiter un grand nombre de vidéos des cellules afin que ces résultats soient représentatifs (environ 800 cellules pour 480 images). Usiigaci constitue donc un gain de temps considérable pour traiter les données comparé à cette méthode puisque la segmentation à la main n'est faite qu'une seule fois pour générer les données d'entraînement.

D'autres algorithmes de segmentation adaptés à la problématique de suivi cellulaire comme Fogbank [16] et PHANTAST [17] permettent d'obtenir des résultats en moins de temps que par du suivi manuel mais sont moins adaptés par rapport à Usiigaci pour le suivi cellule par cellule lorsque la densité cellulaire est élevée sur l'image. L'approche d'Usiigaci permet de traiter les données sans avoir recours à un traitement biologique particulier pour les cellules. Elles sont donc observées dans des conditions « naturelles ». D'autres méthodes de traitement d'images s'appuient sur l'utilisation de composés fluorescents pouvant être toxiques pour la cellule, qui ne sont pas requis ici pour obtenir des résultats satisfaisants.

Néanmoins comme beaucoup d'algorithmes de *deep learning*, si les images à traiter ont été prises dans des conditions trop éloignées de celles utilisées pour l'entraînement, des erreurs peuvent apparaître dans la segmentation et se répercuter dans le suivi cellulaire.

V/ Perspectives

Des perspectives qui pourraient être envisagées sont l'amélioration du dispositif expérimental (confection du système microfluidique, technologie d'électrodes utilisées ...) en le rendant le plus simple et portable possible, ainsi que la simplification de l'utilisation d'Usiigaci. L'idée serait d'automatiser les mesures afin que ce type de microsystèmes combiné à Usiigaci puissent être utilisés par des professionnels du milieu médical pour effectuer des diagnostics sur des patients de manière rapide et précise.

L'algorithme Usiigaci a été utilisé dans le cadre d'un stage de recherche en collaboration avec le LuMIn de l'ENS Paris-Saclay, l'Université de Freiburg en Allemagne et l'Université de Luleå en Suède.

VI/ Bibliographie

- [1] J. Leal, N. Jedrusik, S. Shaner, C. Boehler, M. Asplund « SIROF stabilized PEDOT/PSS allows biocompatible and reversible direct current stimulation capable of driving electrotaxis in cells » *Biomaterials* 2021, Volume 275, August 2021, DOI: 10.1016/j.biomaterials.2021.120949
- [2] A. Shehata, S. Mohammad, M. Sameer and M. Ehab Ragab « A Survey on Hough Transform, Theory, technique and Applications » *International Journal of Computer Science Issues*, Volume 12, Issue 1, No 2, January 2015, DOI: 10.48550/arXiv.1502.02160
- [3] J. Crocker and D. Grier “Methods of Digital Video Microscopy for Colloidal Studies” *Journal of Colloid and Interface Science* 179, 298-310, 1996, DOI: 10.1006/jcis.1996.0217
- [4] H-F. Tsai, J. Gadja, T. Sloan, A. Rares, A. Shen « Usiigaci: Instance-aware cell-tracking in stain-free phase contrast microscopy enabled by machine learning » *Software X*, Volume 9, 230-237 January-June 2019, DOI: 10.1016/j.softx.2019.02.007
- [5] Usiigaci – Github: <https://github.com/oist/Usiigaci>
- [6] J. Hui “Understanding Feature Pyramid Networks for object detection (FPN)” , Medium, 2018: <https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c>
- [7] K. He, X. Zhang, S. Ren, J. Sun “Deep Residual Learning for Image Recognition” , *CVPR*, 2016, DOI: 10.48550/arXiv.1512.03385
- [8] Microsoft, COCO Common Objects in Context : <https://cocodataset.org/#home>
- [9] Mask R-CNN for Object Detection and Segmentation – Matterport, Inc: https://github.com/matterport/Mask_RCNN
- [10] H. Tsai, J. Gadja, T. Sloan, A. Rares, A. Shen « Usiigaci: Instance-aware cell-tracking in stain-free phase contrast microscopy enabled by machine learning – Supplementary information » *Software X*, Volume 9, 230-237 January-June 2019, DOI: 10.1016/j.softx.2019.02.007
- [11] C. Wang “The Vanishing Gradient Problem”, Medium, 2019: <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>
- [12] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie “Feature Pyramid Networks for Object Detection”, *CVPR*, 2017, DOI: 10.48550/arXiv.1612.03144
- [13] J. Chao “Understanding ROI Align”, SuperComputer’s Blog, July 2018 : <https://chao-ji.github.io/jekyll/update/2018/07/20/ROIAlign.html>
- [14] J. Chao “Understanding Bilinear Image Resizing”, SuperComputer’s Blog, July 2018 : <https://chao-ji.github.io/jekyll/update/2018/07/19/BilinearResize.html>
- [15] « Arbres kd », Wikipédia, Juin 2019 : https://fr.wikipedia.org/wiki/Arbre_kd
- [16] J. Chalfoun, M. Majurski, A. Dima, C. Stuelten, A. Peskin and M. Brady “FogBank: a single cell segmentation across multiple cell lines and image modalities”, *BMC Bioinformatics*, 2014, DOI:10.1186/s12859-014-0431-x
- [17] N. Jaccard, L.Griffin, A. Keser, R. Macown, A. Super, F. Veraitch and N. Szita “Automated Method for the Rapid and Precise Estimation of Adherent Cell Culture Characteristics from Phase Contrast Microcopy Images”, *Biotechnology and Bioengineering*, Vol. 111, No. 3, Mars 2014, DOI: 10.1002/bit.25115

L'INTELLIGENCE ARTIFICIELLE AU SERVICE DE LA MOBILITÉ URBAINE À RENNES PAR LACROIX GROUP

REYNHOLDS REINETTE

Smart Data Services Leader chez Lacroix, Rennes, r.reinette@lacroix-lab.com

Résumé : Cet article présente les justifications et les grandes lignes de la mise en œuvre des réseaux de neurones pour de la classification sur des images de carrefours routiers afin d'optimiser le contrôle des feux tricolores et d'augmenter la connaissance de la circulation de la métropole rennaise.

Le Lab d'innovation du groupe LACROIX a pour mission de soutenir l'innovation au sein du groupe LACROIX, équipementier technologique et fournisseur de solutions IoT dont les activités autour de la mobilité, de l'industrie et de l'environnement servent un monde connecté et responsable.

En partenariat avec Rennes Métropole, le Groupe LACROIX expérimente une solution de gestion intelligente du trafic : adapter l'activité des feux de circulation (plans de feux) afin d'améliorer la fluidité du trafic grâce à de l'intelligence artificielle locale.

L'objectif ? Répondre aux enjeux sociétaux liés à la transition écologique par l'optimisation des mobilités douces, l'amélioration de la sécurité routière et la décongestion des centres urbains.

Cette solution se compose de technologies exploitées par LACROIX : la détection et la classification des objets d'une part et l'exploitation de ces données d'autre part au niveau des capteurs placés sur le carrefour. Concrètement, il s'agit d'équiper une caméra, d'intelligence artificielle embarquée pour générer des informations contextuelles (nombre, type d'usagers, position sur le carrefour). Grâce à ces équipements locaux, il est alors possible d'extraire et de calculer tous

les usages du carrefour de manière inclusive et notamment les piétons, cyclistes, ... Ainsi, nous donnons la possibilité à la collectivité de pouvoir prendre en compte tous les usagers et d'appliquer des scénarios d'optimisation de la mobilité de façon automatique.

De plus cette expérimentation permettra de générer des statistiques pour Rennes Métropole, permettant d'enrichir leurs décisions grâce à des données factuelles et contextualisées, par exemple pour investir dans une piste cyclable ou même renforcer la sécurité des mobilités douces.

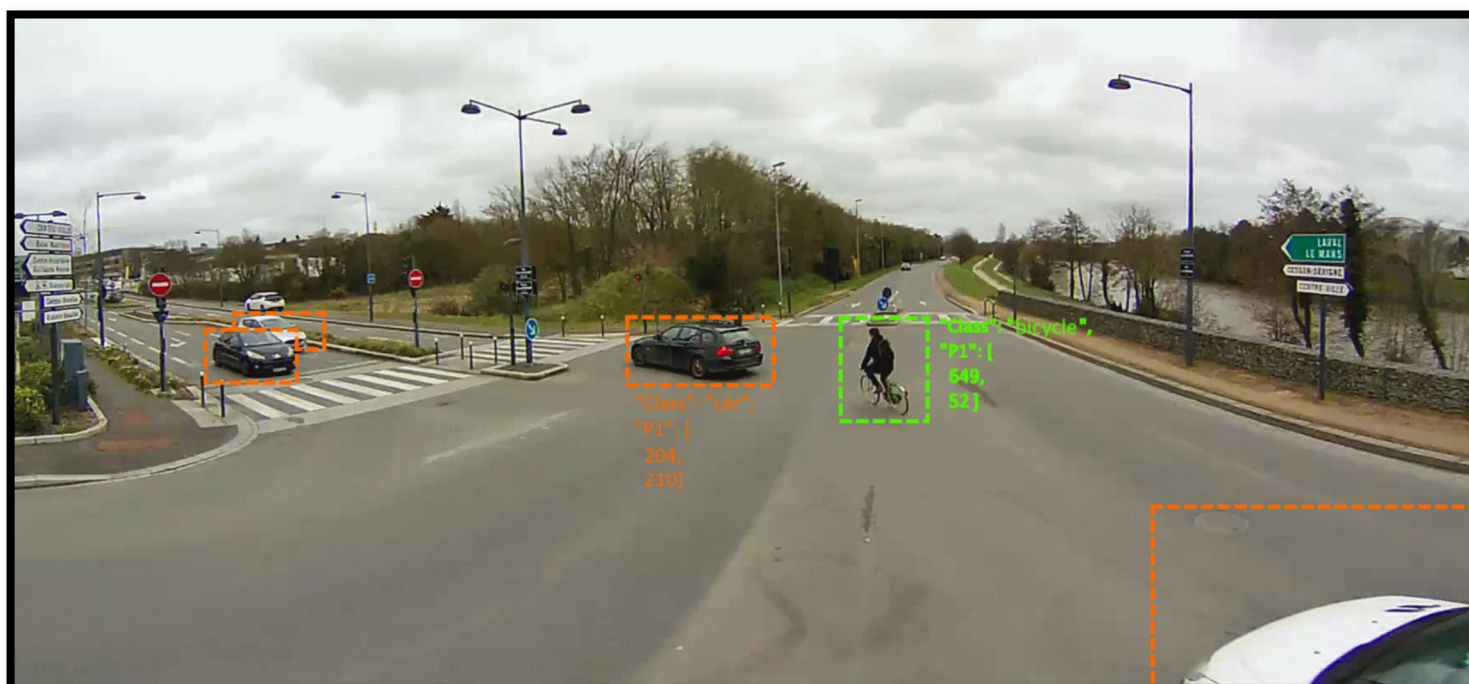


Figure 1: Vue « contextualisée » du carrefour

L'intégration de toutes ces briques logicielle et matérielle, permet de créer le flux de données suivant :

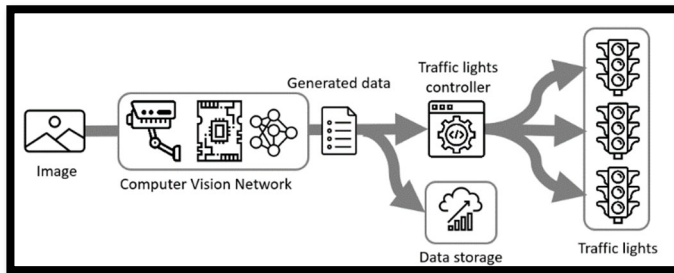


Figure 5 : Flux des données

1) L'image est transformée en données contextuelles grâce aux 2 réseaux

2) Cette donnée est ensuite enrichie par des informations métiers telle que la vitesse, la position GPS, mais surtout la position sur le carrefour

3) Les données générées sont stockées pour permettre une analyse par les services de la mobilité

4) Les données générées sont envoyées vers le contrôleur de feux afin que celui-ci puisse prendre en compte toutes les mobilités, en fonction de différents scénarios (piétons en attente, véhicules en attente, ...)

Concrètement l'utilisation de l'IA dans cette expérimentation permet d'avoir un capteur évolutif ; capable de détecter au fur et à mesure des nouveaux entrainements du réseau de nouvelles classes comme les Personnes à Mobilités Réduites, pour qui les temps de traversée pourraient être aménagés ou encore les trottinettes, afin d'élaborer au mieux des scénarios de mobilité inclusifs et innovants.

Sources :

https://www.villeintelligente-mag.fr/LACROIX-experimente-sa-solution-Smart-Crosswalk-sur-le-territoire-de-Rennes-Metropole_a1198.html

<https://metropole.rennes.fr/une-experimentation-inout-pour-encourager-les-mobilites-actives>

<https://cocodataset.org/#home>

LE SOLEIL SE LÈVE À L'OUEST !

ARNAUD BRUGIER¹

¹Enseignant de SII, département GIM de l'IUT de Saint-Denis, place du 8 mai 1945, 93200 St-Denis, USPN (<https://iutsd.univ-paris13.fr/departement/genie-industriel-maintenance/>), brugier@sorbonne-paris-nord.fr

Résumé : de nombreux projets photovoltaïques voient le jour tout au bout du monde (« Penn ar Bed » en breton), à l'ouest du Finistère, dans la Presqu'île de Crozon.

Ce mode de fourniture d'énergie électrique issu d'une ressource renouvelable est respectueux de l'environnement et permet de rapprocher la production des lieux de consommation. C'est un enjeu important pour la région Bretagne et en particulier ici, à son extrémité, où seulement 8% de l'énergie consommée est produite sur place. Participant ainsi à la sécurisation de l'approvisionnement énergétique, ces projets permettent également de dynamiser le territoire et est créateur d'activité. Ils illustrent aussi des politiques publiques volontaristes et pragmatiques de réduction des consommations, de développement des ENR et d'utilisation des ressources locales sur ce territoire enclavé et ce, tout en favorisant l'implication citoyenne et les échanges : sensibilisation, concertation et événements avec les habitants.e.s, financement, démonstration de solutions...

I/ Contexte du territoire

La Communauté de Communes Presqu'île de Crozon – Aulne Maritime (CCPCAM²) est engagée depuis plusieurs années dans une démarche de développement durable s'appuyant sur une politique QSSEE (Qualité, Sécurité, Social, Environnement, Energie) reconnue par un organisme de certification et visant notamment à contribuer à la prise en compte des défis écologiques, énergétiques et climatiques et notamment réduire ses émissions de gaz à effet de serre et diminuer son impact sur le réchauffement climatique.

Elle a déjà mené à bien plusieurs actions orientées vers la protection de l'environnement et la transition énergétique :

- l'élaboration d'un Plan Climat Air Energie Territorial (PCAET), fruit d'un travail mené depuis 2018,
- la mise en place d'un partenariat avec *Ener'gence*³, l'agence de l'énergie et du climat du Pays de Brest, pour la recherche d'économies d'énergie dans les bâtiments communautaires, la mise en place d'un guichet unique d'information et de conseils (*Tnergie*⁴), des opérations « thermofaçades », des ateliers dans le cadre du réseau des Citoyens du Climat, l'accès à un « cadastre solaire » en ligne⁵, l'édition d'un « guide des gestes écoresponsables et de gestion de l'énergie », ...
- la construction d'une usine de compostage à partir des déchets verts recueillis à la déchetterie,
- la mise en place d'une filière locale complète bois-énergie ainsi que de chaudières bois,
- la pose de modules PV sur le toit de l'Office du Tourisme de Crozon - ainsi que la mise aux normes du bâtiment (ancienne gare) en matière de réglementation thermique – et sur la toiture terrasse du nouveau centre culturel L'Améthyste,
- le développement d'un réseau électrique intelligent « smartgrid » sur les bâtiments communautaires et

la construction d'une centrale solaire photovoltaïque au sol,

- la pose, en 2021, de modules PV intégrés aux vitrages de la piscine communautaire.
- la rénovation thermique des bâtiments communautaires.
- Et réfléchit encore :
- au réaménagement des installations de la déchetterie pour améliorer le recyclage, la réparation et le réemploi des matériaux et appareils,
- au développement de son « smartgrid »,
- au développement d'installations PV en autoconsommation et à des dispositifs de récupération et de valorisation de l'énergie fatale de certains procédés (futur abattoir,...),
- ...

II/ L'exploitation du « bois énergie »

La CCPCAM développe l'utilisation du bois énergie sur son territoire et multiplie les partenariats avec l'Office National des Forêts, le Conservatoire du Littoral et les communes, notamment via la récupération de bois d'opportunité : opération d'entretien des forêts ou de talus et travaux nécessitant l'abattage d'arbres. Par ailleurs, la communauté de communes dispose depuis 6-7 ans d'une vingtaine d'hectares de saules plantés, à courte rotation. En tout c'est un potentiel bois-énergie estimé à 3600t/an qui est mobilisable alors, qu'auparavant, 3500t de bois quittaient annuellement la Presqu'île pour être valorisés ailleurs...

Une plateforme de 6000m² et un hangar de 600m² (Fig. 1) permettent désormais le stockage, le broyage, le criblage et le séchage d'environ 600t de plaquettes de bois par an qui seront utilisées dans les chaufferies communautaires : piscine (250t/an), école Jean Jaurès (150t/an), hôpital/ehpad (200t/an). Le nouveau centre culturel L'Améthyste fonctionne, lui, avec du bois sous forme de granulés et l'installation et son

approvisionnement sont gérés par un prestataire extérieur.

La CCPCAM reste ouverte au développement de nouveaux projets sur le territoire.



