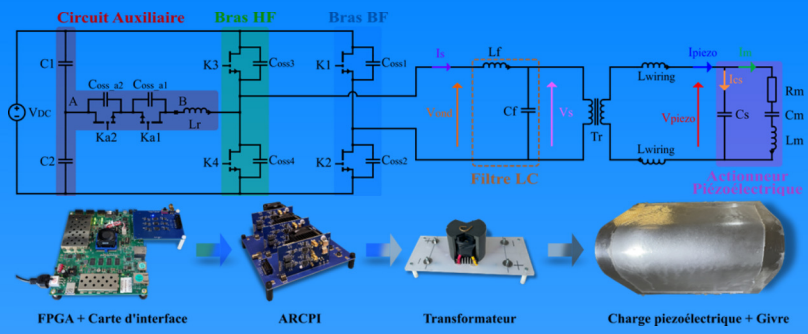
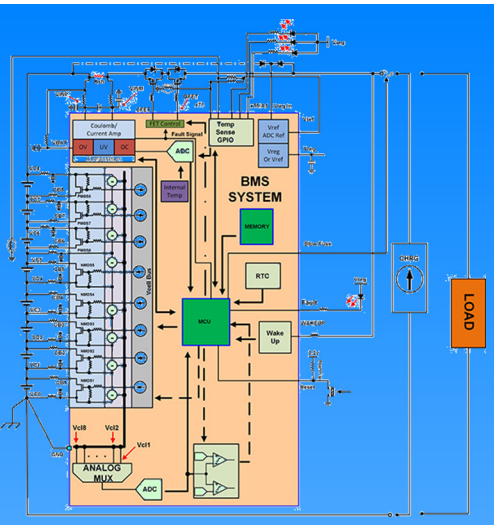
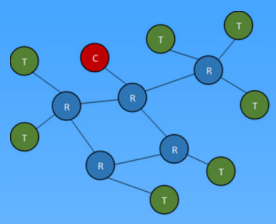
N° 112 – 2e trim. 2024
29ème année

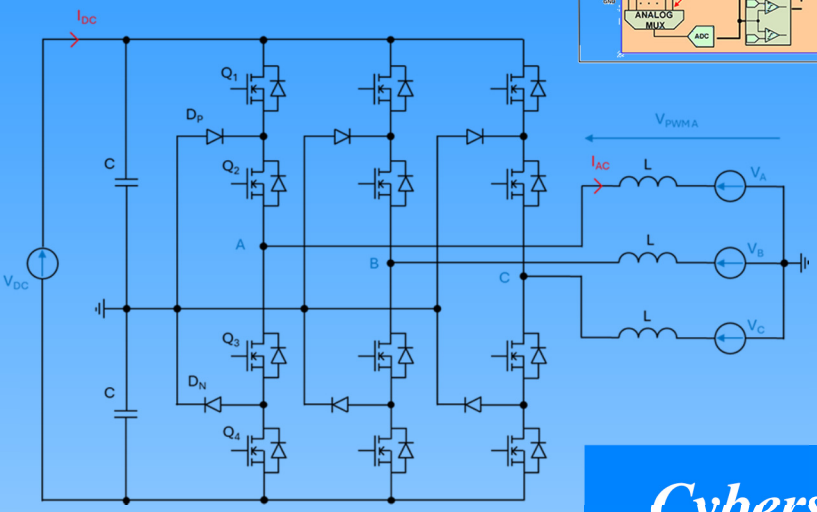
La Revue 3E.I

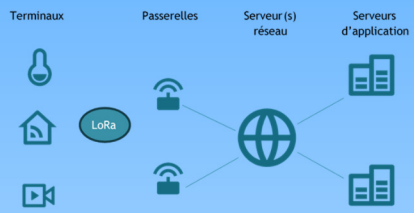
Ressource publiée sur Culture Sciences de l'Ingénieur : https://eduscol.education.fr/sti/si-ens-paris-saclay

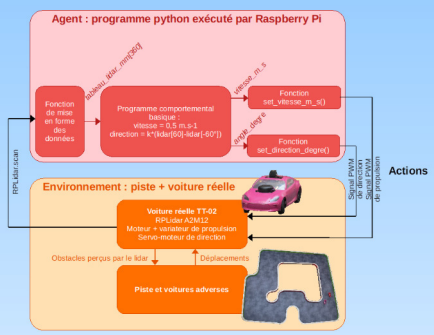


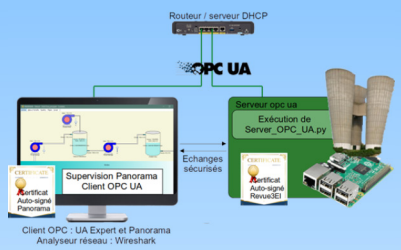












Cybersécurité des systèmes industriels - Électronique de puissance

2/2

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

since 1883

N°112
2ème trimestre 2024

Cybersécurité des systèmes industriels - Électronique de puissance (2/2)

Abonnez-vous à la REE

REVUE DE L'ÉLECTRICITÉ ET DE L'ÉLECTRONIQUE

La REE est une publication trimestrielle de la SEE

Choisissez votre formule d'abonnement :

❑ Version papier

1 an - 4 numéros**

(Parution : mars, juin, octobre, décembre). Distribution postale

Livraison France	Livraison UE	Livraison Hors UE
❑ 135 € TTC	❑ 150 TTC (146,91 € HT*)	❑ 165 TTC (161,61 € HT*)

❑ Version numérique

Accès aux publications numériques

ouvert pendant un an à compter de la date de paiement

France - UE - Hors UE
❑ 90 € TTC (88,15 € HT*)

❑ Version duo

Version imprimée + version numérique

Livraison France	Livraison UE	Livraison Hors UE
❑ 165 € TTC	❑ 180 € TTC (173,30 € HT*)	❑ 195 € TTC (190,99 € HT*)

** Prix HT valide si le pays de facturation est hors UE, ou si la TVA Intracommunautaire est fournie pour un pays de l'UE.*
*** Abonnement papier servi à partir de la date de paiement*

Votre adhésion à la SEE* pour 2024

❑ Standard	❑ Retraité ❑ Enseignant ❑ Jeune actif (< 35 ans)	❑ Etudiant ❑ En recherche d'emploi
130 €	70 €	15 €

** Adhésion d'un an à compter de la date de paiement.*

+ Votre abonnement REE** *(Tarif réservé aux adhérents, version papier)*

❑ Livraison France : 68 € TTC ❑ Livraison UE : 78 € TTC (76,40 € HT*) ❑ Livraison Hors UE : 83 € TTC (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 HT valide si le pays de facturation est hors UE, ou si la TVA Intracommunautaire est fournie pour un pays de l'UE.*
*** Abonnement papier servi à partir de la date de paiement.*

Adresse de livraison

❑ Mr ❑ Mme Prénom* :

Nom* :

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

e-mail* :

Date* Signature* et cachet si il y a lieu :

**Obligatoire*

Votre réglement

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)*

BULLETIN À COMPLÉTER ET RENVOYER À : SEE - 17 rue de l'Amiral Hamelin - 75116 Paris - France
Tél. +33(0)1 56 90 37 17 - abo@see.asso.fr

ABONNEMENT PLUS RAPIDE EN LIGNE : www.see.asso.fr

see since 1883

❑ 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 d'abonnement, valable du 01/12/2023 au 25/11/2024 inclus, dans la limite des quantités disponibles.