Figure 1 : vue de la plateforme couverte de stockage et séchage des plaquettes de bois (ZA de Kerdanvez)

III/ Quelques exemples de centrales PV raccordées au réseau, en autoconsommation et en sites isolés

L'énergie solaire est de plus en plus exploitée sur la Presqu'île au travers d'installations photovoltaïques de typologies et de tailles très différentes. Nous pouvons en citer quelques-unes et en détaillerons deux en particulier dans des chapitres ultérieurs :

- une première installation communautaire ($S=32\text{m}^2$ / $P_c=3,84\text{kWc}$) a été mise en place et intégrée dans la toiture en ardoise de l'ancienne gare (fig.2). Elle produit depuis 2011 près de 4 000 kWh/an.



Figure 2 : vue de la centrale PV en toiture de l'ancienne gare, aujourd'hui gare routière et office de tourisme

- le nouveau centre culturel accueille sur son toit-terrasse une centrale ($P_c=35,89\text{kWc}$) dédiée à la revente totale. Elle est la propriété de la société *Sotraval* qui l'exploite et la maintient. Un écran d'information destiné au grand public est prévu dans le hall d'accueil du lieu.
- une autre installation est sur le point d'être mise en service à la piscine communautaire. Les vitrages en fin de vie y ont été remplacés par des double vitrages photovoltaïques à base de silicium monocristallin de marque ERTEX Solar GmbH (Fig. 3 et 4). La puissance installée est de 6kWc et l'installation fonctionne en autoconsommation

totale en journée (alimentation éclairages, pompes, ...). Le coût des travaux est de 153kEuros subventionnés à 80% dans le cadre de la dotation de soutien de l'Etat à l'investissement local. L'objectif affiché sur ce projet est de communiquer sur les ENR auprès du grand public et non d'obtenir une rentabilité financière rapide.



Figure 3 : vue des vitrages avec cellules PV intégrées depuis l'intérieur de la piscine « Nautil'Ys » (déc. 2021)

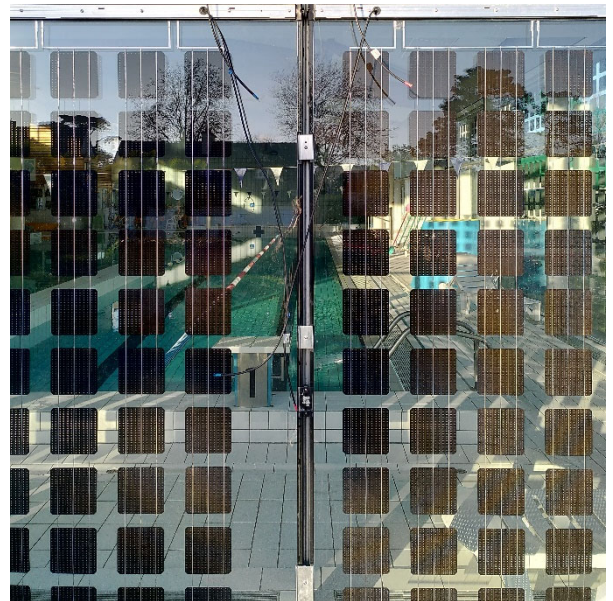


Figure 4 : finalisation de l'installation des vitrages avec cellules PV intégrées (décembre 2021)

- le centre commercial Leclerc situé à Penandreff, Crozon, a fait construire fin 2021 une structure de couverture du parking recouverte de 4800m^2 de modules PV qui devraient permettre de couvrir 25 % des besoins en énergie électrique du point de vente. Le nouveau magasin Lidl, tout proche, a été construit en intégrant 831m^2 de modules PV en toiture,
- plusieurs agriculteurs ont aussi fait le choix de cette technologie depuis quelques années en équipant leurs toitures (rénovations, stabulations, bâtiments neufs (Fig. 5)) ainsi, bien sûr, que de nombreux particuliers : vente totale au réseau et désormais autoconsommation, avec ou sans revente du surplus, sont plébiscités.



Figure 5 : les toitures des deux bâtiments (atelier et point de vente/stock/dégustation) de la cidrerie de Rozavern⁶ sont équipées de modules en surimposition.

- enfin, on peut, en se promenant, rencontrer diverses petites installations en site isolé avec stockage électrochimique : équipements de mesures, de télésurveillance et de communication de données dédiés à la protection des côtes (Fig. 6), bateaux de plaisance, balises en mer ou dispositifs de signalisation à terre, habitations temporaires ou légères, ... parfois associées à un petit aérogénérateur (système hybride).



Figure 6 : exemple d'installation PV en site isolé à la Pointe de Dinan

IV/ Le « smartgrid » Ener'Glaz

A l'occasion de la construction de nouveaux bâtiments communautaires sur la zone d'activités de Kerdanvez (extension du siège communautaire, garage-atelier et hangars de stockage), il a été décidé de lancer

des études d'opportunité et de faisabilité relatives au développement des énergies renouvelables avec, en vue, un certain degré de mixité des sources et niveau d'autosuffisance. Ces études se sont effectivement avérées concluantes.

Le projet a donc été lancé avec le soutien d'Ener'gence, partenaire de la CCPCAM sur les questions énergétiques. A l'issue d'un appel d'offres lancé en 2019, la société *Solvéo Energie*⁷ a été retenue pour être le concepteur – réalisateur du projet. Cette société toulousaine est spécialisée dans le développement, le financement, la construction, l'exploitation et la maintenance de projets innovants permettant de répondre aux besoins électriques par des moyens de production mettant en œuvre des ressources renouvelables locales ainsi qu'à développer l'autoconsommation et l'autoproduction. Ici, le principe d'un « smartgrid » permet de consommer un maximum d'énergie produite localement grâce au stockage et à un système de pilotage intelligent a été proposé.

L'installation réalisée est ainsi constituée de plusieurs champs photovoltaïques cumulant une puissance installée de 60 kWc (modules PV de technologie polycristalline), d'une éolienne de 25 kW (24,5 m de haut en bout de pale / 18 m de hauteur de moyeu (hauteurs limitées à cause de la proximité d'installations militaires) / S=133m²), d'un dispositif de stockage dans des batteries lithium d'une capacité de 134 kWh, d'un accès au réseau de distribution, d'un système de mesure avec plusieurs points de comptage et d'un automate de gestion.

Selon les besoins, l'automate de gestion orientera la production d'énergie renouvelable en priorité vers les équipements communautaires qui sont raccordés au réseau intelligent : bâtiments (siège, atelier-garage, usine de compostage, hangar de tri déchets) et véhicules électriques (5 Renault Zoé, 1 Renault Partner, 1 Opel / alimentation unidirectionnelle via les bornes de recharge gérées par la CCPCAM), puis vers les batteries de stockage. Le reste sera injecté sur le réseau public de distribution d'électricité (Enedis) sous forme de don au fonds de dotation « énergie solidaire » via la coopérative nationale Enercoop⁸. Enedis a été un partenaire important dans ce projet tant au niveau de l'étude du raccordement au point de livraison de la CCPCAM que pour évaluer son impact sur la consommation d'électricité dans cette extrémité du réseau.

Conformément à la loi PACTE relative à la croissance et à la transformation des entreprises, l'installation pourrait être autorisée à fournir de l'énergie à un périmètre « d'autoconsommation collective » plus large (dans un rayon de 2km) que celui des seuls bâtiments communautaires. La CCPCAM étudie donc une éventuelle extension de ce réseau d'énergie renouvelable à l'ensemble de la ZA de Kerdanvez. Le cabinet d'étude *Cohérence Energies*⁹ a été mandaté pour examiner l'opportunité puis la faisabilité de cette extension. Toutes les entreprises intéressées, dont la principale entreprise de production industrielle civile de la Presqu'île, pourront bénéficier d'un diagnostic énergétique de leurs bâtiments, financé entièrement dans le cadre de l'étude. Celle-ci devra

explorer les structures juridiques et les solutions techniques à privilégier dans le cadre de cette expérimentation pour pouvoir considérer l'implication de chaque acteur (public, privé) au « smartgrid » et prendra également en compte l'évolution du parc bâti sur la ZA en promouvant de futures toitures orientées au sud et dimensionnées pour accueillir des modules PV.

IV.1/Quelques chiffres¹⁰

Selon les prévisions (IMEX/Solvéo), 101 500 kWh/an seront directement consommés par l'ensemble des équipements concernés (siège, atelier-garage, hangar de tri déchets et plateforme bois-énergie, usine de compostage ainsi que véhicules électriques). Le reste de la production, soit 37 525 kWh/an, sera injecté sur le réseau Enedis (via une convention d'injection réseau).

La part de l'énergie produite par le système qui sera directement consommée sur place serait donc de 73%. Il s'agit du taux d'autoconsommation optimisé grâce au système de gestion intelligent et aux batteries de stockage.

La part d'autonomie énergétique du site par rapport au réseau serait de 67%. Il s'agit là du taux d'autoproduction.

Concernant le financement du projet, le coût s'élève à 529 790 € HT. Les subventions obtenues sont les suivantes :

- Etat (TEPcv) = 250 000 €
- Région Bretagne (contrat de partenariat) : 75 000 €
- Etat (FSIL) : 25 000 €

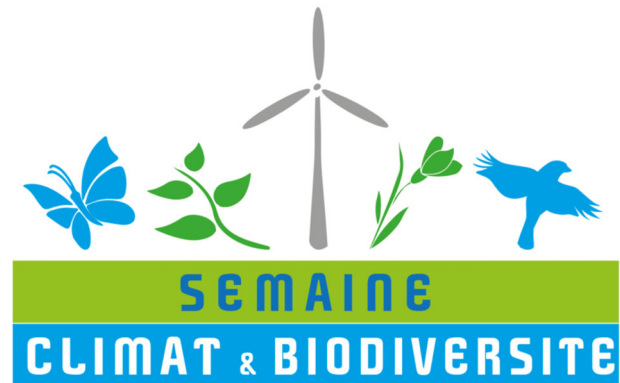
L'économie réalisée est estimée à 250 000 euros sur 20 ans. En termes d'empreinte carbone, cette énergie verte représente 6 tonnes d'émission de CO₂ évitées par an.

IV.2/Le calendrier :

- pré-étude de faisabilité et d'opportunité : 2016
- appel d'offres conception-réalisation : 2017
- études de conception, permis de construire : 2018
- début des travaux : juin 2019
- réception de l'installation : septembre 2019

Enfin, à l'occasion de l'inauguration de ce réseau électrique intelligent, la CCPCAM et ses partenaires ont organisé à la ZA de Kerdanvez, du 30 septembre au 5 octobre 2019, une semaine « Climat et Biodiversité » dédiée à la transition énergétique et écologique en

presqu'île de Crozon et Aulne maritime.



IV.3/ Organisation électrique d'Ener'Glaz

Ce réseau électrique intelligent est organisé à partir de trois tableaux électriques ENR1, ENR2 et ENR3 qui sont présentés ci-après (Fig.7).

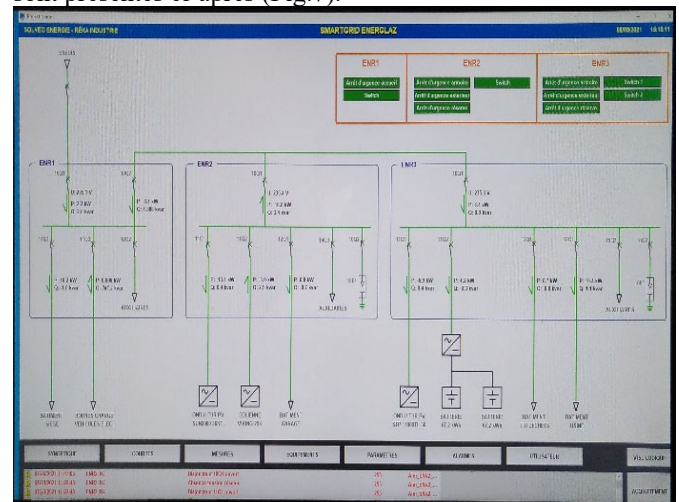


Figure 7 : vue de l'écran de supervision (schéma unifilaire des tableaux et raccordements et flux de puissances) installé dans le bureau du responsable technique de la CCPCAM, ZA de Kerdanvez. Le 5/08/2021 à 15h18min11s.

Tableau ENR1 : raccordé au PdL ENEDIS et aux autres tableaux (ENR2 et ENR3), il permet de d'alimenter le bâtiment du siège de la CCPCAM, les bornes pour véhicules électriques et des récepteurs auxiliaires.

Tableau ENR2 : raccordé à ENR1 et ENR3, il permet le raccordement à l'onduleur de l'éolienne, à l'onduleur de la centrale PV située sur le toit du bâtiment du garage-atelier ainsi que l'alimentation du garage-atelier et des récepteurs auxiliaires (Fig. 8, 9, 10 et 11).



Figure 8 : éolienne (et son onduleur en pied de mât) et centrale PV sur le toit du bâtiment garage-atelier.



Figure 9 : onduleur PV HUAWEI modèle « Sun2000 36KTL » (8 chaînes PV possibles avec 4 entrées MPPT)



Figure 10 : coffret « Solveobox » pour raccordement réseau et mesures/datalogger de supervision pour l'éolienne, les champs PV, le stockage et les récepteurs de ENR2 mais aussi de ENR1 et ENR3. S'y trouvent le jeu de barres de raccordement, un automate et des E/S Wago, les organes de commande et protections ainsi que les borniers.



Figure 11 : petit container extérieur au bâtiment ENR2 contenant l'onduleur PV et le coffret « SolvéoBox ». On remarque les passages de câbles aménagés au sol pour relier les différents tableaux/bâtiments (ENR1 et ENR2 et ENR3)

Tableau ENR3 : raccordé à ENR1 et ENR2, il permet le raccordement à l'onduleur de la centrale PV située sur le toit du bâtiment de gestion des déchets, celui des éléments de stockage électro-chimique ainsi que l'alimentation des bâtiments de tri des déchets et de l'usine de production de compost ainsi que des récepteurs auxiliaires (Fig. 12, 13, 14, 15 et 16).



Figure 12 : bâtiment de tri des déchets bois et vue sur la rangée de modules PV en toiture



Figure 13 : onduleur PV SMA type « Sunny tripower » (capable de fournir de la puissance réactive sur demande) et vue sur un module PV du type de ceux implantés en toiture.



Figure 14 : les deux coffrets de 67kWh chacun contenant les unités de stockage (TESVOLT type « TSHV70 » / 14 batteries de stockage avec cellules « Lithium NMC prismatic Samsung SDI » et équilibrage actif)



Figure 15 : chargeur de batterie SMA type « Sunny tripower storage 60 » associé aux batteries TESVOLT.

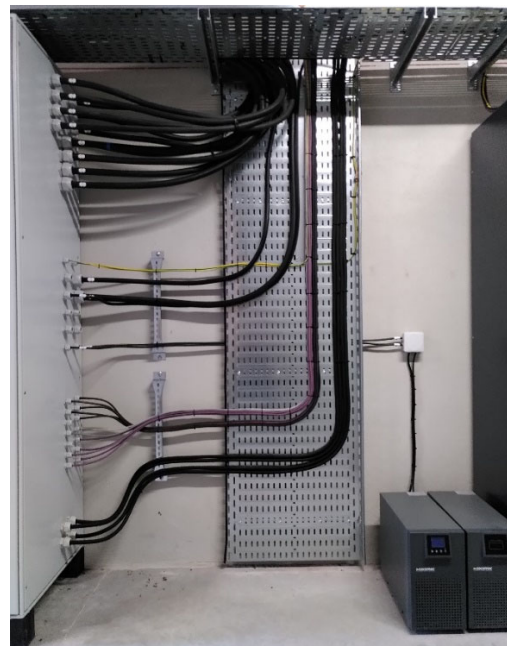


Figure 16 : chemins de câbles pour raccordement des constituants au coffret ENER3

IV.4/Exploitation et maintenance

La CCPCAM n'a pas les compétences spécifiques pour assurer en interne les opérations de maintenance de cette installation (corrective, préventive) au-delà du niveau 1 (qui nécessitent un démontage ou l'ouverture de l'équipement). Un contrat de maintenance multi-technique est envisagé pour faire suite aux différents contrats constructeurs souscrits à la livraison de chaque équipement.

Cependant, un outil logiciel de supervision « SOLVEO Energie » / « REKA Industrie » (Fig. 17 et 18) permet le suivi détaillé de l'installation par le responsable technique de la CCPCAM : synoptique et schéma unifilaire avec données temps réel, productions et consommations instantanées par départ et pour chaque bâtiment, état du stockage électro-chimique et des échanges avec le réseau de distribution (aux formats de courbes et tableaux de valeurs), journal des alarmes sur des seuils et défauts, acquittement, ...

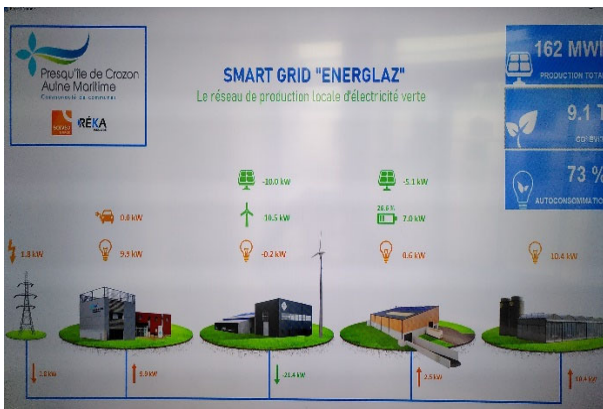


Figure 17 : vue de l'écran de communication au grand public installé dans le hall du siège de la CCPCAM, ZA de Kerdanvez. Le 5/08/2021 aux alentours de 15h. De gauche à droite : le réseau de distribution, le siège de la CCPACAM (ENR1), le bâtiment atelier-garage (ENR2) ainsi que le bâtiment de tri des déchets et celui de l'usine à compost (ENR3).

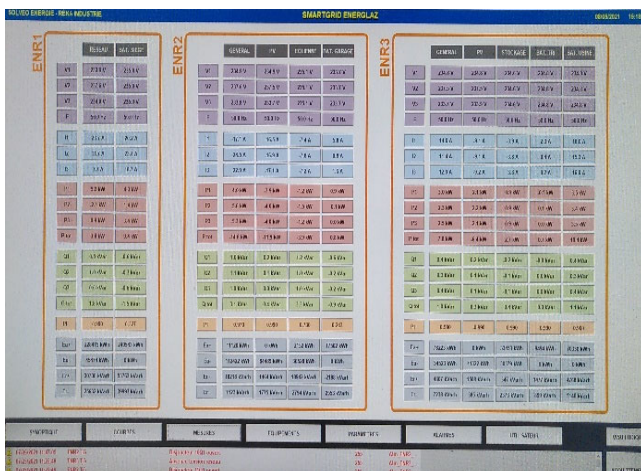


Figure 18 : vue de l'écran de supervision installé dans le bureau du responsable technique de la CCPCAM, ZA de Kerdanvez. Le 5/08/2021 à 15h18min38s. Pour chaque coffret (ENR1, ENR2, ENR3) et par phase du réseau : tension (V), fréquence (Hz), courant (A), Puissances active (kW) et réactive (kVAR), Facteur de puissance, Energies active (kWh) et réactive (kVARh) cumulées

IV.4.1/ Eolienne

L'installation est constituée (Fig. 19, 20 et 21) :

- de l'éolienne danoise Viking VS25 distribuée en France par *ERM Energie*¹¹,
 - d'une armoire en pied de mât de protection et de régulation avec contrôleur IC110,
 - d'un mât,
 - du raccordement au petit container pour livraison et comptage de l'énergie.
- et sa mise en service a eu lieu juste avant le premier confinement, en 2020.



Figure 19 : éolienne Viking VS25 du « smart grid »

OPERATING DATA		APPROVAL	
Nominal power	10 up to 25kW	Type approval according to IEC 61400-2 standard SWT Class III	
Hub height	18 m	GEAR	
Total height	24.5 m	Manufacturer	STM
Power control	Stall	Type	ERX 1501
Active wind range	3 - 25 m/s	SAFETY	
Cut-out wind speed	25 m/s	Brakes	Electromechanical fail safe MAYR 10600-410-3
Survival wind	55 m/s	TOWER	
Yaw system	Active, automatic cable untwist	Type	Viking Wind Gm Pole
Operational temperature	-20C° to +40 C°	Viking VS is produced with lower-bill to ease both installation and maintenance.	
ROTOR		CONTROL SYSTEM	
Type	Stall with tip brake	Manufacturer	Viking Wind
Rotational direction	Clockwise	PERFORMANCE	
Blades	3	Annual production in Denmark Up to 95.000 kWh	
Rotor diameter	13 m	VIKING VS - ANNUAL ENERGY PRODUCTION (mean)	
Swept area	133 m²	Annual averages wind m/s	
Rotor material	Fibre glass	Weibull k	4,5 5,5 6,5 7,5
RPM	Variable up to 65	2	31.000 56.000 76.500 96.000
Tip-speed	Variable up to 45 m/s		
GENERATOR			
Generator	VEIM Motors GmbH		
Type	G21R 200 L4 HW		
Voltage	3 x 400V + N		
Frequency	50 - 60 Hz		
Rated RPM	1525		

Figure 20 : extrait de la documentation de la Viking VS25

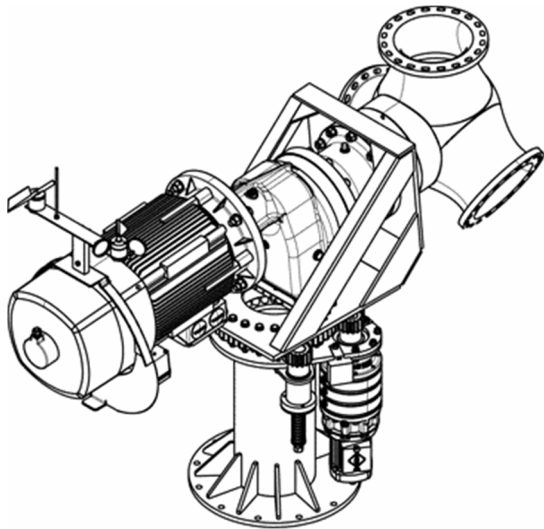


Figure 21 : vue de la chaîne de transmission de puissance de la Viking VS25. De droite à gauche : le moyeu (ou « hub »), le frein mécanique, le multiplicateur et la génératrice asynchrone triphasée sur laquelle sont montés l'anémomètre et la girouette). En bas, à droite, le système mécanique motorisé d'orientation de la nacelle face au vent (ou « yaw »).

Peu de temps après cette mise en service, le responsable technique de la CCPCAM a constaté un dysfonctionnement au niveau du système mécanique d'orientation de la nacelle (« yaw ») et l'aérogénérateur a été arrêté entre les mois de novembre 2020 et mars 2021. Une intervention, difficile en période sanitaire compliquée, a eu lieu dans le cadre du contrat de maintenance d'un an avec le constructeur et l'aérogénérateur est reparti depuis mars 2021 suite au remplacement du moteur du « yaw ». A partir du 27 décembre 2021, plusieurs arrêts se sont produits sur déclenchements d'une protection électrique contre les défauts d'isolement. Une intervention corrective a été réalisée le 23 février 2022 suite au diagnostic de défaillance du système permettant aux câbles à l'intérieur du mât de tourner, créant ainsi des efforts de traction et une déconnexion d'un conducteur dans la boîte à bornes de la génératrice. La visite annuelle de maintenance préventive a été également réalisée à cette occasion et l'installation produit à nouveau.

IV.4.2/ Exploitation des données

Des données (2021) issues de mesures aux différents points de soutirage de l'installation « smartgrid » ainsi que des documents complémentaires nous ont été fournis. Ces données incluent donc des périodes pendant lesquelles l'éolienne était à l'arrêt. Nous proposerons ultérieurement un complément à cet article permettant d'ébaucher quelques conclusions sur les écarts éventuels entre prévisions et mesures et de présenter les préconisations de maintenance du constructeur de l'aérogénérateur.

V/ Centrale solaire PV au sol

La CCPCAM travaille à ce projet de ferme solaire depuis une douzaine d'années. Six ans après avoir obtenu le permis de construire et après cinq tentatives auprès de la Commission de Régulation de l'Énergie (CRE), dont la dernière en 2019, fructueuse, la société briochine IEL¹² (Initiatives & Énergies Locales) a enfin pu lancer la construction de cette centrale, raccordée au réseau de distribution Enedis. « Lors de chaque appel d'offres, le projet crozonnais était en concurrence avec l'ensemble des projets nationaux », rappelle Loïc Picot, directeur d'IEL. « Nous avons échoué à chaque session et le solaire dans l'Ouest était alors quelque chose d'assez difficile à vendre... ». Ce projet, lauréat en 2019 de l'appel d'offres national pour les installations type « centrale au sol » d'une puissance supérieure à 250 kWc, est donc porté par la société IEL en partenariat avec la Communauté de Communes Presqu'île de Crozon – Aulne Maritime (CCPCAM).

Ce sont des entreprises locales (Fig. 22) qui s'y sont investies et principalement :

- ARMORGREEN (BE),
 - IEL (Initiatives & Énergies Locales) spécialisé dans le développement, l'installation et l'exploitation de projets d'énergies renouvelables, éoliens et solaires (centrales au sol, en toiture, autoconsommation) qui, via « Ouest Énergies 1 », assure la maîtrise d'ouvrage du projet et, via « IEL Exploitation », exploite la centrale et en assure la maintenance.
 - ...
- accompagnées par la CCPCAM pour le suivi.



Figure 22 : panneau d'information présentant le rôle de chaque entreprise dans le projet

Les travaux ont démarré au cours de l'été 2020 et la centrale solaire au sol de Kerdanvez (Fig. 23) a été mise en service en février 2021 et officiellement inaugurée le vendredi 10 septembre 2021 dans le cadre d'une journée portes ouvertes¹³. Les travaux ont duré 6 mois pour l'installation des panneaux et bénéficié d'une préparation spécifique du terrain de ce site particulier en amont.



Figure 23 : vue de la centrale PV au sol installée sur l'ancienne décharge de la ZA de Kerdanvez, Crozon.

Cette centrale solaire, qui s'étend sur un site de 6,3 hectares (ancien centre d'enfouissement technique de déchets ménagers fermé et désaffecté depuis 2000, un terrain artificialisé et sans valeur agronomique qui est ainsi valorisé, situé à proximité du réseau électrique local) compte près de 6 500 panneaux photovoltaïques de marque *DMEGC Solar* (pour une puissance installée de 2,27MWc / onduleurs de chaînes de marque *FIMER*) fixés au sol sur des châssis montés sur plots béton, orientés plein sud, inclinés à 25 degrés et dont le niveau de performance est garanti 30 ans par le constructeur (Fig. 24, 25 et 26).

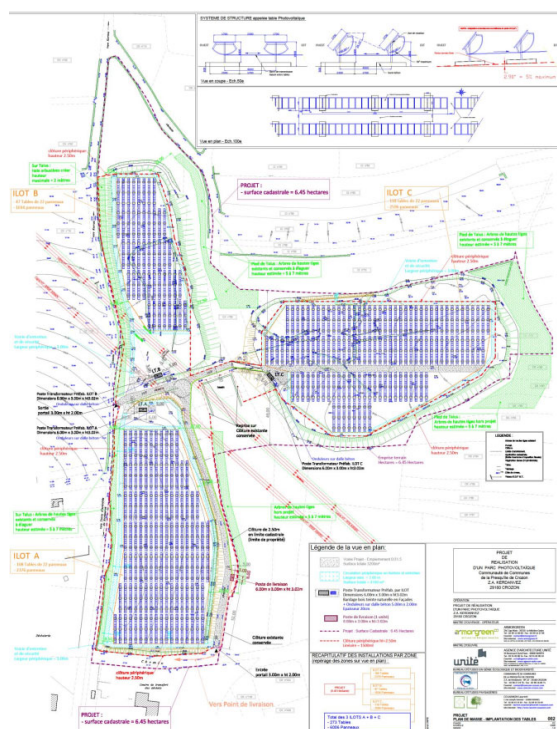


Figure 24 : schéma d'implantation de la centrale PV au sol dans la ZA de Kerdanvez (Crozon), d'après Lendopolis.

Par la revente totale de l'électricité au producteur d'énergie EDF, elle permettra d'alimenter en énergie solaire l'équivalent de la consommation d'environ 800 foyers, chauffage électrique inclus, soit une production estimée par le bureau d'études à 2,8GWh/an équivalent à un fonctionnement d'environ 1200h/an en conditions STC. A titre de comparaison, cette production équivaut également à la recharge d'un parc de 750 véhicules électriques totalisant 11 millions de kilomètres par an (750 véhicules x 15000 km).

Le temps de retour énergétique, c'est-à-dire la durée nécessaire pour produire l'énergie nécessaire à la réalisation du projet, est estimé à moins de 3 années. En fin d'exploitation du projet, les panneaux solaires photovoltaïques seront démontés puis recyclés à 94 % par la filière PV CYCLE¹⁴ dans un souci de valorisation.



Figure 25 : vues des modules montés sur châssis lors de la phase de construction de la centrale PV



Figure 26 : poste HTA de livraison au réseau de distribution public Enedis (Sn=2MVA), ZA de Kerdanvez

V.1/ Financement

L'investissement privé est évalué à 2 millions d'euros. Le tarif d'achat du kWh par EDF se situe autour de 7 centimes d'euros. Le partenariat avec la CCPCAM prend la forme d'un bail emphytéotique de 30 ans impliquant que l'exploitant doit rendre le terrain en l'état et rétrocéder l'installation. L'opération est donc positive pour le contribuable, d'autant que les retombées économiques fiscales sont estimées à 27 300 € par an, réparties entre les collectivités territoriales.

Des mini-bons de financement participatif ont été émis par la société Initiatives Energies Locales (IEL), maison mère du Groupe IEL, qui détient 60% des titres de la société de projet Ouest Energies 1, elle-même propriétaire de la centrale. Le solde est détenu par la société Horizon Photovoltaïque 6, regroupant les investisseurs Lendopolis¹⁵ ayant participé à la première opération (2020) sous forme d'actions.

Le financement de ces investissements privés est donc composé de :

- opérations de financement participatif en actions réalisée fin 2020,
- financement bancaire,
- apports du Groupe IEL, dont une partie financée par l'émission de mini-bons.

Après une première opération en actions d'un montant de 100 K€ réalisée avec succès fin 2020, le Groupe IEL a souhaité réaliser une émission de mini-bons d'un montant maximum de 100k€, avec l'objectif de faire bénéficier aux habitants d'une partie des retombées économiques du projet de la ferme solaire de Crozon. De ce fait, cette collecte a été exclusivement ouverte aux habitants de la Communauté de Communes Presqu'île de Crozon - Aulne Maritime. Les personnes intéressées par cette opportunité de placement vertueux et local ont pu se renseigner ou souscrire directement sur www.lendopolis.com.

V.2/ Exploitation et maintenance (O&M)

IEL est un opérateur solaire et éolien actif sur le Grand Ouest depuis 2004 qui développe, finance, construit et exploite et maintien des parcs éoliens terrestres et des centrales solaires au sol et en toiture (environ 500 à ce jour). Pour l'O&M, il utilise des outils de suivi à distance (Fig. 27) permettant de :

- suivre en permanence la production sans se déplacer,
- s'assurer de toujours produire de manière optimale,
- détecter aussitôt le moindre dysfonctionnement,
- bénéficier d'un service rapide de dépannage pour limiter les pertes de production,
- prévenir les risques d'incendie et diminuer les coûts d'assurance.

Selon les contrats les prestations sont les suivantes :

- assistance administrative dans les relations avec EDF et Enedis,

- tarif préférentiel lors des dépannages,
 - contrôle annuel de l'installation avec
 - vérification thermographique,
 - remise d'un rapport pour l'assurance,
 - surveillance à distance de l'installation par IEL
 - une fois par jour,
 - mise à disposition d'un espace internet pour
 - suivre la production en direct + historique,
 - engagement d'intervention sous 2 jours ouvrés,
 - astreinte 7j/7 (week-end et jours fériés),
- avec, en option, différentes solutions de nettoyage (intervention en nacelle ou avec un robot nettoyeur) pour maintenir sa production initiale, optimiser sa production et pérenniser la centrale.



Figure 27 : technicien de surveillance et télé-maintenance chez IEL

VI/ Conclusions

Contrairement à ce qui s'est fait vers la Pointe du Raz¹⁶ suite à l'épisode nucléaire de Plogoff¹⁷, le développement du photovoltaïque sur ce territoire enclavé entre la baie de Douarnenez, le Menez Hom et la rade de Brest est aussi dû à la difficulté d'y implanter, malgré une ressource abondante, des parcs éoliens tant les installations militaires y sont nombreuses (base aéronavale de Lanvéoc, sémaphore du Cap de la Chèvre, Ile longue, ...

Cette particularité de la Presqu'île permet d'insister sur le caractère propre et unique de tout projet de développement des ENR sur un territoire donné, nécessitant, comme le fait la CCPCAM, de miser sur des ressources présentes et pérennes une fois des études prospectives faites.

Et c'est cela qui doit intéresser nos lectrices et lecteurs aujourd'hui car, au plus près de chez eux, de nombreux projets de réduction des consommations d'énergie (sobriété et efficacité énergétiques) et de production d'énergie à partir de ressources locales renouvelables voient le jour et peuvent, ainsi, servir de supports pédagogiques originaux. Il sont alors vecteurs de sens auprès des élèves et étudiant.e.s de ce territoire car ils en comprennent d'autant mieux les enjeux et le contexte.

VII/ Remerciements et participations

Je remercie vivement pour leur aide et les informations fournies les personnels de la CCPCAM :

- Hubert le Brenn, Directeur Général des Services,
- Didier Prijac, Responsable technique bâtiments, énergies et piscine,
- Laetitia Velly, Responsable communication et qualité.

Ainsi que les personnes et entreprises ayant répondu aux nombreuses sollicitations et questions posées : IEL (Florence Bechen, chargée de communication), Lendopolis, ENEDIS,

1 chiffres 2019 donnés dans « CCPCAM – guide gestion de l'énergie », page 3.

2 www.comcom-crozon.com/

3 www.energence.net/

4 <https://tinergie-brest.fr/>

5 www.pays-de-brest.fr/cadastre-solaire

6 www.rozavern.fr

7 www.solveo-energie.com/

8 www.enercoop.fr/

9 <https://coherence-energies.fr/>

10 www.comcom-crozon.com/environnement/aide-aux-economies-d-energie/energlaz-reseau-electrique-intelligent/

Merci aussi au quotidien Le Télégramme¹⁸ et à ses pages Crozon m'ayant permis de recueillir des compléments d'informations présentés ici et au magazine de la CCPCAM « Horizon » dans lequel divers articles ont été rédigés au sujet de ces installations.

Enfin, merci à mes collègues Gilles Feld et Franck Le Gall pour leurs relectures et conseils, toujours pertinents et bienvenus.