La Revue 3E.I
*Publication trimestrielle de la **SEE***

SOCIETE 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
www.see.asso.fr

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

Hébergé par :

Culture Sciences de l'Ingénieur

4 avenue des Sciences, 91190 Gif sur Yvette
tel : 01 81 87 55 22
https://eduscol.education.fr/sti/si-ens-paris-saclay

La Revue 3E.I

3EI : Enseigner l'Electrotechnique et l'Electronique Industrielle

La Revue 3EI, Édition SEE,
17 rue de l'Amiral Hamelin
75116 PARIS

Directeur de la publication
François GERIN
Président de la SEE

Rédacteur en Chef
Franck LE GALL

Adresser les propositions d'article à :
revue3ei@gmail.com

Communication :
Mme. Mélisande DE LASSENCE
Communication1@see.asso.fr
01 56 90 37 17

Dépôt Légal : juin 2024

Commission Paritaire 1222 G 78028

ISSN 1252-770X

Sommaire du n° 112

p. 2 *Éditorial*

Thème : Cybersécurité des systèmes industriels (partie 2)

Cybersécurité des systèmes automatisés industriels

p. 4 *Louis Lalay, « OPC UA, un protocole sécurisé pour l'automatisme industriel Mise en œuvre d'un serveur OPC UA sécurisé et de sa supervision »*

Cybersécurité des objets connectés

p.27 *Maxime Sechehaye, « Sécurité du protocole LoRaWAN »*

p. 34 *Maxime Sechehaye, « Sécurité de ZigBee »*

Thème : Innovations en cours en électronique de puissance (partie 2)

p. 41 *Modar Jomaa et al., « Choix d'une topologie de conversion adaptée à un système de dégivrage piézoélectrique »*

p. 53 *Adrien Voldoire, « Conception des onduleurs de tension : Comparaison entre une structure classique et une structure multiniveau NPC »*

Hors Thème :

p. 71 *Rania Bennami et al., « Apprentissage par renforcement et transfert simulation vers réalité pour la conduite de voitures autonomes »*

p. 101 *Arnaud Sivert et al. « Gestion de la Charge de Batteries Lithium (BMS) »*

Editorial

Cybersécurité des systèmes industriels et innovations en électronique de puissance

Vous avez été nombreux à consulter le numéro 111 de la revue 3EI sur les sites de Culture Sciences de l'Ingénieur et de la SEE, saluant ainsi sa nouvelle version numérique. Le comité de rédaction se réjouit de ce succès et souhaite que cette dynamique s'amplifie avec l'aide du plus grand nombre. N'hésitez-pas à proposer vos contributions et à diffuser la revue 3EI au public le plus large, par exemple, sur/avec la liste de diffusion renater : https://groupes.renater.fr/sympa/info/revue3ei

Dans ce second numéro de l'année 2024, nous vous proposons d'approfondir les thèmes de la « cybersécurité des systèmes industriels » et des « innovations en électronique de puissance ».

« Dossier : Cybersécurité des systèmes industriels »

Après une première introduction à la cybersécurité des systèmes industriels proposée à l'ouverture de ce dossier, nous nous concentrons ici sur une application d'automatisme industriel avec OPC UA et sur la sécurité des objets connectés avec une attention particulière portée sur les protocoles utilisés pour le LoRaWAN et le ZigBee

L'article d'application pratique de OPC UA est proposé par Louis Lalay et Anthony Juton. Il décrit la simulation d'un système de château d'eau dialoguant en OPC UA avec le superviseur. Cette étude réalisée à l'aide d'un nano-ordinateur Raspberry Pi 4 bon marché peut aisément être utilisée en TP et permet l'étude d'OPC UA en tant que protocole sécurisé d'automatisme industriel.

Les trois articles sur la sécurité des objets connectés, écrits par Maxime Seychehay, permettent de bien comprendre quels sont les outils mis à disposition pour sécuriser les échanges de données.
Dans les articles sur le LoRaWAN et sur le ZigBee l'auteur insiste sur les choix importants qu'un développeur d'application IoT doit faire pour assurer la sécurité de son application.

Il y aura un dernier volet cybersécurité au troisième trimestre 2024, avec notamment une application pédagogique bluetooth Low Energy sécurisée, toujours écrite par Maxime Seychehay. Il serait intéressant que les personnes ayant des applications cybersécurité à partager proposent leur sujet.

« Dossier : Innovations en cours en électronique de puissance »

Après la série d'articles d'introduction publiés dans le numéro 111, nous vous proposons ici deux articles d'approfondissement sur le thème de l'électronique de puissance.

L'article de Modar Jomaa et de son équipe du Satie, décrit l'application de l'électronique de puissance au problème du dégivrage des ailes d'avion. La solution de dégivrage piézoélectrique proposée dans cet article semble être une alternative efficace et plus économique en termes de coût, masse et encombrement que les technologies traditionnelles. Plusieurs topologies de convertisseurs statiques sont présentées et simulées. Un démonstrateur de la solution retenue a été développé pour valider le choix proposé.

L'article d'Adrien Voldoire permet d'approfondir nos connaissances sur la conception des onduleurs de tension, qu'ils soient utilisés pour l'entrainement de machines triphasées ou pour la connexion à des réseaux de distribution ou embarqués. L'analyse comparative de trois dimensionnements permet de comprendre les enjeux sous-tendant le dimensionnement des onduleurs de tension et l'usage de structures multiniveaux.

« Hors Thème »

Les lecteurs des articles sur les voitures autonomes (CoVaPSy) seront heureux de découvrir le texte de Rania Bennami et de son équipe. Ce texte présente la voiture 1/10ème instrumentée qui permet d'appréhender le transfert « simulation → réalité » et les difficultés associées pour une mise en œuvre concrète et matérielle de l'intelligence artificielle.