VIII/ Webographie

11 www.erm-energies.com/

12 www.iel-energie.com/

13 www.comcom-crozon.com/actualites/centrale-solaire-portes-ouvertes-et-financement-participatif-le-11-septembre/

14 www.pvcycle.fr

15 www.lendopolis.com/projet/centrale-solaire-iel-crozon-2

16 <https://www.capsizuntourisme.fr/vivre-explorer/detente-loisirs/loisirs/la-maison-du-vent-1858493>

17 www.lesmutins.org/plogoff-des-pierres-contre-des-2090

18 <https://www.letelegramme.fr/finistere/crozon/>

RESEAUX DE DISTRIBUTION

Compensation, Filtrage, Amortissement

JACQUES COURAULT

Résumé : Dans l'industrie, les réseaux de distribution sont soumis à une consommation de puissance réactive préjudiciable à la tenue des tensions et à des harmoniques en relation avec les divers actionneurs présents dans les usines. Compensation de la puissance réactive et filtrage des harmoniques constituent des obligations pour le respect des normes de distribution et de transport... L'amortissement des structures de filtrage, pour éviter les surtensions d'enclenchement par exemple, n'est généralement pas suffisamment pris en compte...

I/ Introduction

La qualité de la tension, aussi bien au niveau du transport qu'au niveau de la distribution, fait l'objet de nombreuses réglementations.

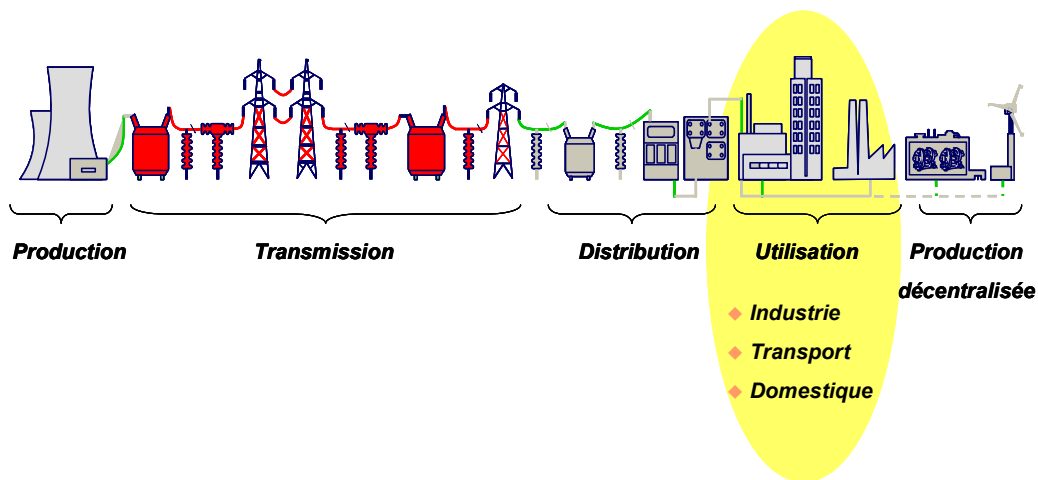


Figure 1 : De la production à la consommation

Les normes : CEI61000-3-6, CEI 61000-3-7, CEI 61000-4-7, CEI 61000-4-15, CEI 61000-4-30... précisent les contraintes qui réglementent la distribution de la tension. En fait, il faut distinguer les problèmes qui ont pour origine la transmission et ceux qui proviennent de la distribution.

Les causes principales sont des courts-circuits dans les postes, sur les lignes aériennes ou câbles souterrains, dont l'origine est atmosphérique, foudre, givre ou tempêtes, il s'agit donc du vieillissement des isolations, mais il y a aussi les erreurs humaines comme de fausses manœuvres.

I.1/ Défauts de transmission

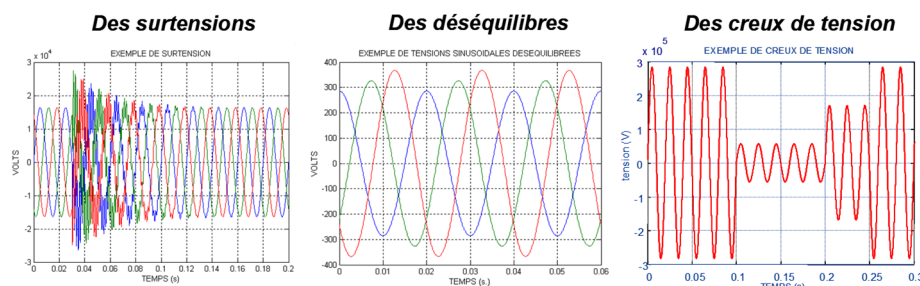


Figure 2 : Conséquences principales des défauts de transmission

1.2/ Réseaux de distribution

Pour avoir une énergie fiable, conduisant à une utilisation satisfaisante la tension, celle-ci doit être d'une qualité adaptée aux besoins. Mais le raccordement au réseau de transport est normé pour protéger les lignes et les autres distributions ou autres utilisateurs.

La pollution des réseaux de distribution est liée d'une part aux harmoniques, aux interharmoniques, et harmoniques irréguliers, qui résultent des convertisseurs à base d'électronique de puissance :

- Harmoniques 5, 7, 11, 13....produits par les ponts de Graëtz...
- Interharmoniques, 3, 9....introduits par des ponts de Graëtz, déséquilibrés par une composante inverse sur les tensions d'alimentation.
- Harmoniques irréguliers, ils sont générés par les convertisseurs indirects qui alimentent des charges à une fréquence différente de celle du réseau.

et d'autre part, au flicker qui est une gêne visuelle, maximale à 9 Hz. L'évaluation de la sévérité de cette gêne est mesurée par le Pst... Un Pst de 1 est assez souvent considéré comme une limite au-delà de laquelle il y a gêne. Cela correspond très approximativement à une variation relative de tension de 0,3 % à une fréquence de 9 Hz. Le flicker est principalement engendré par les fours à arc, les grosses machines à souder et les laminoirs.

Les déséquilibres ont pour origine principale, la traction ferroviaire (alimentation des caténaires en 25 kV, 50 Hz) et les charges domestiques.

Flicker et déséquilibres sont contrôlés dynamiquement par des équipements de type SVC (thyristors) ou STATCOM (IGBT). Encore aujourd'hui, le contrôle des harmoniques est statique, avec des filtres de type LC.¹

II/ Filtrage passif

Pourquoi filtrer ? Quand il existe une pollution harmonique, la réponse est simple, c'est pour éliminer les distorsions engendrées par lesdits harmoniques de courant. En absence de pollution harmonique, la réponse est moins évidente, surtout si l'on souhaite rester synthétique...

II.1/ Absence d'harmoniques

Des filtres peuvent être nécessaires pour assurer la stabilité d'un réseau.

- En général la minimisation du réactif est une préoccupation que l'on cherche à satisfaire en

disposant sur le réseau des batteries de condensateurs. L'enclenchement de ces batteries doit se faire en limitant les creux de tension donc les courants au moment de la mise sous tension. Pour cela des inductances sont mises en série : les ensembles condensateurs inductances forment des filtres qui sont calculés de manière à ne pas introduire des résonances qui seraient cause de surtensions sur les installations.

- Une ligne aérienne ou un câble souterrain peuvent être assimilés à une structure répartie d'inductances et de condensateurs, susceptibles d'osciller et d'être à l'origine de dysfonctionnements graves des installations. D'où assez souvent la nécessité d'introduire des amortissements pour les fréquences possibles d'oscillation. La solution la plus classique, d'introduire ces amortissements, est de le faire avec des filtres amortis.

II.2/ Présence d'harmoniques

Les harmoniques ont pour origine, soit des fonctionnements non linéaires (des saturations) ou des charges non linéaires. L'électronique de puissance, gros variateurs de vitesse ou gros redresseurs pour la conversion de l'énergie, sont des pollueurs de plus en plus répandus : absorption de puissance réactive et production d'harmoniques de courant qui sont à l'origine de déformations inacceptables des réseaux, quand les puissances de court-circuit sont faibles par rapport à la puissance de conversion au sens large. D'où la nécessité d'abord de compenser, puis de filtrer pour respecter les normes au point de raccordement sur le réseau distributeur.

Prise en compte des harmoniques de courant, prise en compte de la puissance de court-circuit pour éviter les résonances parallèles ou antirésonances qui engendreraient des surtensions inacceptables pour les charges alimentées par le réseau.

Le calcul d'un filtre doit prendre en compte

- Les harmoniques générés par la charge.
- Les harmoniques éventuellement pré-existants sur le réseau.
- Les caractéristiques du réseau, puissance de court-circuit (ses variations) et les autres filtres éventuellement présents sur le réseau.
- Variations de fréquence du réseau. Très important pour les réseaux îlotés, exemple typique les réseaux de bord des bateaux...
- Les fréquences de télétransmission (de moins en moins utilisées).
- TIF (Telephone Interference Factor) voir IEEE Std 519-1992, il s'agit d'une courbe de

fréquence MLI supérieure à 2000 Hz, c'est sans problème avec des MOSFET, mais limite avec des IGBT et presque impossible avec des IEGT, ce n'est envisageable qu'avec un fort déclassement en courant...

¹ Note 1 : Les filtres actifs, sont connus depuis une trentaine d'années, mais ils demeurent des outils réservés aux puissances faibles... Leur principe de fonctionnement est de générer des harmoniques en opposition de phase avec les harmoniques générés par une charge polluante. Pour éliminer un harmonique de rang 13 du 50 Hz, il faut être efficace à 650 Hz, il est donc nécessaire d'avoir une

pondération, pour une valeur maximale approximative de 10000, des harmoniques de courant en fonction de la fréquence : son maximum est à environ 2500 Hz.

- La tolérance des divers composants, inductances et condensateurs. La variation de ces composants dans le temps, c'est éventuellement le cas des condensateurs avec fusibles incorporés.

Tout ceci pour répondre d'une part, aux normes imposées par le distributeur d'énergie, et d'autre part pour assurer un fonctionnement satisfaisant des installations.

II.3/ Calcul des filtres passifs

Le calcul des filtres est basé sur l'analyse de l'impédance harmonique Z_h , vue par les consommateurs d'énergie qui produisent des courants harmoniques : soit ces courants sont connus, soit ils font l'objet de suppositions... La prise en compte du réseau est plus ou moins complexe, dans les cas les plus simples une simple inductance est suffisante pour le représenter.

$$S_{cc} = \frac{U^2}{X} \text{ avec } X = L\omega_r \quad I_{cc} = \frac{V}{X} \quad (1)$$

Dans certains cas il existe déjà des filtres, qui sont insuffisants, il faut en tenir compte, parfois l'alimentation du réseau de distribution s'effectue avec des câbles, une structure L,C dont la fréquence de résonance est généralement comprise entre les rangs 60 et 80 du fondamental. Dans ce dernier cas il importe de vérifier l'absence de surtensions engendrées, soit par les harmoniques de courant, soit lors des enclenchements.

La première question qui se pose est de déterminer à quel niveau il convient d'installer la compensation et le filtrage ? Sur le réseau interne de distribution d'une usine, de métallurgie par exemple, il existe plusieurs tensions, en général, les équipements de forte puissance, souvent à l'origine des pollutions, sont alimentés par les tensions les plus élevées d'un site. C'est généralement à ce niveau qu'il convient de mettre en place le filtrage, d'une part pour garantir la qualité de la tension distribuée sur le site et d'autre part pour être conforme aux exigences du fournisseur d'énergie.

Ensuite il faut déterminer la puissance réactive de compensation à installer : soit P la puissance active, Q_n la consommation naturelle de puissance réactive et Q_s la consommation souhaitée, Q_{comp} étant la puissance réactive à installer.

$$Q_{comp} = Q_n - Q_s = P(\tan(\phi_n) - \tan(\phi_s))$$

La puissance réactive Q_{comp} étant à répartir entre les rangs de filtrage, nous donne la somme des condensateurs à mettre en place

$$Q_{comp} = \sum U^2 C \omega_r$$

II.3.1/ La compensation

Il existe des réseaux de distribution sur lesquels il n'y pas d'harmoniques, pas de pollution. C'est assez rare, étant donné la pénétration dans les procédés, des conversions statiques... Il peut alors n'exister qu'un besoin de compensation réalisable, en première approche de manière simple, avec des condensateurs. Mettre un condensateur en parallèle sur un réseau, c'est créer une résonance parallèle, on parle alors d'antirésonance, avec ce réseau (1).

$$Z = \frac{Lp}{1+LCp^2} \rightarrow Z = \frac{jL\omega}{1-LC\omega^2} = \frac{jL\omega}{1-\left(\frac{\omega}{\omega_0}\right)^2} \quad (2)$$

avec

$$\omega_0 = \frac{1}{\sqrt{LC}}$$

Z_h concerne toutes les fréquences sauf le fondamental, c'est pourquoi le réseau n'est représenté que par son inductance L sur la figure (3). Le tracé de $Z_h=g(f)$ de cette même figure est fait avec cette hypothèse. Il apparaît logiquement une antirésonance à la fréquence f_0 pour laquelle Z_h prend une valeur très importante, qui pose un problème... En effet, si un courant harmonique imprévu à la fréquence f_0 apparaît, même très faible, son influence sur la tension sera significative. En plus, et c'est le point le plus critique, l'enclenchement conduit à de fortes surintensités et à la production transitoire d'harmoniques : bref, à un enclenchement très perturbant pour le réseau.

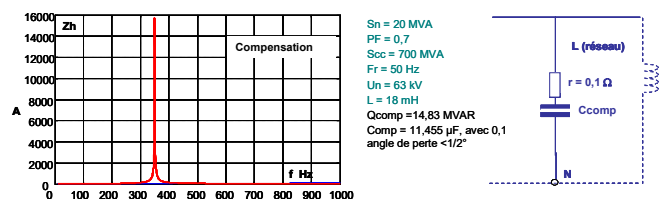


Figure 3 – La compensation, avec juste un condensateur....

Pour les raisons qui viennent d'être évoquées, la compensation dite « sèche » est rarement utilisée.

II.3.2/ Le filtrage

Le filtrage ne consiste pas à bloquer les harmoniques, mais à les court-circuiter avant qu'ils atteignent le réseau. Le blocage des harmoniques ne permettrait pas aux convertisseurs statiques un fonctionnement satisfaisant. Le court-circuit est donc réalisé en accordant la capacité de compensation avec une inductance sur la fréquence d'un harmonique indésirable, connu ou suspecté. La figure 4 représente des filtres accordés sur les rangs 5 et 7 (rangs bas classiques des ponts de Graëtz).

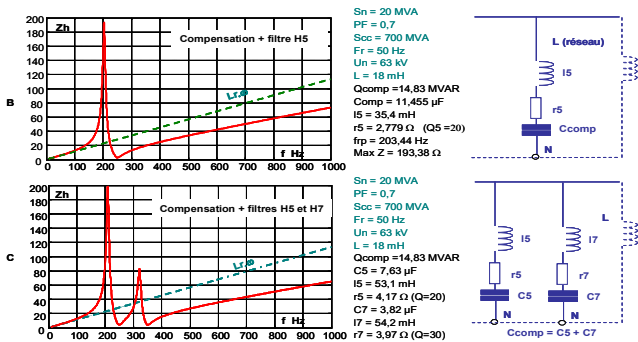


Figure 4 – Deux filtres classiques sur les rangs 5 et 7 du fondamental.

Pour être synthétique nous nous limitons à un calcul avec le rang 5, avec plusieurs branches de filtres, le calcul est nécessairement numérique sous Matlab ou autres logiciels dédiés...

Le fait d'agencer le condensateur de compensation en filtre, modifie légèrement la puissance réactive au fondamental.

$$Q_f = \frac{U^2}{Z_f} = \frac{U^2 \cdot C \cdot \omega}{1 - L \cdot C \cdot \omega^2} \text{ en posant } hx = \frac{F_0}{Fr}$$

on obtient $Q_f = \frac{U^2 \cdot C \cdot \omega}{1 - \frac{1}{hx^2}}$ d'où $C = Q_f \cdot \frac{1 - \frac{1}{hx^2}}{U^2 \cdot \omega}$

Même pour $hx = 5$, la correction par rapport à C_{comp} est faible, pour les rangs supérieurs elle est négligeable. L'analyse de $Z_h = g(f)$, qui prend en compte le réseau et le filtre, est toujours faite en considérant l'impédance du réseau $Z_r = L \cdot \omega$. La figure 4 montre clairement une antirésonance à $f_p \sim 203$ Hz, qui a pour origine le filtre et le réseau : le maximum de $Z_h(f_p) = 193 \Omega$ est bien supérieur à l'impédance du réseau à cette même fréquence, environ 22Ω .

$$\omega_p \approx \omega_d = \frac{1}{\sqrt{(L+l)C}} = \frac{1}{\sqrt{\left(\frac{U^2}{S_{cc}\omega} + \frac{1}{C\omega_0^2}\right)C}}$$

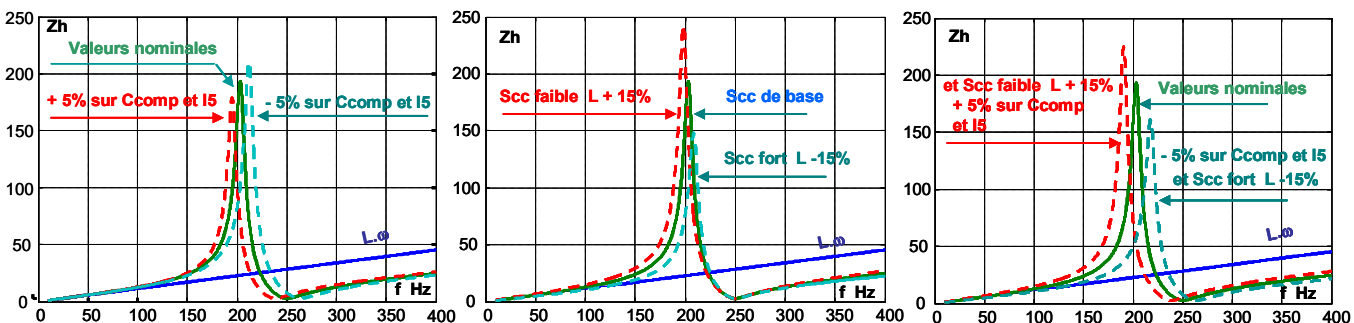


Figure 5 – Influence de la variation des composants, pour un filtre sur le rang 5.

Du fait des évolutions de la courbe $Z_h = g(f)$ quand les composants du filtre évoluent ou qu'ils sont à la limite de la tolérance des spécifications, la fréquence d'accord et la fréquence d'antirésonance sont modifiées et affectent la qualité du filtrage. Le point le plus

critique, c'est la variation de S_{cc} vis-à-vis de l'antirésonance.

Les valeurs des composants du filtre peuvent varier dans le temps, en particulier la valeur du condensateur. Ce dernier est construit avec des mises en parallèle de condensateurs élémentaires, protégés par des fusibles

$$\omega_p \approx \omega_d = \frac{1}{\sqrt{1 + \frac{Q_f h x^2}{S_{cc}}}} \quad (3)$$

cela signifie, en particulier, que pour S_{cc} important, les fréquences f_p et f_0 sont proches :

$$f_p \approx f_0 \cdot \sqrt{\frac{l}{L+l}}$$

Mais ce qui est le plus critique c'est l'importance du rapport Z_h/Z_r à la fréquence f_p ... Dans le cas de la figure 4 on peut supposer qu'il n'existe aucun harmonique à la fréquence f_p , l'harmonique de tension serait amplifié du fait de l'importance de Z_h ... f_p se trouve centré par rapport à 150 Hz et 250 Hz, c'est donc la proximité du 150 Hz qui peut engendrer des difficultés suite à l'évolution dans le temps des composants du filtre ou les variations de la puissance de court-circuit. Sur un réseau, suite à des régimes transitoires, du fait de la saturation de certains circuits magnétiques il est toujours possible d'avoir de l'harmonique 3.

$$Z_h = \frac{L \cdot l}{L+l} \cdot p \cdot \frac{p^2 + \frac{r}{l} \cdot p + \frac{1}{L \cdot C}}{p^2 + \frac{r}{L+l} \cdot p + \frac{1}{(l+L) \cdot C}}$$

$$Z_h = L \cdot p \cdot \left(\frac{\omega d}{\omega n}\right)^2 \cdot \frac{p^2 + 2 \cdot z n \cdot \omega n \cdot p + \omega n^2}{p^2 + 2 \cdot z d \cdot \omega d \cdot p + \omega d^2} \quad (4)$$

$$\frac{Z_{hmax}}{L \cdot \omega d} \approx \frac{1 - \left(\frac{\omega d}{\omega n}\right)^2}{2 \cdot z d} \approx \frac{L \cdot \omega d}{r} \approx \frac{L \cdot \omega r}{r} \cdot \frac{\omega d}{\omega r} = \frac{X \cdot \omega d}{r \cdot \omega r}$$

ceci pour un filtre à une branche.

Cette formule fait apparaître deux fréquences particulières, ωn au numérateur, c'est la fréquence de filtrage et ωd , au dénominateur qui est très proche de la fréquence de résonance, quand $z d$, l'amortissement est faible : elle est donnée en (3).

incorporés (c'est une option). Bien qu'une surveillance du point neutre des filtres soit réalisée, pour contrôler l'intégrité des condensateurs, la rupture de ces fusibles peut engendrer des dysfonctionnements dans le fonctionnement des filtres.

L'augmentation de l'impédance Z_h à la fréquence de 200 Hz n'est généralement pas admissible et demande l'introduction d'un amortissement. La figure 6 montre que s'il existe un harmonique à la fréquence de 200 Hz (assez peu probable à 200 Hz), il est amplifié aussi bien dans le réseau que dans le filtre, d'où pollution et surcharge du filtre.

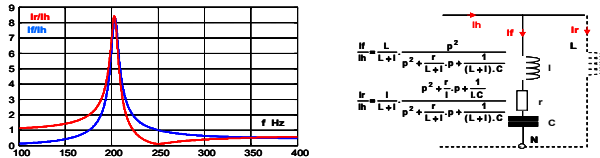


Figure 6 – Courants filtre et réseau, cas B.

II.3.3/ L'amortissement passif.

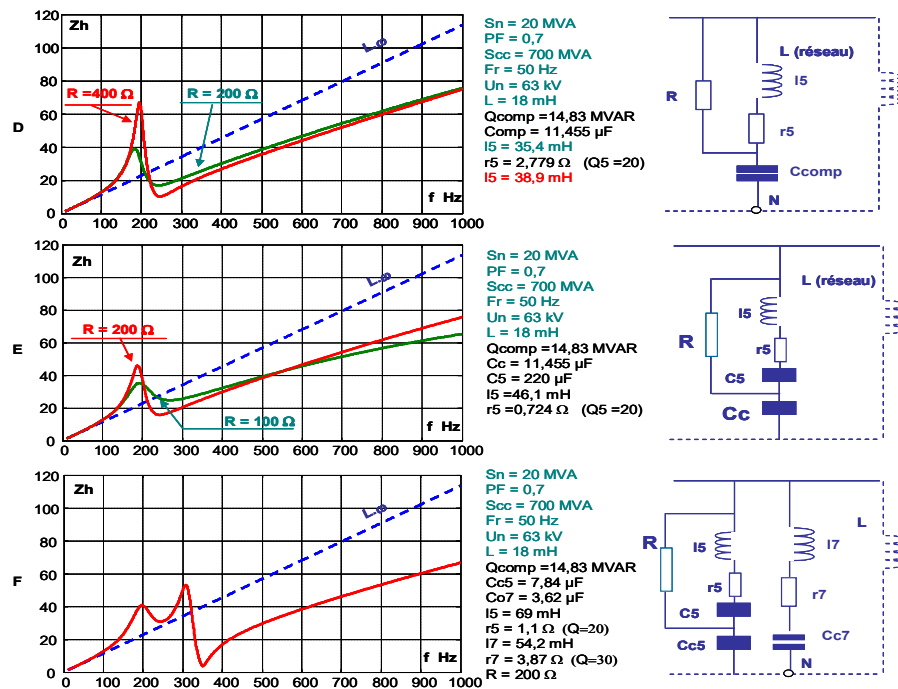


Figure 7 – Amortissement des antirésonances.

Le calcul complet des pertes dans une résistance suppose la connaissance des courants harmoniques, c'est la particulier le cas pour le schéma F de la figure 7. Pour ce type filtre les pertes au fondamental sont nulles du fait de l'accord I_5, C_5 . Mais pour le schéma E, les pertes au fondamental existent et peuvent être facilement calculées. : le courant dans C_{comp} étant connu, c'est la compensation, on peut calculer la tension aux bornes de R , avec les valeurs retenues on arrive pour $R = 200 \Omega$, à 10,44 kW, soit 31,32 kW pour les trois résistances, pertes auxquelles il convient d'ajouter les pertes dues aux harmoniques. Ces pertes ne sont pas négligeables...

Le but de l'amortissement est de limiter l'amplitude des antirésonances. Quand il est passif, il s'agit d'introduire des pertes résistives dans le schéma des filtres. La difficulté est de limiter l'amplitude des antirésonances en affectant le moins possible le filtrage : dans la formule (4) il est évident que la résistance série r , a autant d'influence sur la résonance série que sur la résonance parallèle... Un choix fréquent est de disposer en parallèle sur l'inductance du filtre une résistance R pour l'amortir. C'est cas du schéma D de la figure (7). Dans les structures plus complexes, 4 ou 5 branches, cette résistance R est plutôt introduite sur les rangs les plus élevés.

Pour certaines applications, les fours à arc par exemple, le premier rang de filtrage est entre 3 et 5. Dans ces cas, la structure E est généralement utilisée. Son intérêt réside dans le fait que les pertes, au fondamental, dans la résistance R , sont nulles. Cela est dû à un accord de I_5, C_5 sur le fondamental : c'est donc le condensateur C_{c5} qui fait la compensation... La fréquence d'accord série de ce filtre est bien difficile à approcher simplement, en général on y arrive de manière itérative en démarant le calcul avec les données suivantes :

$$I_5 \cdot C_5 = \frac{1}{\omega r^2}, \quad C_{c5} \text{ est connu (compensation)}$$

$$\frac{15 \cdot c5 \cdot Cc5}{Cc5 + c5} \approx \frac{1}{\omega^2} \text{ soit } C5 \approx Cc5 \cdot \left(\left(\frac{\omega 5}{\omega r} \right)^2 - 1 \right)$$

Clairement il s'agit d'un mauvais filtre, mais d'un bon amortisseur. La valeur de C5 montre à l'évidence le manque d'intérêt de ce filtre pour les rangs d'accord supérieurs à 5 (C5 ~ 24.Cc5).

Ces quelques considérations montrent bien qu'il y a un dilemme entre filtrage et amortissement...

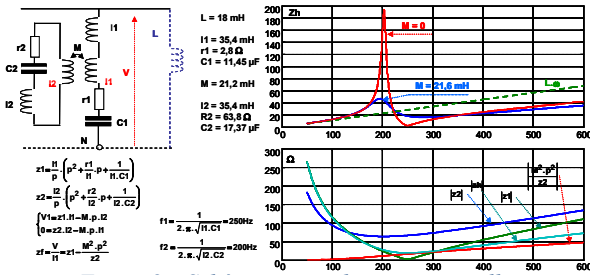


Figure 8 – Schéma avec inductance mutuelle.

Ce dispositif de filtrage, figure 8, n'est évoqué ici que par souci d'exhaustivité, il est assez peu pratiqué. Le principe est assez simple, par l'intermédiaire d'une

inductance mutuelle on introduit un filtre secondaire série accordé sur la résonance parallèle... A cette fréquence on introduit donc une résistance une résistance en série avec le filtre primaire, se qui a pour effet d'amortir la résonance. Globalement le filtre est du quatrième ordre et assez délicat à calculer. On constate qu'il s'agit d'un filtre assez mauvais, mais d'un bon amortisseur. De plus ses pertes sont supérieures à un filtre de type « C » représenté en E...

II.3.4/ L'enclenchement des filtres passifs

L'enclenchement des filtres sur un réseau, n'est généralement pas une manœuvre journalière, mais elle doit néanmoins ne pas introduire des déclenchements par manque tension ou surintensité. Il importe donc de vérifier la valeur transitoire des surintensités et leur durée.

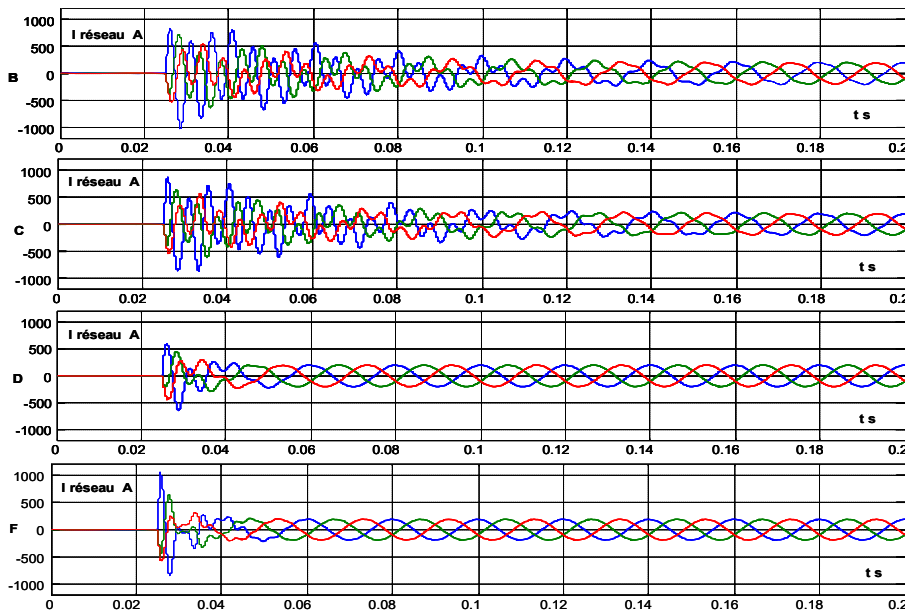


Figure 9 – Enclenchement de différents filtres (Matlab/Simulink).

Rappelons les valeurs du réseau qui sont mentionnées sur les figures précédentes. Un = 63 kV, Scc = 700 MVA, puissance nominale = 20 MVA. D'où Icc = 6,4 kA, In = 550 A, If = 136 A. Pendant les premières millisecondes qui suivent l'enclenchement, les courants ne sont pas sinusoïdaux, Pour qualifier les courants d'enclenchement il est donc préférable d'introduire la valeur crête des courants : soit Inc = 778 A et Ifc = 192 A. Les courants d'enclenchement ne dépassent pas 1000 A, donc les protections du réseau ne seront pas sollicitées. Toutefois une gestion du réseau peut être nécessaire pour éviter un déclenchement. Il est

fréquent de masquer une surintensité prévisible pendant le transitoire qui lui est associé (dans le cas présent, 20 à 40 ms).

Il est à noter que le régime transitoire est d'autant plus long, que le filtre est mal amorti.

II.3.5/ L'amortissement dynamique

II.3.5.a/ Principe de l'amortissement dynamique

Pour lever le dilemme filtrage, amortissement on peut avoir recours à l'amortissement dynamique. Cette pratique a un intérêt surtout pour les antirésonances basses ($f_p < x.fr$),

L'amortissement dynamique est réalisé avec un convertisseur statique de type MLI, dont le contrôle permet l'élimination ou la réduction des antirésonances... Ledit convertisseur statique doit donc avec une bande passante compatible avec son action souhaitée sur les antirésonances f_p , c'est-à-dire une fréquence de découpage égale à 3 ou 4 fois f_p . Or, les fréquences f_p sont toujours inférieures aux fréquences d'accord, qui correspondent à la fréquence des harmoniques présents.

Cette description peut faire penser à un filtre actif... Mais la finalité et le dimensionnement du convertisseur MLI ne sont pas les mêmes. Un filtre actif mesure les harmoniques de la charge polluante et en génère d'autres en opposition de phase... Un filtre actif doit avoir une précision bien plus grande qu'un amortisseur, donc une bande passante plus élevée, donc une fréquence de découpage plus élevée.

La formule 4 montre que l'antirésonance a pour origine son dénominateur, les amortissements des polynômes du numérateur et du dénominateur sont presque identiques... Le but de l'amortissement dynamique est d'augmenter l'amortissement du dénominateur, sans toucher à celui du numérateur, dont dépend la qualité du filtrage. Pour cela on introduit une source de courant I_a en parallèle sur le filtre.

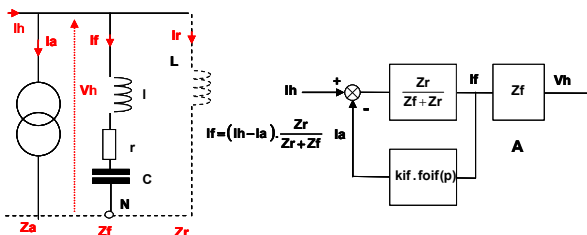


Figure 10 - Principe de l'amortissement dynamique

Le courant I_a est lié au courant I_f par une relation de la forme, $kif.foif(p)$... D'où la création d'un système bouclé...

$$\frac{I_f}{I_h} = \frac{\frac{Z_r}{Z_r+Z_f}}{1+kif.foif(p).\frac{Z_r}{Z_r+Z_f}} = \frac{Z_r}{Z_f+Z_r.(1+kif.foif(p))}$$

$$\frac{I_f}{I_h} = \frac{L.p}{(i+L).p+r+L.kif.foif(p).p+\frac{1}{C.p}}$$

si nous posons $foif(p) = \frac{1}{p}$ nous obtenons,

$$\frac{I_f}{I_h} = \frac{L.p}{(i+L).p+r+L.kif+\frac{1}{C.p}}$$

le terme $L.kif$ se comporte comme une résistance, il participe ainsi à l'amortissement et cela sans affecter la fréquence propre ou le gain global du système. Le calcul de l'impédance harmonique du réseau est immédiat...

$$Z_h = \frac{V_h}{I_h} = \frac{L.p.\left(l.p+r+\frac{1}{C.p}\right)}{(i+L).p+r+L.kif+\frac{1}{C.p}}$$

$$Z_h = L.p.\left(\frac{\omega d}{\omega n}\right)^2 \cdot \frac{p^2+2.zn.\omega n.p+\omega n^2}{p^2+2.zd.\omega d.p+\omega d^2}$$

$$\text{avec } Z_d = \frac{1}{2}(r+kif.L)\sqrt{\frac{C}{L+i}}$$

L'augmentation de l'impédance Z_h à la fréquence de résonance parallèle ωd , par rapport à l'impédance du réseau à cette même valeur a pour expression est β_{ad} :

$$\beta_{ad} = \frac{Z_h \max}{L.\omega d} \approx \frac{1-\left(\frac{\omega d}{\omega n}\right)^2}{2.zd} \approx \frac{L.\omega d}{r+kif.L}$$

$$\beta_{ad} \approx \frac{L.\omega r}{r+kif.L} \cdot \frac{\omega d}{\omega r} = \frac{X_r}{r+kif.L} \cdot \frac{\omega d}{\omega r}$$

Le rapport entre β_{ad} (avec amortissement dynamique) et β_{sad} (sans amortissement dynamique) est :

$$\frac{\beta_{ad}}{\beta_{sad}} = \frac{r}{r+kif.L}, \text{ dans l'exemple pris en compte, figure 4B, } r = 2,8 \Omega, L = 18 \text{ mH, } Z_{\max} = 193,4 \Omega$$

si nous souhaitons réduire l'impédance maximale à 50Ω , kif doit avoir pour valeur 446 r/s .