Nous refermons ce numéro avec la publication de l'équipe d'Arnaud Sivert sur la gestion de la charge des batteries au Lithium. Cet article basé sur des simulations permet de se faire une idée claire des performances des chargeurs et des améliorations que l'on peut en attendre quant à l'équilibrage des cellules ainsi que la gestion des paramètres que sont l'intensité du courant de charge, la tension d'alimentation du chargeur et la température de fonctionnement.

OPC UA, un protocole sécurisé pour l'automatisme industriel

Mise en œuvre d'un serveur OPC UA sécurisé et de sa supervision

La Revue 3E.I
Culture Sciences de l'Ingénieur

Édité le 05/06/2024

école normale supérieure paris-saclay

Louis LALAY[1] – Anthony JUTON[2]

[1] *Étudiant agrégé de sciences de l'ingénieur, doctorant à Télécom Paris*

[2] *Professeur agrégé, ENS Paris Saclay, DER Nikola Tesla*

Cette ressource fait partie du N°112 de La Revue 3EI du 2ème trimestre 2024.

L'objectif de cette ressource est de présenter une application pratique de OPC UA, utilisant un nano-ordinateur raspberry Pi 4 bon marché (90 euros), pour simuler un système de château d'eau dialoguant en OPC UA avec le superviseur. Ce dispositif facile à dupliquer peut alors être un support pour des travaux pratiques autour du protocole OPC UA (supervision, étude des mécanismes de sécurisation du protocole). L'objectif étant l'étude d'OPC UA en tant que protocole sécurisé d'automatisme industriel, nous avons limité au maximum le besoin de connaissance de Linux et de python.

En cas de difficulté ou d'éléments peu clairs dans la ressource, il est possible d'échanger sur la liste https://groupes.renater.fr/sympa/info/revue3ei *pour résoudre les problèmes et améliorer la ressource.*

Pour travailler sur une seule machine (ce qui nous semble moins pédagogique), il est possible de remplacer le nano-ordinateur au choix :

- Par une machine virtuelle (de préférence Linux pour pouvoir utiliser OpenSSL),
- En exécutant les scripts python directement sur le système d'exploitation Windows où fonctionne Panorama

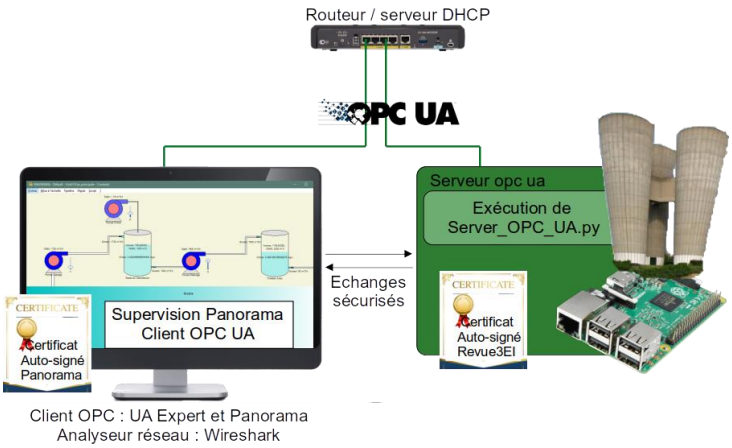


Figure 1 : Schéma synoptique de l'installation simulée présentée dans cette ressource

La ressource commence par une rapide présentation d'OPC UA et de son fonctionnement pour pouvoir aborder ensuite la mise en œuvre dans le cadre de l'activité pratique présentée. Cette présentation peut être avantageusement complétée par les vidéos d'Hervé Discours [6]. Le lecteur pourra ensuite approfondir avec les supports mis à disposition par OPC Uacademic [2].

1 - Présentation de OPC UA

1.1 - Histoire de la fondation [1]

Dans les années 90, Microsoft COM et DCOM[1] dominent la communication industrielle. En 1995, Fisher-Rosemount, Intellution, Opto 22 et Rockwell Software s'associent pour créer un standard de communication basé sur COM et DCOM nommé OPC, raccourcis de OLE[2] for Process Control. La fondation OPC (opcfoundation.org) est officiellement créée en 1996. Fin 1996, OPC-DA[3], une version simplifiée des spécifications OPC voit le jour : c'est la version qui a rendu OPC populaire. Cette version se base sur des protocoles Microsoft, et n'est donc supportée que par Windows.

En 2006, OPC UA[4] est créé afin de devenir indépendant de la plateforme hôte. Cette version est donc compatible avec plus de services. L'ancienne version est alors appelée OPC Classic.

Depuis, la fondation a surtout mis à jour ces deux protocoles et créé des liens avec différents industriels. En effet, un organisme peut intégrer la fondation pour faciliter la mise en œuvre des spécifications OPC à son propre matériel. La fondation OPC compte actuellement 850 membres, dont tous les acteurs importants de l'automatisme industriel (Siemens, Rockwell Automation, Wago, B&R), de la supervision (Arc Informatique, Codra, …), de la robotique industrielle (Fanuc, Staubli, Omron, …), de la production d'électricité (ABB, Alstom, General Electric, Schneider Electric, …), et même des fabricants de composants (Microchip, ST Microelectronics, ...)

Aujourd'hui, OPC signifie Open Plateform Communications. Dans la suite, on s'intéressera uniquement à OPC UA, dans un contexte multiplateforme et sécurisé.

1.2 - Domaines d'utilisation principaux

Le protocole OPC UA est surtout utilisé en automatisme, pour la communication entre un superviseur et des automates, ou pour la communication des automates entre eux. OPC UA permet la communication sécurisée de données, avec une multitude de services utiles à la supervision d'installations industrielles, par exemple :

- Alarmes,
- Horodatage,
- Historisation,
- etc.

Intérêt majeur de OPC UA par rapport aux protocoles de supervision traditionnels (Modbus TCP, BACnet Profinet, …), la communication se fait au travers de sessions ouvertes entre des clients et des serveurs et sécurisées par le populaire protocole SSL. Le client (un logiciel de supervision par exemple) envoie des requêtes aux serveurs (des automates par exemple) pour obtenir des informations du processus en cours et/ou pour donner des consignes afin de modifier ce processus.