Nous avons volontairement confondu fréquence propre et fréquence de résonance, ce qui est légitime pour les amortissements faibles inférieurs à $0,5$

$$\omega_{Re} = \omega_0 \sqrt{1-z^2}$$

L'actionneur, avec lequel la fonction d'amortissement dynamique est réalisée, doit au moins être actif à la fréquence d'antirésonance, soit à 200 Hz .

Les courbes ci-contre de la figure 11 montre clairement l'influence de kif sur l'antirésonance et son absence d'influence sur la résonance série, donc sur la qualité du filtrage.

Nous avons introduit dans nos calculs un filtrage sur kif à 3000 r/s pour tenir compte de la dynamique de l'actionneur de type MLI.

Etre actif à 200 Hz suppose, au moins pour l'actionneur, une bande passante de 500 Hz , soit une fréquence minimale de découpage de 1500 Hz , qui est compatible avec des équipements de puissance de moyenne de l'ordre de 5 MVA .

La fréquence découpage apparente peut être augmentée en utilisant des topologies imbriquées multicellulaires. En quatre niveaux par exemple, une fréquence de découpage de 1000 Hz donne une fréquence, en sortie de convertisseur de 3000 Hz .

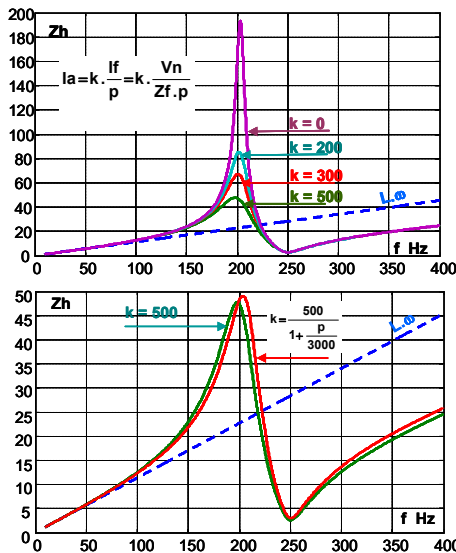


Figure 11 – Quelques résultats de l'amortissement dynamique.

Nous avons choisi de lier le courant Ia au courant If, on peut légitimement se demander si d'autres solutions étaient envisageables ? Par exemple une liaison au courant réseau Ir ou à la tension Vh ?

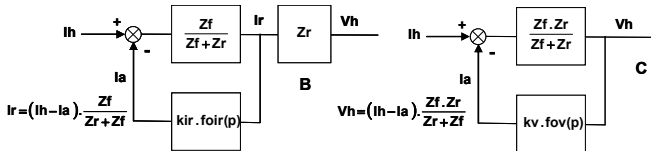


Figure 12 – Autres solutions envisageables.

Quand on effectue les calculs qui correspondent à ces deux structures, on constate assez vite qu'elles sont moins « pures » que la première retenue : liaison de Ia avec If... En effet on constate qu'il y a bien une action sur l'amortissement, mais que la fréquence propre est affectée par le bouclage ainsi que le gain global du système : c'est la raison pour laquelle nous ne les retenons pas...

II.3.5.b/ Mise en place de l'amortissement actif

De plus la structure retenue fonctionne même dans le cas d'un filtre plus complexe, comprenant une branche H5 et une branche H7, pour autant que la bande passante de la source de courant Ia le permette...

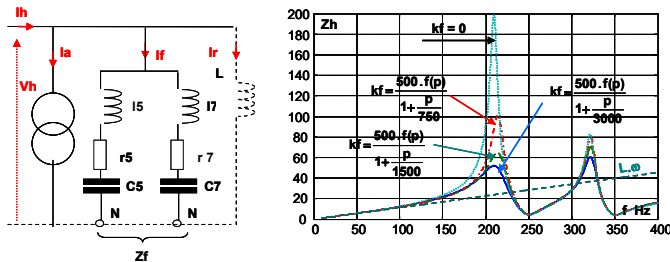


Figure 13 – Introduction de H7.

Il est nécessaire, à ce niveau de l'explication, de faire quelques considérations sur le dimensionnement en courant, de la source de courant...

Dans le cas d'une charge produisant des harmoniques 5, 7... Il est logique de constater que ces courants sont essentiellement absorbés par les filtres (If/Ih = 1).

Ces courants sont également perçus faiblement par la source de courant :

$$\left(\frac{I_a}{I_h}\right)_{h=5} \approx 0,24 \text{ et } \left(\frac{I_a}{I_h}\right)_{h=7} \approx 0,2$$

si des courants ayant pour origine des harmoniques irréguliers, existent, ils sont faibles et bien que l'amplification soit forte (0,77) le niveau de courant pour l'actionneur est faible et non dimensionnant.

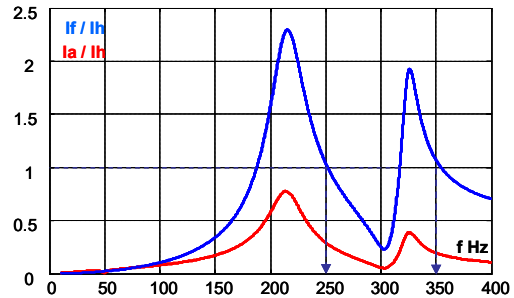


Figure 14 – Courants filtre et actionneur de la source de courant, en fonction du courant harmonique, Ih généré par la charge

Rappelons, que les schémas utilisés ne sont valables que pour les courants harmoniques. Néanmoins, quant on considère un courant Ia = kif.If/p, Ia contient la composante fondamentale du courant dans les filtres (courant de compensation). Ce courant serait très dimensionnant pour l'actionneur alors qu'il n'a rien à voir avec l'amortissement : il convient donc de l'éliminer par une forte réjection. D'où l'expression utilisée pour introduire le coefficient kif, (voir f(p) dans la figure 13).

$$kif = k_o \cdot \left(\frac{p^2 + 2.0,002 \cdot \omega_r \cdot p + \omega_r^2}{p^2 + 2.0,05 \cdot \omega_r \cdot p + \omega_r^2} \right)^2 \cdot \frac{1}{1 + \frac{p}{3000}}$$

Dans les lignes qui précèdent nous avons évoqué un système bouclé, il importe d'en vérifier la stabilité par une analyse sommaire de la fonction de transfert en boucle ouverte...

$$G_{bo} = \frac{Z_r}{Z_f + Z_r} \cdot k_o \cdot \left(\frac{p^2 + 2.0,002 \cdot \omega_r \cdot p + \omega_r^2}{p^2 + 2.0,05 \cdot \omega_r \cdot p + \omega_r^2} \right)^2 \cdot \frac{1}{1 + \frac{p}{3000}}$$

Les deux cercles qui apparaissent correspondent respectivement aux filtres sur H5 et H7.

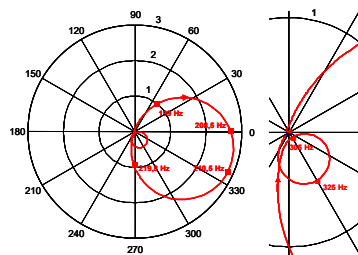


Figure 15 – Analyse de la stabilité avec ko = 500 r/s

La rotation de l'axe diamétral, d'environ -20° , des cercles est due à la bande passante de l'actionneur, soit à la présence du filtre à 3000 r/s. Bande passante qui aussi pour conséquence une réduction du diamètre de ces cercles

La figure 16 est la simulation globale d'un système de filtrage avec amortissement dynamique. Les harmoniques sont introduits par le biais d'un pont de diodes, alimenté directement en 63 kV, ce qui n'est pas très réaliste, mais le but était uniquement d'introduire des harmoniques.

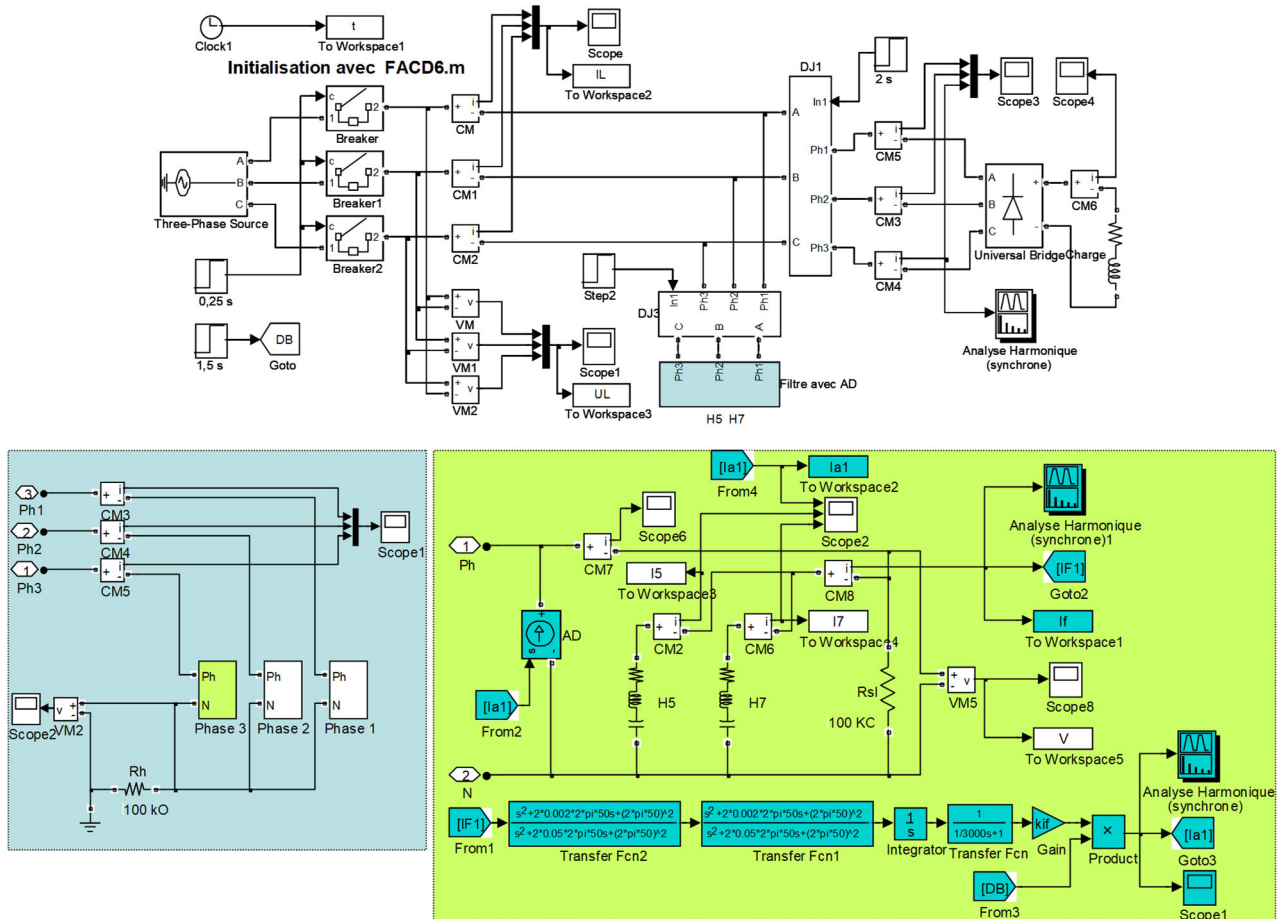


Figure 16 – Simulation globale de l'amortissement dynamique.

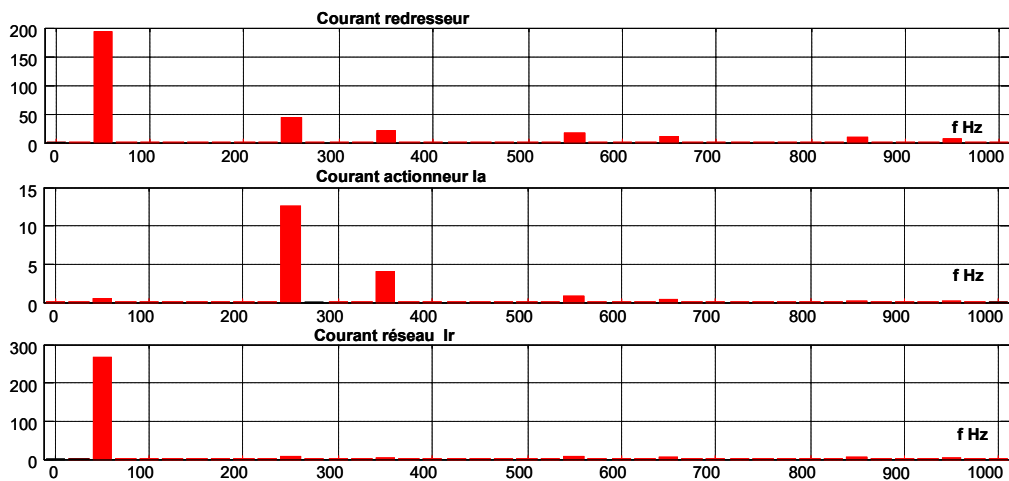


Figure 17 – Résultats de la simulation de la figure 16

II.3.5.c/ Limites de l'amortissement actif

Ces éléments de dimensionnement prennent en compte deux rangs de filtre, H5 et H7. L'amortissement dynamique est efficace pour les deux rangs, mais compte tenu de la bande passante retenue pour l'actionneur, son action est beaucoup moins efficace pour le rang 7 que pour le rang 5. De plus, nous avons pris en compte une réjection du fondamental en introduisant la fonction $f(p)$ dans le coefficient kif, ceci dans le but de ne pas solliciter l'actionneur par le courant de compensation du facteur de puissance. Cette réjection à 50 Hz n'a pas ou peu d'influence sur l'amortissement.

Sur la figure 6, à l'antirésonance, le rapport entre le courant harmonique total et le courant dans le filtre est de l'ordre de 8,5, avec l'amortissement dynamique il n'est plus que de 2,2... C'est un gain appréciable, mais on peut se demander le pourquoi de l'intérêt d'un tel gain, puisqu'en principe, il n'y a pas d'harmonique à 200 Hz ? Dans les régimes transitoires tout peut arriver, en particulier de l'harmonique 3, dû à des saturations, de plus, une composante inverse sur les tenions d'alimentation des ponts de Graëtz engendre de l'harmonique 3 et 150 Hz c'est assez proche de 200 Hz.

Mais il y a pire quand le réseau alimente des actionneurs à grande dynamique, pour des laminoirs par exemple. Dans ce cas il peut y avoir des accrochages entre les boucles de régulation et les filtres de rang bas, d'où bien sûr des dysfonctionnements des installations et des échauffements, voire la destruction des filtres.

Cependant, cette remarque liée à la dynamique des entrainements, était surtout pertinente pour les convertisseurs directs à base de ponts de Graëtz : alimentation de moteurs de moteurs à courant continu ou de machines à courant alternatif avec des cycloconvertisseurs. Avec les alimentations indirectes en tension, ponts de diodes, condensateurs et convertisseurs à IGBT, MLI, la dynamique est généralement plus faible coté réseau que coté machine, du fait de la présence des éléments de stockage, que sont les condensateurs. Donc les risques d'accrochage sur les filtres du réseau sont considérablement réduits.

De plus les ponts de diodes consomment moins de puissance réactive que les ponts à thyristors. Les harmoniques produits sur le réseau sont plus faibles, du moins pour les rangs élevés. Il faut également noter que les gros équipements, perturbateurs, sont au minimum alimenté en 12 pulses (dodécaphasé), voire en 18 pulses, et même 24 pulses. En 24 pulses les premiers rangs qui apparaissent, sont à 1150 Hz et 1250 Hz, et cela avec de très faibles amplitudes... Dans ces conditions, le filtrage est bien difficile à justifier.

Mais le besoin de compensation demeure... D'où la nécessité de condensateurs sur le réseau avec le problème des antirésonances et des courants d'enclenchement desdits condensateurs, ce qui conduit à mettre des inductances en série avec ces condensateurs. Le choix naturel est donc un mauvais

filtre, mais bon amortisseur, c'est le schéma E de la figure 7.

II.3.5.d/ Dimensionnement de l'actionneur

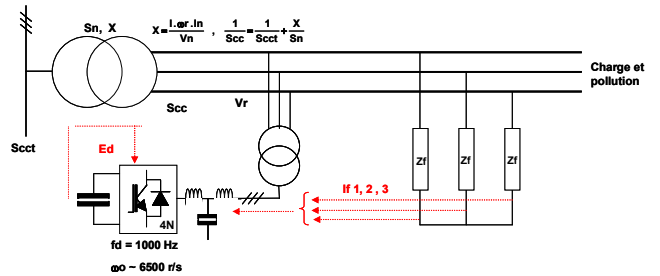


Figure 18 – Mise place de la solution.

L'actionneur Ia est introduit sur le réseau au moyen d'un transformateur abaisseur. Le rapport de transformation est déterminé pour ramener le courant Ia et la tension à des valeurs compatibles avec le convertisseur MLI. Pour les fortes puissances ce convertisseur peut être multiniveaux à cellules imbriquées, ce qui permet d'augmenter la fréquence de découpage vue du réseau (si fd est la fréquence de découpage des IGBT, avec un quatre niveaux, on obtient une fréquence de découpage apparente de 3.fd).

La puissance de dimensionnement d'un transformateur est liée au courant (section de cuivre), à la tension et à la fréquence (nombre de spires et section de fer)... Le schéma proposé suppose l'injection du courant d'amortissement avec un transformateur raccordé au niveau du filtre. Donc, bien que les courants soient faibles, du fait de la tension, la puissance du transformateur d'injection est relativement importante. D'où l'idée, encore une fois, de reconsidérer le schéma E de la figure 7, qui introduit l'amortissement en éliminant la tension fondamentale.