[1] Distributed Component Object Model
[2] Microsoft Object Linking & Embedding
[3] OPC for Data Acces
[4] OPC Unified Architecture

1.3 - Rôle de la fondation OPC

Figure 2 : Logo de la fondation OPC

La fondation OPC est en charge du développement et du maintien des spécifications du protocole. C'est ensuite aux constructeurs de mettre en œuvre ces spécifications pour créer un serveur OPC UA ou un client OPC UA. Par exemple, Siemens implémente un serveur OPC UA sur ses automates S7-1200. De la même manière, Codra, entreprise développant le logiciel de supervision Panorama, implémente un client OPC UA sur Panorama, ce qui permet ainsi la supervision sécurisée des automates Siemens. Siemens et Codra faisant partie de la fondation OPC, leurs implémentations du protocole est validée par la fondation.

2 - Fonctionnement de OPC UA

Cette partie présente différentes possibilités offertes par OPC UA et leur fonctionnement.

2.1 - Session de communication

On s'intéresse ici à la structure de la communication OPC UA, le support utilisé, les différentes couches et leur rôle.

OPC UA utilise les couches TCP/IP (couches 4 et 3) sur des couches liaison de données / physique Ethernet ou Wifi (couches 2 et 1). Une fois la connexion TCP établie, une session OPC UA est ouverte au niveau de la couche 5 (voir Figure 3). Les interactions entre les clients et les serveurs requièrent un modèle à état. Les informations de l'état sont définies dans la session, qui est une connexion logique entre les clients et les serveurs. Les sessions sont indépendantes des protocoles de communication sous-jacents, et ne sont fermées que sur une requête de fermeture ou l'inactivité d'un client. Dans une session, on retrouve par exemple des détails sur les utilisateurs (nom d'utilisateur, mot de passe, autorisations, etc.), le mode de communication (Publisher/Subscriber ou Request/Response), etc.

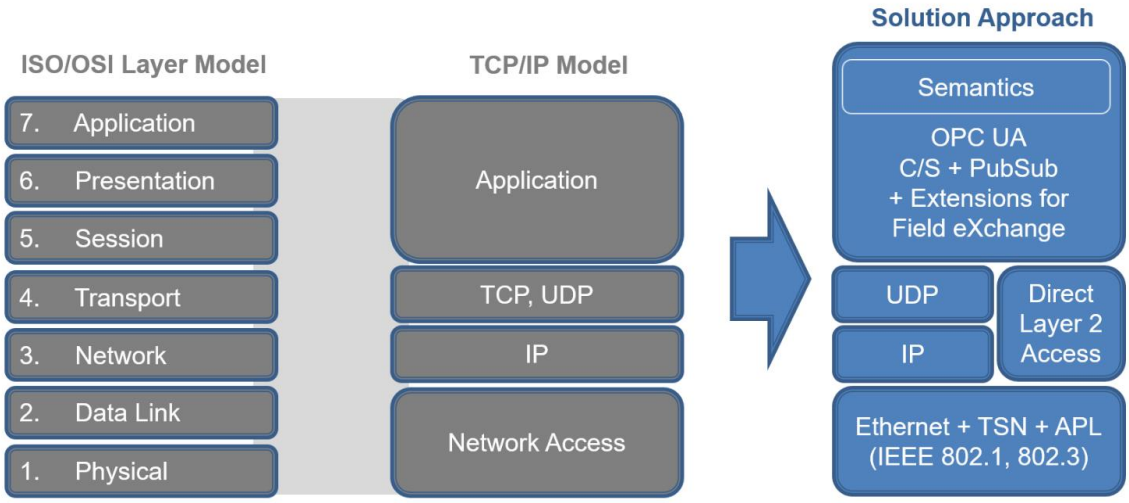


Figure 3 : Couches réseaux OPC UA (source OPC UAcademic)

2.2 - Sécurité

La sécurité est un des points forts de OPC UA, comparé aux anciens protocoles de supervision (Modbus par exemple). La sécurisation des communications en OPC UA s'appuie sur OpenSSL, comme HTTPs (cf ressource [13]).

Le service de sécurisation est indépendant du fonctionnement de l'application OPC UA, ce service de sécurisation reposant sur la *Communication Stack*. La *Communication Stack* communique via un Secure Channel. Un Secure Channel est une connexion logique longue durée entre un client et un serveur. Il est mis en place par un échange de certificats X.509 (client et serveur) et de clés publiques (clé client et clé serveur) aboutissant à un échange de clés de chiffrement symétrique permettant le chiffrement des échanges suivants, jusqu'à la fermeture du Secure Channel. Une application OPC UA ignore tout message qui ne suit pas la politique de sécurité. Une session est associée à un unique *Secure Channel*. Lors d'une connexion, le *Secure Channel* est d'abord ouvert, puis la session est ensuite ouverte en utilisant ce canal sécurisé, comme cela sera présenté dans la partie pratique un peu plus loin dans cette ressource.

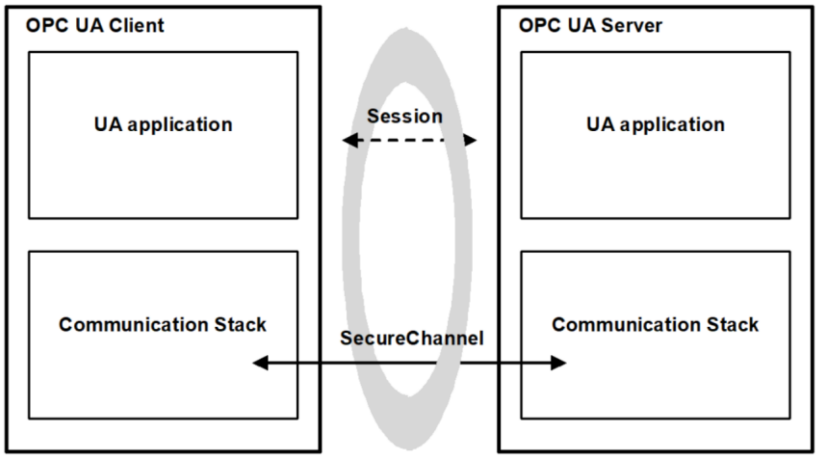


Figure 4 : Sécurité (source opcfoundation.org)

Pour les échanges de données via le *Secure Channel*, il existe 3 niveaux de sécurité :

- Aucun
- Avec chiffrement
- Avec chiffrement et authentification

2.3 - Discovery

Le service *Discovery* est décrit en détail dans la Partie 12.4 de la référence OPC UA. Ce service permet aux applications OPC UA de rechercher d'autres applications. En général, ce sont les serveurs qui proposent ce service afin que des clients s'y connectent, mais certains clients peuvent avoir une connexion inversée.

Avant même de créer une session, le service de *Discovery* permet d'identifier un serveur. Cela permet aussi d'avoir une vue d'ensemble de l'application, avec par exemple les variables qu'elle propose. Les variables étant désignées par un nom explicite, le développeur évite ainsi les erreurs d'adressage ou d'unité, typiques de modbus (on y désigne l'adresse de la variable et on récupère sa valeur brute, sans unité).

2.4 - Espace d'adresses

Une fois que la connexion est établie et sécurisée, il est possible d'accéder à l'espace d'adresses et de commencer à échanger des messages.

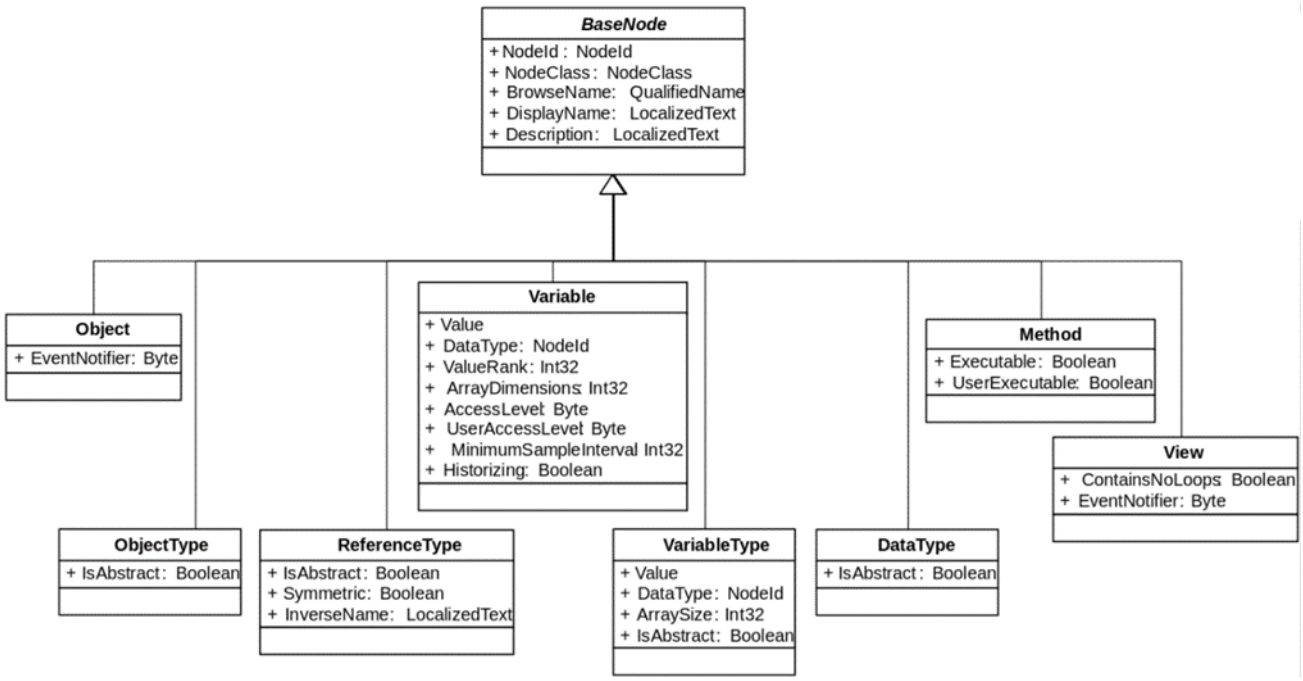


Figure 5 : Address Space (Source OPC UAcademic)

L'espace d'adresses définit comment est organisée une application OPC UA. C'est ici que se trouvent les variables par exemple. Chaque élément de l'espace d'adresse est un nœud, qui peut être une des 8 classes présentées sur la figure 5 ci-dessus et définies comme suit :

- **Object** : Utilisé pour représenter des systèmes, des composants, des objets réels, etc. Les objets sont liés entre eux par des références ;
- **ObjectType** : Définition pour les objets ;
- **ReferenceType** : Définition pour les références ;
- **Variable** : Stocke des données pour un objet ;
- **VariableType** : Définition pour les variables ;
- **DataType** : Type pour les données ;
- **Method** : Fonction ;
- **View** : Définit un sous ensemble de l'espace d'adresses. La vue par défaut est l'espace d'adresses en entier.

Un exemple d'espace d'adresses simple est proposé sur la figure 6. La racine de l'espace d'adresse est l'URL du serveur. On a ensuite (à la fin de la liste ici) un objet présentant les informations relatives au serveur (mode de communication, type de sécurité, etc.) puis les objets liés au process, avec des valeurs associées. L'espace d'adresses se construit à partir de l'installation réelle, en encapsulant les variables dans des objets. On peut ensuite encapsuler les objets dans d'autres objets plus grands. Les *View* permettent ensuite de sélectionner une partie de l'arborescence.

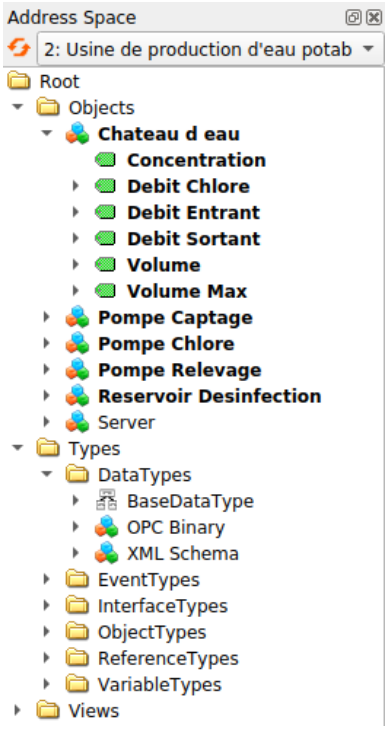


Figure 6 : Espace d'adresse de l'application château d'eau

2.5 - Modes de communication

Il existe deux modes de communication entre les applications. Les deux formes peuvent être adoptées et vont dépendre des cas d'utilisation. Dans les deux types de communication, les variables sont horodatées. La figure 7 présente les deux modes de communication.

Client/Server : Ce mode de communication est établi entre un client et un serveur. Lors de l'ouverture de la session, le serveur met en place un abonnement du client sur les variables qui l'intéressent. La session restant ouverte, le client n'est notifié que lorsque les valeurs ont changé. Le temps de rafraîchissement des valeurs côté serveur est réglable.