II.3.5.e/ Elimination de la tension fondamentale

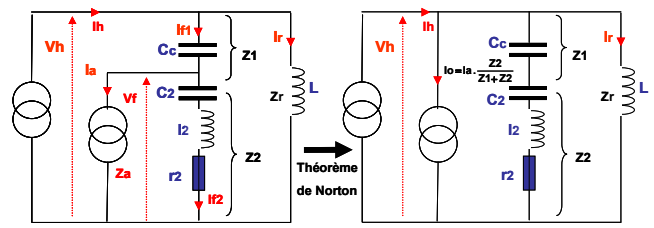
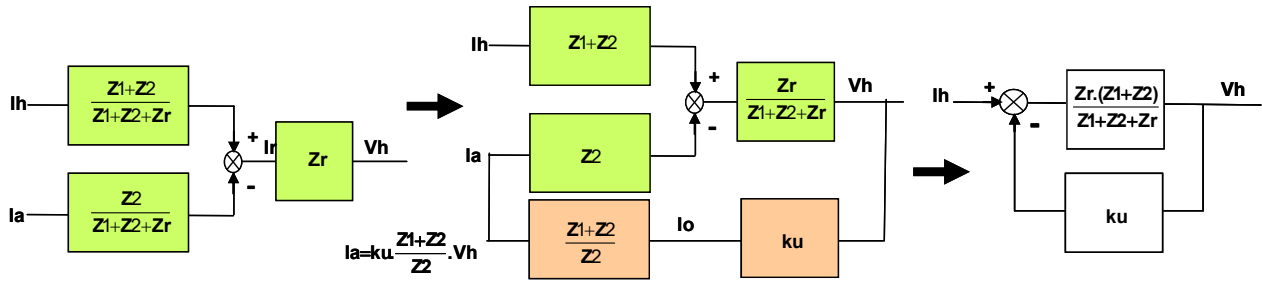


Figure 19 : Amortissement dynamique en éliminant la tension fondamentale, $I_2 C_2 \omega_r^2 = 1$.

La figure 19 est l'aménagement de la figure 7 dans le cas de l'amortissement dynamique. La figure 19, à droite montre que nous sommes revenus au cas de base de la figure 10, mais avec une injection de courant plus complexe, qui fait intervenir les impédances Z1 et Z2. De plus, la liaison du courant Ia avec le courant If est impossible... On a donc recours à une liaison avec Vh, sachant bien que le bouclage par Vh n'est pas idéal, pour des raisons déjà évoquées dans les lignes qui précèdent.

Figure 20 : Introduction d'un bouclage par V_h

Pour des raisons de stabilité on est conduit à introduire Z_1 et Z_2 dans la liaison, ou retour, de V_h vers I_a ,

$$I_a = k_u \cdot \frac{Z_1 + Z_2}{Z_2} \cdot V_h$$

ce qui permet d'obtenir la représentation finale du système, figure 20 à droite.

$$I_a = k_u \cdot \left(\frac{p^2 + 2 \cdot 0,002 \cdot \omega_r \cdot p + \omega_r^2}{p^2 + 2 \cdot 0,05 \cdot \omega_r \cdot p + \omega_r^2} \right)^2 \cdot \frac{Z_1 + Z_2}{Z_2} \cdot \frac{V_h}{1 + \frac{p}{3000}}$$

en prenant en compte l'élimination du fondamental et la bande passante de l'actionneur.

Notons que :

$$Z = Z_r \cdot \frac{Z_1 + Z_2}{Z_1 + Z_2 + Z_r}$$

est l'impédance harmonique sans amortissement dynamique.

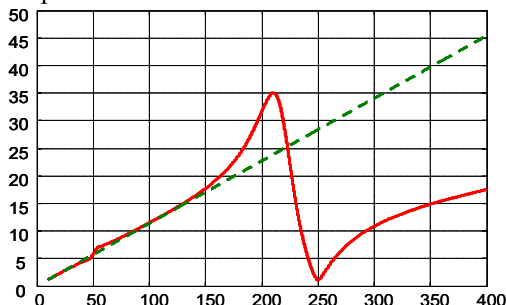
$$Z_h = \frac{V_h}{I_h} = \frac{Z}{1 + k_u \cdot Z}$$

d'où la fonction de transfert en boucle ouverte,

$$G_{bo} = k_u \cdot \left(\frac{p^2 + 2 \cdot 0,002 \cdot \omega_r \cdot p + \omega_r^2}{p^2 + 2 \cdot 0,05 \cdot \omega_r \cdot p + \omega_r^2} \right)^2 \cdot \frac{Z}{1 + \frac{p}{3000}}$$

La figure 21 montre un résultat satisfaisant en apparence... Cependant, la présence de Z_2 au dénominateur, dans le retour, introduit une augmentation très importante du fondamental, que le dispositif de réjection à fr a bien des difficultés à combattre. Donc dans la subsiste un courant fondamental qui introduit une instabilité du système surtout dans le cas générique d'un filtre en étoile à neutre isolé...

Cette structure d'amortissement dynamique ne doit cependant pas être rejetée... Elle requière une structure plus complexe de contrôle...

Figure 21 : Impédance harmonique en fonction de la fréquence avec $k_u = 0,03$

III/ Conclusions

La variation de vitesse de forte puissance est un domaine d'enseignement complexe, qui fait appel à des notions pluridisciplinaires : électricité, mécanique, automatique et thermique, voire à la chimie pour les systèmes de refroidissement sophistiqués.

- Electricité et automatique sont les domaines de base.
- Mécanique, pour les problèmes vibratoires (en association avec l'automatique) et la tenue aux efforts électrodynamiques.
- La thermique et la chimie pour les systèmes de refroidissement, avec l'hydrodynamique des fluides.
- Il est logique que la compatibilité électrodynamique des réseaux, constitue une part significative d'un enseignement de l'électronique de forte puissance.

Le nombre des entrainements de forte puissance ($P > 20$ MW) dans l'industrie est limité mais significatif : pétrochimie (vapocraqueurs de 20 à 40 MW), lanceurs d'alternateurs pour les stations de pompage (de l'ordre de 25 MW), gros ventilateurs pour l'aéronautique (jusqu'à 50 MW), propulsion des navires (jusqu'à 25 MW)... En marge de la variation de vitesse, les gros redresseurs pour l'électrochimie (on atteint le GW pour l'alimentation des alumineries)... La mise en place de tels équipements est inconcevable sans une prise en compte rigoureuse des réseaux d'alimentation.

Nous avons évoqué l'amortissement dynamique des réseaux, bien qu'il soit encore peu utilisé. En fait, il ne l'est aujourd'hui que dans les cas de « petits réseaux » alimentés par des convertisseurs statiques, comme en traction ferroviaire pour l'alimentation statique des caténaires en 16 Hz 2/3, par exemple.

Le résonateur piézoélectrique pour l'électronique de puissance

EMILE. BIGOT

CEA Grenoble, 51D, D333, 17 rue des Martyrs - 38054 Grenoble
emile.bigot@cea.fr

Résumé : Cet article présente l'intérêt de l'emploi de résonateurs piézoélectriques dans le domaine de la conversion de puissance. Le modèle équivalent du résonateur piézoélectrique et les contraintes associées à son emploi seront d'abord présentés. Ensuite, une topologie convertisseur abaisseur de tension utilisant un résonateur piézoélectrique sera proposée et une simulation LTSpice sera mise en avant.

I/ Importance consacrée à la réduction des passifs

L'usage d'appareils électroniques a explosé ces 20 dernières années. Cette massification a été rendue possible par sa miniaturisation et la multiplication des fonctions par puces (« l'ère du microcontrôleur »). Au vu des progrès faits dans le domaine des transistors, la place prise par les composants passifs est de moins en moins négligeable.

De nos jours, de multiples niveaux de tension cohabitent au sein d'un même appareil pour permettre un fonctionnement optimal de chacun de ses composants. Il est alors nécessaire d'adapter les niveaux de tension entre charges et sources au moyen de convertisseurs de puissance. Pour leur bon fonctionnement, on utilise nécessairement des composants de stockage aussi bien sous forme magnétique (inductance) qu'électrostatique (condensateur). L'essentiel des améliorations portées ces dernières années à ces composants passifs ont été réalisées, d'une part sur leur packaging, avec une meilleure intégration sur les cartes électroniques, et d'autres part via la montée en fréquence, réduisant ainsi la quantité d'énergie à stocker sur une période de fonctionnement et donc le volume du composant.

La montée en fréquence est l'une des pistes les plus suivies, l'utilisation nouvelle des transistors de puissance dits « grand gap » (SiC, GaN) permet d'envisager des fonctionnements au mégahertz. On constate cependant que même si les performances des condensateurs et des transistors restent très bonnes en haute fréquence, celles des composants magnétiques dégringolent [1]. Ceci est imputé en partie aux problématiques d'effet de peau et de pertes fer. Une solution apportée en 2017 par l'équipe du laboratoire L2EP au CEA Grenoble est de remplacer les composants à stockage magnétique par une solution à stockage mécanique [2]. Pour ce faire, l'emploi de matériaux piézoélectriques est proposé. Ceux-ci permettent la conversion d'un champ électrostatique en une déformation et vice-versa. Cet article s'intéressera donc au développement du modèle mathématique approprié pour ces nouvelles topologies de

convertisseur, avant de développer une simulation sous LTSpice.

II/ Le résonateur piézoélectrique

II.1/ Modèle équivalent des résonateurs piézoélectriques et choix du matériau

Van-Dyke propose en 1928 un modèle électrique équivalent du résonateur piézoélectrique [3]. Celui-ci est composé de branches mécaniques RLC en parallèle d'un condensateur physique correspondant aux électrodes situées de part et d'autre du matériau piézoélectrique (Figure 1).

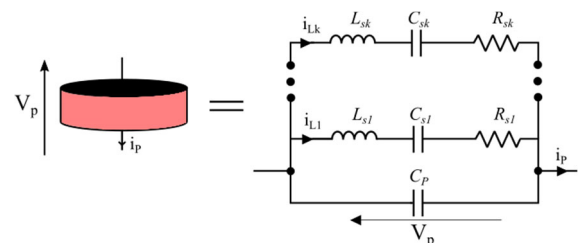


Figure 1: Modèle de Van-Dyke

Chacune des branches mécaniques correspond à un mode de résonance de l'élément piézoélectrique qui dépend directement de la géométrie de celui-ci. Erhart et al [4] ont résolu, pour les géométries les plus simples, les équations de la piézoélectricité. On peut alors déterminer de façon analytique la valeur de chacun des éléments du modèle équivalent.

De par sa géométrie, le cylindre court permet de facilement découpler deux modes propres de fonctionnement, l'un suivant l'épaisseur et l'autre suivant le diamètre. Pour simplifier l'étude nous considérerons uniquement le fonctionnement en mode épaisseur, en supposant que le diamètre et l'épaisseur sont d'ordres de grandeur suffisamment différents. Les valeurs des composants du modèle équivalent pour le mode épaisseur sont alors :

$$C_p = \frac{\varepsilon_{33}^S A_e}{2c} \quad (1.1)$$

$$C_{sh} = \frac{8C_p k_t^2}{h^2 \pi^2 - 8k_t^2} \quad (1.2)$$

$$L_{sh} = \frac{c^2 \rho}{2C_0 k_t^2 c_{33}^D} \quad (1.3)$$

$$w_{rh} = \frac{1}{\sqrt{L_{sh} C_{sh}}} \quad (1.4)$$

$$w_{arh} = \sqrt{\frac{C_{sh} + C_p}{C_p L_{sh} C_{sh}}} \quad (1.5)$$

$$R_{sh} = \frac{Q_m}{w_{rh} C_{sh}} \quad (1.6)$$

Avec A_e la surface d'électrode, c la demi-épaisseur du disque considéré, ϵ_{33}^S la permittivité diélectrique à contrainte constante, k_t le facteur de couplage du mode épaisseur piézoélectrique, ρ la densité du matériau, c_{33}^D le module d'élasticité à induction électrique constante, Q_m le facteur de qualité du matériau et w_{rh} la pulsation de résonance de la résonance LC mécanique.

L'objectif devient alors de diminuer les pertes au sein du matériau tout en préservant un fort taux de conversion électromécanique. Il a été montré que pour un dimensionnement favorisant la densité volumique de puissance transférable par notre composant, la figure de mérite $k^2 Q_m$ permet de classer efficacement chacun des matériaux [5]. Le PZT C-213 de FujiCeramic apparaît alors être le matériau PZT qui présente la figure de mérite la plus élevée [6].

Enfin, pour remplacer la fonction inductance d'un système, le résonateur piézoélectrique doit fonctionner entre les fréquences de résonance (f_r) et d'anti-résonance (f_{ar}) du mode considéré (voir Figure 2), là où son impédance possède une phase similaire à celle d'une inductance (+90°).

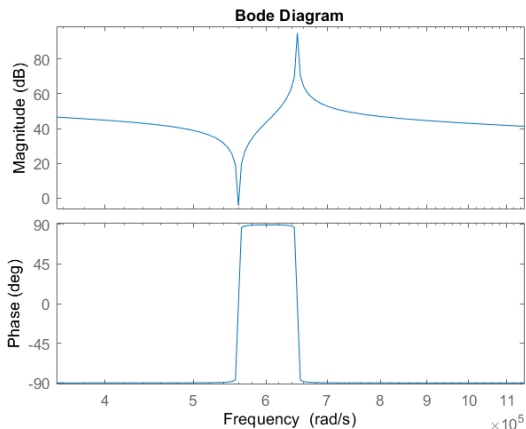


Figure 2: Diagramme de Bode du modèle équivalent de Van-Dyke pour un résonateur piézoélectrique. Entre les fréquences de résonance et d'anti-résonance la phase est de 90°, comme une inductance.

II.2/ Application sur un résonateur circulaire de 20x2mm

En supposant un résonateur piézoélectrique fait du matériau PZT-C213, d'après le constructeur Fujiceramic [7] les paramètres matériaux sont :

Module de Young à champ électrique fixe : Y_{11}^E	82 GPa
Module de Young à champ électrique fixe : Y_{33}^E	66 GPa
Coefficient de poisson : ν	0.29
Facteur de qualité mécanique : Q_m	2500
Densité de matériau : ρ	7800 kg/m ³
Constante de charge : d_{33}	310 pm/V
Constante de charge : d_{31}	-135 pm/V
Constante de charge : d_{15}	510 pm/V
Permittivité relative à contrainte fixe : ϵ_{33}^T	1470
Couplage 33 : k_{33}	70%
Couplage tangentiel : k_t	48%

Table 1: Paramètres constructeurs du PZT C-213

On remarque que les paramètres c_{33}^D et ϵ_{33}^S employés dans notre modèle analytique ne sont pas fournis. Pour les obtenir, nous nous appuyerons sur les équations de la piézoélectricité de [8].

Pour un disque polarisée suivant l'axe 3, les matrices des coefficients piézoélectriques (g_{ij} , d_{ij} , e_{ij} , h_{ij}) ont pour forme :

$$\mathbf{d} = \begin{pmatrix} 0 & 0 & 0 & 0 & d_{15} \\ 0 & 0 & 0 & d_{15} & 0 \\ d_{31} & d_{31} & d_{33} & 0 & 0 \end{pmatrix} \quad (2)$$

De plus, pour un matériau isotrope transverse (avec pour axe de révolution l'axe 3), la matrice de souplesse à champ électrique constant est :

$$\mathbf{S}^E = \begin{pmatrix} \frac{1}{Y_{11}^E} & -\frac{\nu}{Y_{11}^E} & -\frac{\nu}{Y_{33}^E} & 0 & 0 & 0 \\ -\frac{\nu}{Y_{11}^E} & \frac{1}{Y_{11}^E} & -\frac{\nu}{Y_{33}^E} & 0 & 0 & 0 \\ -\frac{\nu}{Y_{33}^E} & -\frac{\nu}{Y_{33}^E} & \frac{1}{Y_{33}^E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{Y_{44}^E} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{Y_{44}^E} & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(s_{11} - s_{12}) \end{pmatrix} \quad (3)$$

La matrice de rigidité \mathbf{C}^E est obtenue par inversion de \mathbf{S}^E . D'après [8], les paramètres piézoélectriques sont liés par les formules suivantes :

$$e_{ij}^S - \epsilon_{ij}^T = -e_{i\alpha} d_{j\alpha} \quad (4.1)$$

$$e_{i\alpha} = d_{i\beta} c_{\beta\alpha}^E \quad (4.2)$$

$$k_t = \frac{e_{33}}{\sqrt{c_{33}^D \epsilon_{33}^S}} \quad (4.3)$$

On détermine alors les relations ci-dessous :

$$e_{33} = (2d_{31} c_{13}^E + d_{33} c_{33}^E) \quad (5.1)$$

$$\epsilon_{33}^S = \epsilon_{33}^T - e_{33} d_{33} \quad (5.2)$$

$$c_{33}^D = \frac{e_{33}^2}{k_t^2 \epsilon_{33}^S} \quad (5.3)$$

$$c_{13}^E = \frac{1(v + \nu^2)}{\Delta} \frac{Y_{33}^E Y_{11}^E}{Y_{11}^E Y_{33}^E} \quad (5.4)$$

$$c_{33}^E = \frac{1(1 - \nu^2)}{\Delta} \frac{Y_{11}^E}{Y_{11}^E} \quad (5.5)$$

$$\Delta = \frac{1 - \nu^2}{Y_{11}^E Y_{33}^E} - \frac{2\nu^2 + 2\nu^3}{Y_{11}^E Y_{33}^E} \quad (5.6)$$

Finalement en considérant un disque résonant de 20mm de diamètre possédant une épaisseur de 2mm fait de PZT-C213, on trouve les paramètres équivalents suivants :

C0	1.26 nF
C	288 pF
L	94.9 μH
R	229 mΩ
f _r	962 kHz
f _{ar}	1.07 MHz

Table 2: Paramètres du modèle équivalent d'un résonateur piézoélectrique (20mm x 2mm) pour une excitation sur l'épaisseur à l'harmonique 1

II.3/ Contrainte portant sur les résonateurs piézoélectriques

Comme le montre la Figure 1, la capacité parallèle de notre composant empêche toute connexion brutale du résonateur à une source de tension, au risque de perdre une quantité significative d'énergie. Pour un fonctionnement à efficacité maximale, on laisse évoluer la tension à ses bornes de manière naturelle entre 2 connexions, de sorte à fonctionner en mode « Zero Voltage Switching » (ZVS). Le cycle consiste donc à alterner des phases de connexion pour permettre un échange de charges et des phases de déconnexion pour permettre l'évolution naturelle de la tension aux bornes du résonateur.