Publication/Subscription : Ce mode de communication met en relation un éditeur et des abonnés. Les applications n'échangent pas les messages directement, mais passent par un intermédiaire. Les *Publishers* envoient leurs données à l'intermédiaire sans savoir s'il y a des abonnés. De leur côté, les abonnés signalent à l'intermédiaire qu'ils sont intéressés par une donnée, sans savoir si la donnée est encore mise à jour. Lorsqu'une donnée est modifiée (et uniquement lorsqu'elle est modifiée), les abonnés sont notifiés du changement et peuvent récupérer la donnée à jour. Cela permet à plusieurs clients de s'abonner aux mêmes données sans ouvrir de session supplémentaire.

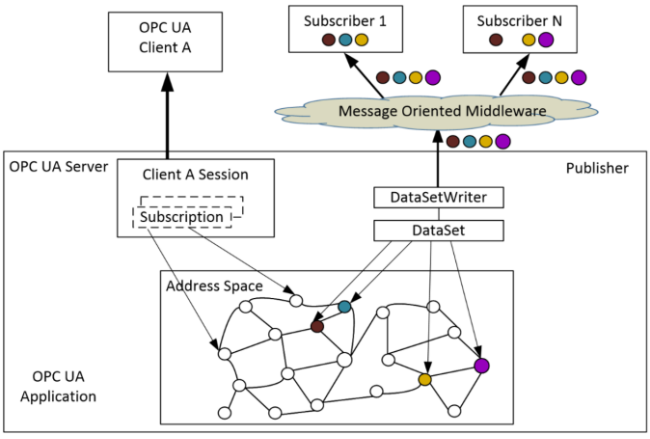


Figure 7 : Modes de communication (source Partie 1 6.6)

2.6 - Alarmes

La Partie 9 de la référence OPC UA explicite le fonctionnement des alarmes dans leur détail.

Les applications OPC UA permettent la création d'alarmes, et donc de simplifier la supervision de processus. Une alarme se déclenche par exemple lorsqu'une mesure sort d'une plage de valeur définie et alerte l'opérateur (via le logiciel de supervision).

L'application présentée ici en exemple gagnerait à utiliser les alarmes pour indiquer par exemple un débordement du château d'eau.

3 - Mise en œuvre du serveur OPC UA / château d'eau

Dans cette partie, on s'intéresse à la mise en œuvre d'un serveur OPC UA simulant un château d'eau, en 3 étapes.

- Démarrage d'un serveur OPC UA FreeOpcUa, qui implémente une grande partie des fonctionnalités OPC UA en langage Python (et en C++, inutilisé ici). Il n'est pas nécessaire de maîtriser python pour mettre en place ce qui suit. Le Client OPC UA est le logiciel gratuit UAExpert.
- Sécurisation de la connexion, toujours avec le serveur OPC UA FreeOpcUa et le client UAExpert
- Supervision avec le logiciel Panorama, client OPC UA, du serveur OPC UA simulant un château d'eau. Ce serveur utilise là encore FreeOpcUa.

3.1 - OPC UA avec Python

FreeOpcUa est un projet open source et ne fournit donc aucune garantie, mais il est très complet pour en faire un projet de démonstration sur un nano-ordinateur comme une carte Raspberry Pi. FreeOpcUa implémente les fonctionnalités essentielles de OPC UA, notamment la sécurisation des échanges.

3.2 - Installation du serveur

On configure un serveur OPC UA en langage Python. L'objectif est d'implémenter les fonctionnalités de base du serveur, afin de pouvoir observer les trames avec des outils tels que Wireshark, pour mettre en évidence les échanges, non sécurisés pour l'instant, via OPC UA.

Installation du système d'exploitation sur le nano-ordinateur

Pour installer le système d'exploitation Raspberry OS sur le nano-ordinateur raspberry Pi 4 et l'accès à distance, on peut suivre le document référencé ici [8]. On obtient ainsi un nano-ordinateur avec un système d'exploitation linux dont le bureau à distance est accessible via VNC Viewer. Il est aussi possible d'utiliser une machine virtuelle ou de faire tourner le serveur directement sur l'OS du client.

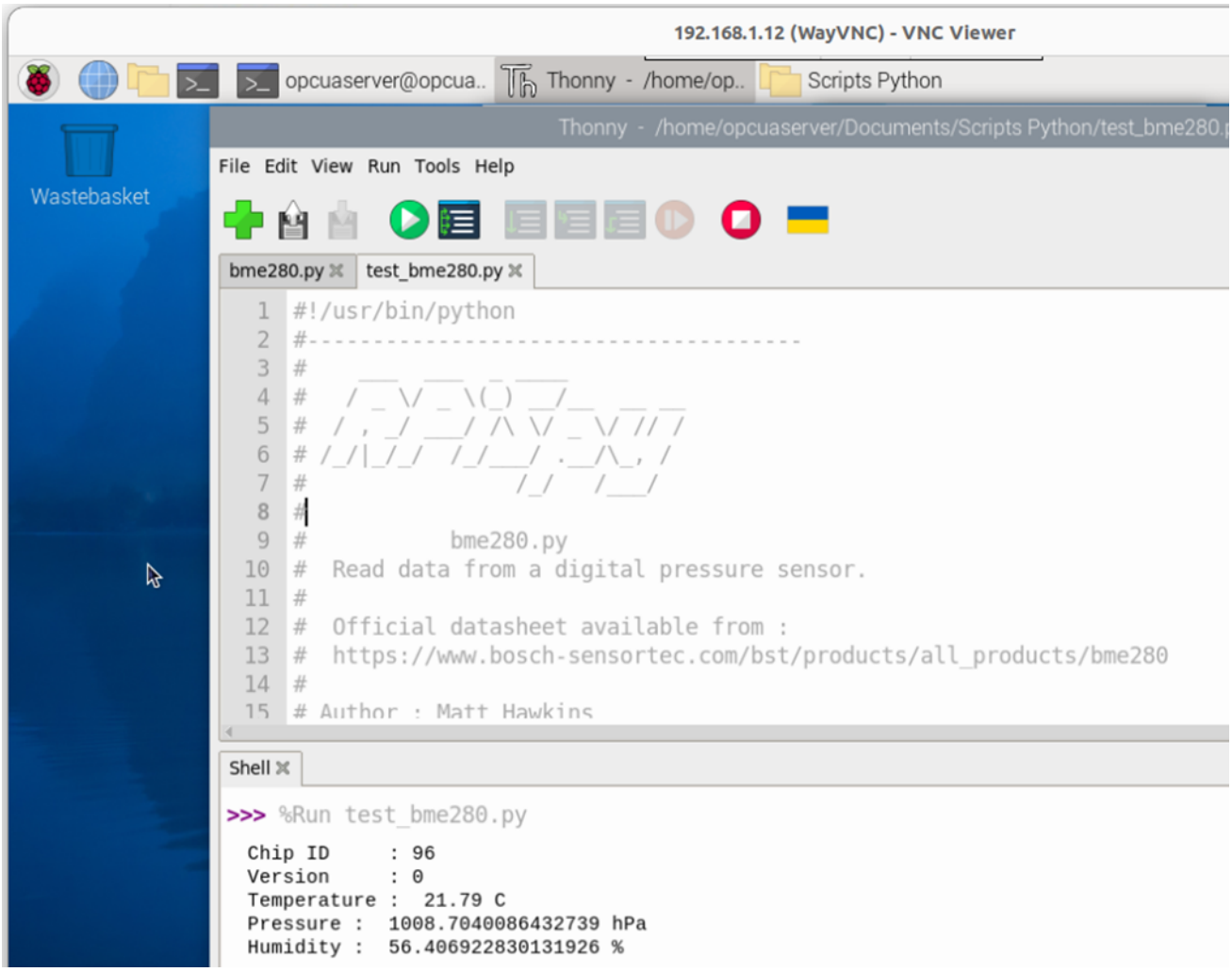


Figure 8 : Exécution d'un programme test_bme280.py via l'accès au bureau à distance du nano-ordinateur Raspberry Pi 4 par VNC Viewer

Installation du module serveur OPC UA

On installe tout d'abord le module python Free OPC-UA nommé **asyncua**. Le nano-ordinateur ne servant que de serveur OPC UA, on ne s'encombre pas du système d'environnements virtuels python, d'où l'option --break-system-packages.

```
pip install asyncua --break-system-packages
```

3.3 - Test simple du serveur OPC UA

Pour commencer, on teste sur une configuration simple :

- Pas de sécurité sur le serveur (prog *test_server_OPC_UA.py*)
- Juste une variable qui s'incrémente sur le serveur,
- Un client tout prêt : UA Expert,
- L'analyseur de réseau Wireshark pour observer les échanges.

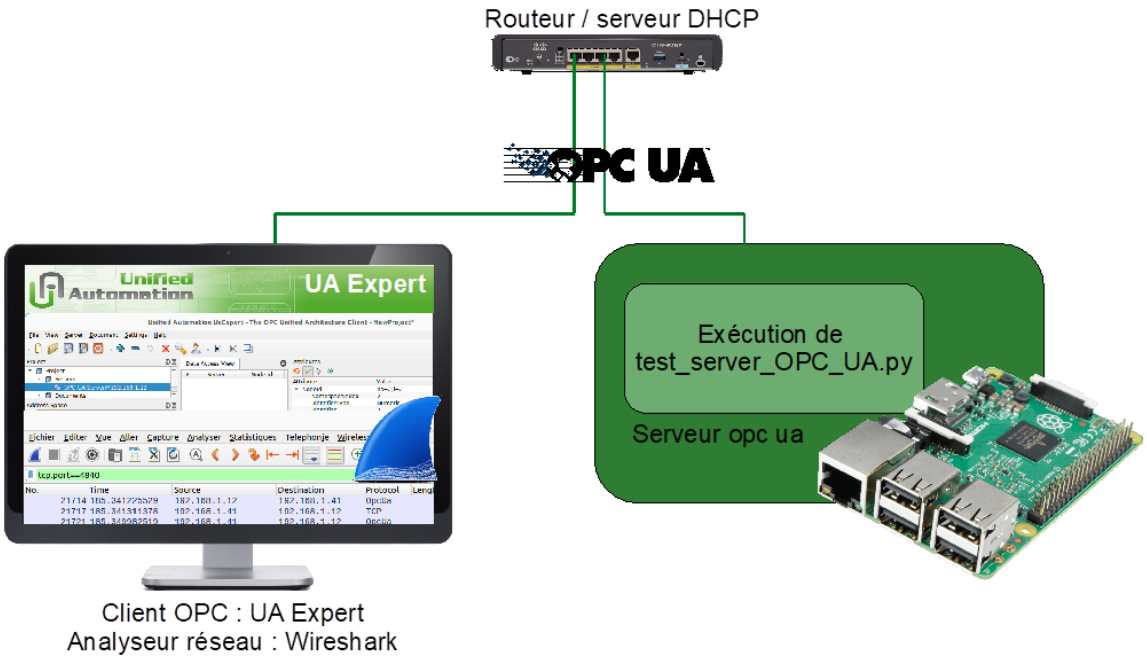


Figure 9 : Première application avec un serveur OPC UA simple, le client UA Expert et Wireshark

Côté serveur (Raspberry Pi 4 avec le module python asyncua)

Le programme du serveur *test_server_OPC_UA.py* est fourni avec cette ressource. Attention, l'adresse IP indiqué ligne 9 doit être l'adresse IP du serveur (donc de la raspberry Pi) :

```
#Programme de test du serveur OPC UA
from opcua import Server, ua
import time

def main():
    # Instanciation du serveur
    server = Server()
    # URL de la Raspberrypi qui hÃ©berge le serveur
    server.set_endpoint("opc.tcp://192.168.1.12:4840/UA/SampleServer")
    # nom du serveur
    server.set_server_name("OPC-UA-Server")
    # Sécurité (rien sur le programme de test)
    server.set_security_policy([ua.SecurityPolicyType.NoSecurity])
    # Nom de l'espace d'adresse
    name = "progDeTest_OPCUA"
    noeud_test = server.register_namespace(name)
    # Création d'un noeud racine
    node = server.get_objects_node()
    # Ajout d'un noeud pour les variables que l'on veut partager
    monObjet    = node.add_object(noeud_test, "monObjet")
    # Création de 2 variables, dont une accessible en écriture
    maVariable1 = monObjet.add_variable(noeud_test,"variableALire",0)
    maVariable2 = monObjet.add_variable(noeud_test,"variableAEcrire",0)
    maVariable2.set_writable()
    ancienneValeur2 = maVariable2.get_value()
    valeur1 = 3.14

    # Demarrage du serveur
    server.start()
    while True:
        # Lecture de la valeur de la variable accessible en écriture
        nouvelleValeur2 = maVariable2.get_value()
        if nouvelleValeur2 != ancienneValeur2 :
            ancienneValeur2 = nouvelleValeur2
            print("valeur recue : ",nouvelleValeur2)
        # Incrémentation de la valeur de maVariable1
        valeur1 = valeur1 + 0.1
        maVariable1.set_value(valeur1)
        time.sleep(1)

if __name__ == "__main__":
    main()
```