Sur un cycle de fonctionnement, la quantité totale de charges échangées avec le piézoélectrique doit être nulle afin d'éviter une dérive en tension de celui-ci. De même, le bilan énergétique sur le résonateur piézoélectrique doit être nul (aux pertes près) pour éviter une dérive sur l'amplitude des oscillations mécaniques (oscillations entretenues). Ainsi en notant Q_k la quantité de charge apportée pendant la phase de connexion k et V_k , la tension associée, les contraintes suivantes sont déduites :

$$\sum_k Q_k = 0 \quad (6.1)$$

$$\sum_k V_k Q_k = E_{PRLOSS} \quad (6.2)$$

III/ Etude d'un convertisseur abaisseur de tension

III.1/ Modèle analytique

Dans le cadre de cette article, nous étudierons la topologie proposée par Benjamin Pollet [9] pour la fonction abaisseur de tension (voir Figure 3). En supposant un facteur de qualité du matériau suffisamment élevé (>1000 en pratique), l'hypothèse

d'un courant traversant la branche mécanique de manière quasiment sinusoïdale est faite.

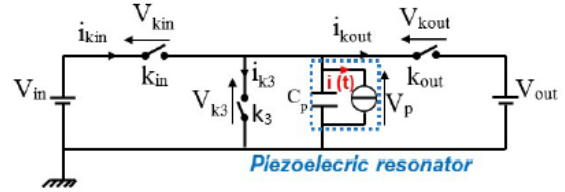


Figure 3: Convertisseur abaisseur de tension proposé par Benjamin Pollet [9]

Afin de respecter les contraintes des équations (6), tout en permettant un transfert d'énergie à la sortie, un cycle comprenant un minima de trois paliers de tension différents doit être appliqué. Au vu de la topologie employée, les paliers de tension (V_{in} , V_{out} , Zero) sont ici appliqués au résonateur piézoélectrique. Ces paliers de tension permettent au résonateur piézoélectrique d'échanger des charges avec la sortie, avec l'entrée et avec une tension nulle. Ce dernier palier sert à équilibrer la quantité de charges échangée sur une période.

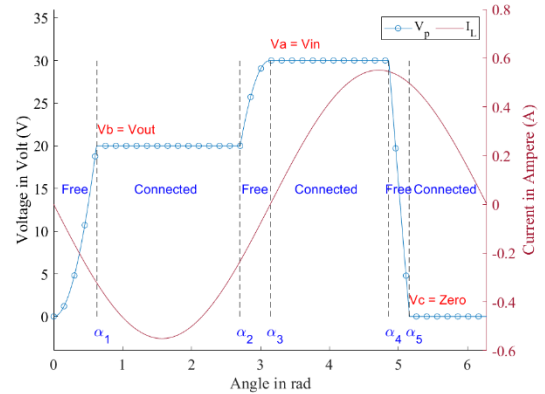


Figure 4: Cycle à trois paliers pour la conversion abaisseur de tension

Suivant les travaux de Mustapha Touhami [5], le taux de transfert de charges à la sortie peut être calculé. De même, nous pouvons déterminer la quantité de courant dit « circulant » nécessaire au maintien du fonctionnement en mode ZVS, ainsi que la part du courant utile au sein de la branche mécanique du résonateur. Les courants dans la branche mécanique s'expriment alors tel que :

$$i_L = -I_L \sin(\omega t) \quad (7.1)$$

$$I_L = I_{circul} + I_{useful} \quad (7.2)$$

$$I_{circul} = \frac{\omega C_p}{2} V_{in} \quad (7.3)$$

$$I_{useful} = \pi I_{out} \quad (7.4)$$

En raison des contraintes subies par le résonateur, le courant maximal admissible par celui-ci est limité. Il est de 1A/cm² à la fréquence de 1MHz et varie de l'ordre d'un facteur 5 par décade de fréquence d'après [5] pour le PZT C-213 précédemment choisi. En utilisant ce facteur de dimensionnement et suivant le taux de conversion et la puissance désirée, on peut donc calculer de manière complète les dimensions du résonateur piézoélectrique.

Enfin, en négligeant les capacités parasites des transistors et en supposant le courant traversant la branche mécanique sinusoïdal, les instants de fermeture des transistors respectant les contraintes d'équilibre énergétique et de charges sont calculés :

$$a_i = \cos(\alpha_i) \tag{8.1}$$

$$a_1 = \frac{(V_{out} - 0)\omega C_p}{I_L} - 1 \tag{8.2}$$

$$a_2 = \frac{-(V_{in} - V_{out})\omega C_p}{I_L} + 1 \tag{8.3}$$

$$a_3 = 1 \tag{8.4}$$

$$a_4 = a_3 - (a_2 - a_1) \frac{V_{out} - 0}{V_{in} - 0} + \frac{R_L I_L \pi}{V_{in}} \tag{8.5}$$

$$a_5 = (0 - V_{in}) * \frac{C_p \omega}{I_L} + a_4 \tag{8.6}$$

On remarquera que la fréquence de fonctionnement du système reste indéterminée. Pour le moment, aucune équipe n'a trouvé de manière élégante de la déterminer[10]. Cette variable reste donc inconnue et est pilotée tel que le fonctionnement se fasse en ZVS au niveau des interrupteurs à la puissance de fonctionnement désirée. Une solution employée est d'observer les instants d'annulation du courant et de les utiliser pour commander l'ouverture et la fermeture des interrupteurs, on peut alors parler « d'auto-pilotage de la fréquence ».

III.2/ Simulation sous LTSpice

Cette partie propose de simuler le convertisseur précédemment étudié au moyen de l'outil gratuit, LTSpice.

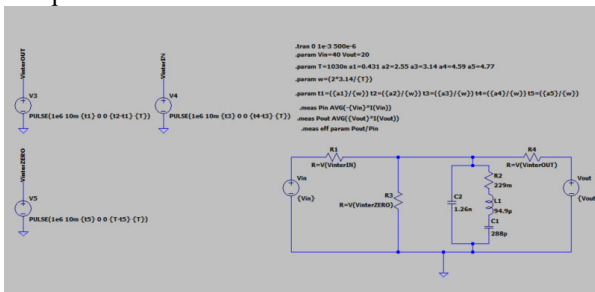


Figure 5 : Simulation d'un convertisseur à résonateur piézoélectrique sous LTSpice

Les paramètres du résonateur sont ceux déterminés dans la partie II, les instants de commutation sont calculés en utilisant l'approximation de la partie III.1 pour une fréquence donnée. Enfin, la fréquence de fonctionnement est ajustée jusqu'à obtenir les fermetures aux zéros de tension au niveau des interrupteurs. De même, les angles, « instants » de commutation sont aussi ajustés. Le tableau ci-dessous présente les angles calculés avec les ajustements et la fréquence de fonctionnement trouvée pour une puissance de sortie visée de 10W pour un abaissement de 40V vers 20V. Par souci de simplicité, les interrupteurs sont ici représentés par des résistances pilotées.

α_1	0.431 rad
α_2	2.71-0.16 rad
α_3	3.14 rad

α_4	4.59 rad
α_5	4.77 rad
f_{cons}	971 kHz

Table 3 : Paramètres calculés pour une conversion de 40V vers 20V, ajustements en rouge

Ci-dessous est présentée la tension obtenue aux bornes du piézoélectrique. On constate que le cycle à trois paliers est obtenu, et que la commutation à zéro de tension est réalisée.

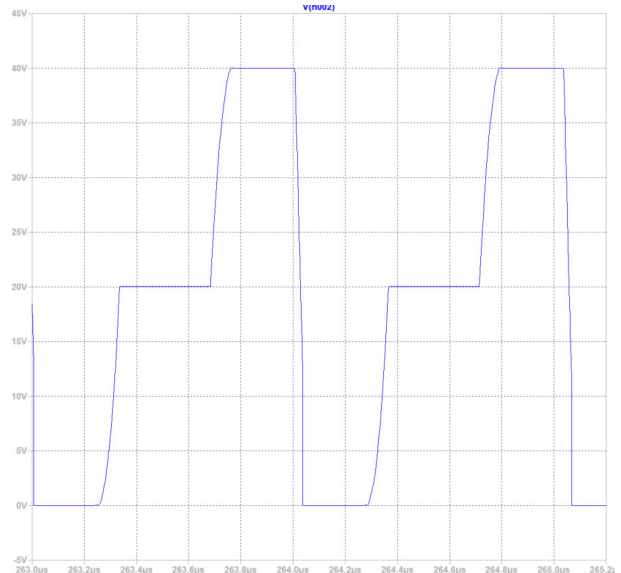


Figure 6 : Cycle simulé sous LTSpice

Enfin, la puissance en sortie du convertisseur est mesurée après un certain temps de fonctionnement (mise en service du résonateur). On constate qu'un rendement de 94.6% est obtenu, ceci confirme la possibilité de réaliser des convertisseurs à fort rendement et haute fréquence. Cependant, on constate que la puissance de sortie est quasiment deux fois inférieure à celle visée. Ceci montre bien les limites du modèle précédemment établi.

```
pin: AVG(- 40 *i(vin))=6.50727 |
pout: AVG( 20 *i(vout))=6.15996
eff: pout/pin=0.946628
```

Figure 7 : Puissance de sortie et rendement du convertisseur

IV/ Conclusion et perspectives

Il apparaît donc que les gains en termes d'efficacité et de miniaturisation induits par l'emploi de résonateurs piézoélectrique se font au détriment d'une commande simple. Cependant, en raison de la facilité de fabrication des éléments piézoélectriques et des difficultés rencontrées pour la montée en fréquence des éléments magnétiques, on peut espérer une introduction dans le futur de ce nouveau composant passif situé à cheval entre le condensateur et l'inductance.

Dans les réalisations les plus récentes, on notera l'élaboration de convertisseurs présentant des pics de rendements avoisinant les 98% pour une gamme de

puissance allant de la centaine de milliwatts à la centaine de watts.

Aujourd'hui, la recherche portant sur l'emploi de résonateurs piézoélectrique pour la conversion de puissance en est à ses débuts, de nombreux cycles et topologies sont envisagés et différents matériaux sont à l'étude.

v/ Bibliographie

- [1] C. R. Sullivan, B. A. Reese, A. L. F. Stein, et P. A. Kyaw, « On size and magnetics: Why small efficient power inductors are rare », in 2016 International Symposium on 3D Power Electronics Integration and Manufacturing (3D-PEIM), juin 2016, p. 1-23. doi: 10.1109/3DPEIM.2016.7570571.
- [2] A. Marzoughi, R. Burgos, et D. Boroyevich, « Optimum Design Guidelines for the Modular Multilevel Converter in Active Front-End Applications: Considerations for Passive Component Reduction », IEEE Power Electronics Magazine, vol. 5, no 2, p. 56-65, juin 2018, doi: 10.1109/MPPEL.2018.2821958.
- [3] K. Van Dyke, « The Piezo-Electric Resonator and Its Equivalent Network ». <https://xplqa30.ieee.org/document/1670053/>.
- [4] J. Erhart, P. Púlpán, et M. Pustka, « Piezoelectric Ceramic Resonators (Resonance Frequency and Equivalent Electrical Circuit) », in Piezoelectric Ceramic Resonators, J. Erhart, P. Púlpán, et M. Pustka, Éd. Cham: Springer International Publishing, 2017, p. 29-118. doi: 10.1007/978-3-319-42481-1_3.
- [5] M. Touhami et al., « Piezoelectric Materials for the DC-DC Converters Based on Piezoelectric Resonators », in 2021 IEEE 22nd Workshop on Control and Modelling of Power Electronics (COMPEL), nov. 2021, p. 1-8. doi: 10.1109/COMPEL52922.2021.9645999.
- [6] J. D. Boles, J. E. Bonavia, P. L. Acosta, Y. Ramadass, J. Lang, et D. J. Perreault, « Evaluating Piezoelectric Materials and Vibration Modes for Power Conversion », IEEE Transactions on Power Electronics, p. 1-1, 2021, doi: 10.1109/TPEL.2021.3114350.
- [7] « piezoelectric ceramics ». http://www.fujicera.co.jp/managed/wp-content/themes/fujicera/digitalbook/en/elements/index_h5.html#2.
- [8] J. Erhart, P. Púlpán, et M. Pustka, « Piezoelectricity and Piezoelectric Properties », in Piezoelectric Ceramic Resonators, J. Erhart, P. Púlpán, et M. Pustka, Éd. Cham: Springer International Publishing, 2017, p. 1-9. doi: 10.1007/978-3-319-42481-1_1.
- [9] B. Pollet, « Convertisseurs DC-DC piézoélectrique avec stockage provisoire d'énergie sous forme mécanique », phdthesis, Université Paris Saclay (COMUE), 2019. Consulté le: 23 février 2022. [En ligne]. Disponible sur: <https://tel.archives-ouvertes.fr/tel-02415837>
- [10] L. De Araujo Pereira, A. Morel, M. Touhami, T. Lamorelle, G. Despesse, et G. Pillonnet, « Operating Frequency Prediction of Piezoelectric DC-DC converters », IEEE Transactions on Power Electronics, 2021, doi: 10.1109/TPEL.2021.3115182.

Abonnez-vous à la

La REE est une publication trimestrielle de la SEE

REE

REVUE DE
L'ÉLECTRICITÉ
ET DE
L'ÉLECTRONIQUE

Choisissez votre formule d'abonnement pour 2022:



Version papier

5 numéros : mars, mai, juillet, octobre, décembre.
Distribution postale

France & UE	Hors UE
<input type="checkbox"/> 125 €TTC	<input type="checkbox"/> 142,43 €HT*



Version numérique

Accès aux publications numériques
ouvert pendant un an à compter de la date de paiement

France & UE	Hors UE
<input type="checkbox"/> 90 €TTC	<input type="checkbox"/> 88,15 €HT



Version duo

Version imprimée
+ version numérique

France & UE	Hors UE
<input type="checkbox"/> 155 €TTC	<input type="checkbox"/> 171,81 €HT*

Votre adhésion à la SEE

<input type="checkbox"/> Standard	<input type="checkbox"/> Retraité	<input type="checkbox"/> Enseignant	<input type="checkbox"/> Jeune actif (< 35 ans)	<input type="checkbox"/> Etudiant	<input type="checkbox"/> En recherche d'emploi
125 €TTC	65 €TTC			15 €TTC	

+ Votre abonnement REE (Tarif réservé aux adhérents, version papier)

France & UE: 63 €TTC Hors UE: 81,70 €HT*

TVA de la revue REE: 2,1%. Adhésion collective possible via des conventions de partenariat - Contactez-nous à : sg@see.asso.fr
* Prix final incluant des frais de transports de 20 €

Adresse de livraison

Nom* :

Prénom* :

Adresse* :

Code postal* : Pays* :

Ville* :

Tél.* :

e-mail* :

*Obligatoire

Adresse de facturation (Si différente)

Je joins le bon de commande administratif N°
et je désire recevoir une facture au nom de mon employeur pour paiement à réception

Raison sociale de l'employeur :

Service : Activité (facultatif) :

Adresse :

Code postal : Ville :

Pays : N° TVA :

N° TVA intracommunautaire : obligatoire pour règlement HT en UE hors de France

Votre règlement

Je règle la somme de €

par Chèque à l'ordre de la SEE
 Virement après réception de la facture
 Carte bancaire (Visa, Eurocard/Mastercard)

N° Carte

Date de validité N° cryptogramme (3 derniers chiffres au dos de la carte)

e-mail* :

Date* Signature* et cachet si il y a lieu :

*Obligatoire

BULLETIN À COMPLÉTER ET RENVOYER À : SEE- 17 rue de l'Amiral Hamelin - 75116 Paris - France
Tél. +33(0)156 90 37 17- abo@see.asso.fr

ABONNEMENT PLUS RAPIDE : www.see.asso.fr

Je consens à recevoir les autres diffusions de la SEE & de ses activités (congrès, soirées débats, revues, etc.) qui sont extérieures aux diffusions liées à mon abonnement.

Conformément aux dispositions légales et réglementaires en matière de données personnelles, les informations recueillies sur ce formulaire sont enregistrées dans un fichier informatisé par la SEE (Société de l'électricité, de l'électronique et des technologies de l'information et de la communication) pour la mise en place et le suivi de l'abonnement souscrit ainsi que pour l'envoi de courriers, e-mails de dé Abonnements. Elles sont conservées et sont destinées à être utilisées par la SEE et les prestataires techniques de la SEE afin de permettre la bonne réception du magazine et d'assurer le service client. Vous pouvez exercer votre droit d'accès aux données vous concernant par courrier : SEE- Service abonnements 17 rue de l'Amiral Hamelin 75116 Paris ou par le formulaire de contact du site web : www.see.asso.fr. Offre valable du 01/10/2021 au 30/09/2022 dans la limite des quantités disponibles.