On exécute le programme sur la raspberry Pi, avec l'éditeur python Thonny par exemple :

Quelques lignes de commande permettent de vérifier le bon fonctionnement du serveur OPC, directement depuis sur un terminal du nano-ordinateur Raspberry Pi, avec bien sûr l'adresse IP correcte :

- uals --url=opc.tcp://192.168.1.12:4840
- uals --url=opc.tcp://192.168.1.12:4840 --nodeid i=85
- uaread --url=opc.tcp://192.168.1.12:4840 --nodeid "ns=2;i=2"
- uaread --url=opc.tcp://192.168.1.12:4840 --path "0:Objects,2:monObjet,2:variableALire"

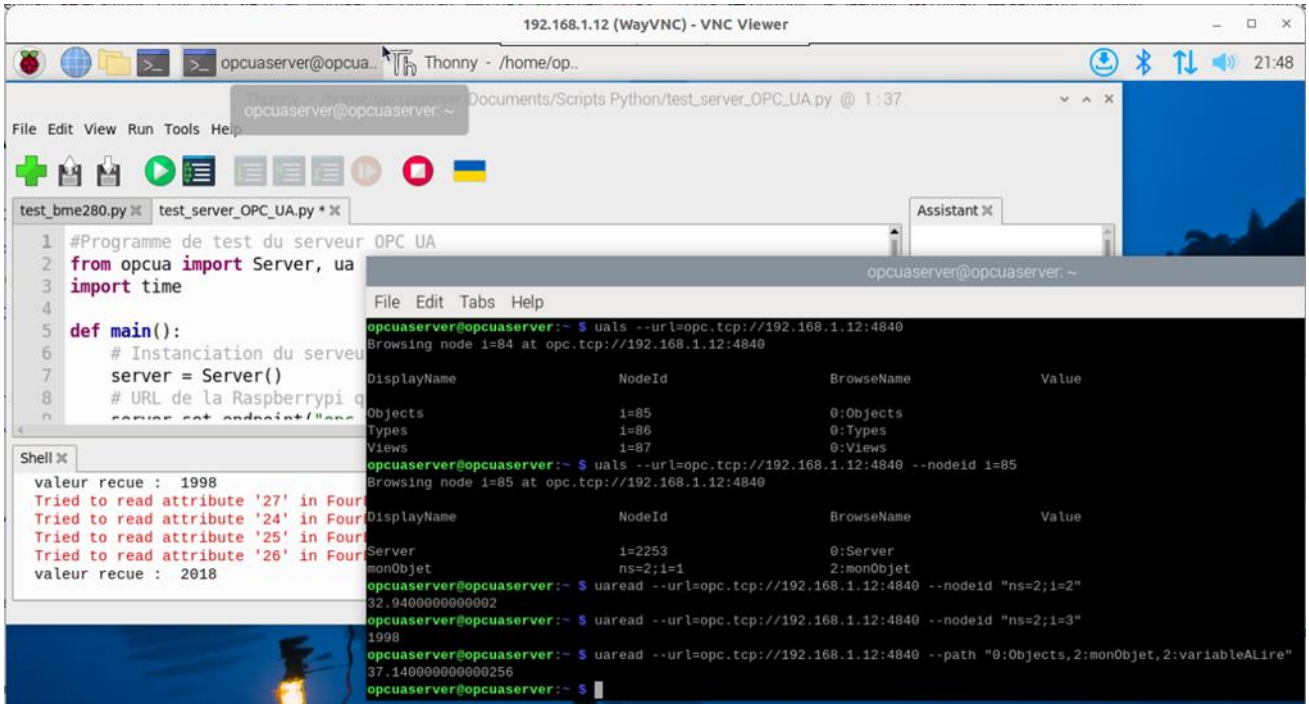


Figure 10 : Bureau du nano-ordinateur Raspberry Pi, serveur OPC UA avec l'IDE Thonny pour l'exécution du programme et la console pour les tests

Côté client (UA Expert sur le PC)

UA Expert [9] est un client OPC UA gratuit, robuste, permettant d'utiliser la plupart des fonctionnalités OPC UA.

Télécharger, installer et lancer UA Expert.

Ajouter le serveur :

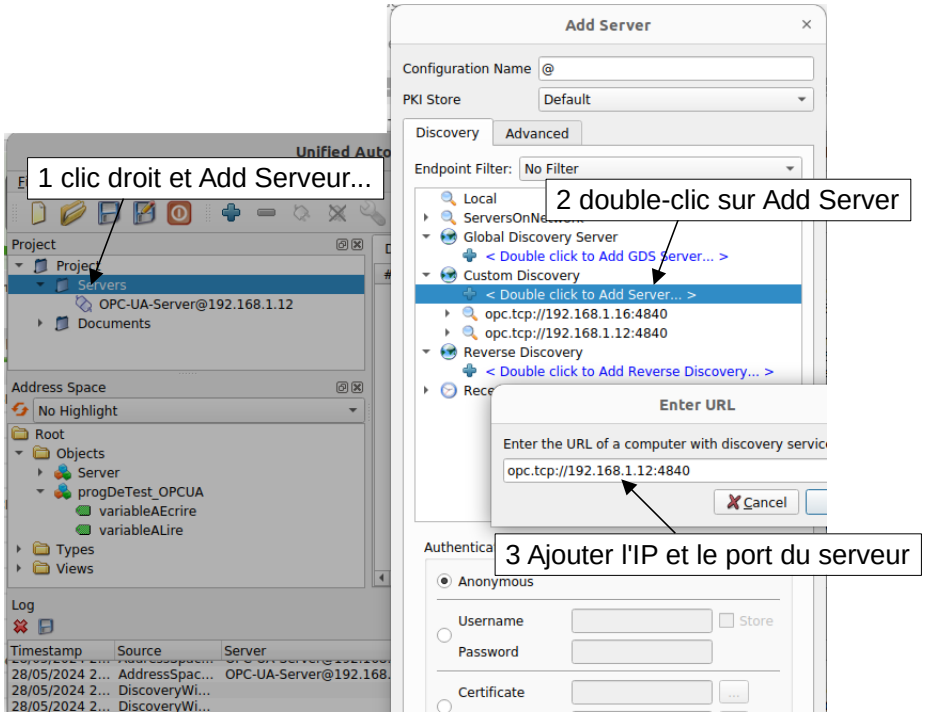


Figure 11 : Ajout d'un serveur sur le client UA Expert

Une fois le serveur dans la liste des Servers de la zone Project, un clic droit dessus permet de s'y connecter.

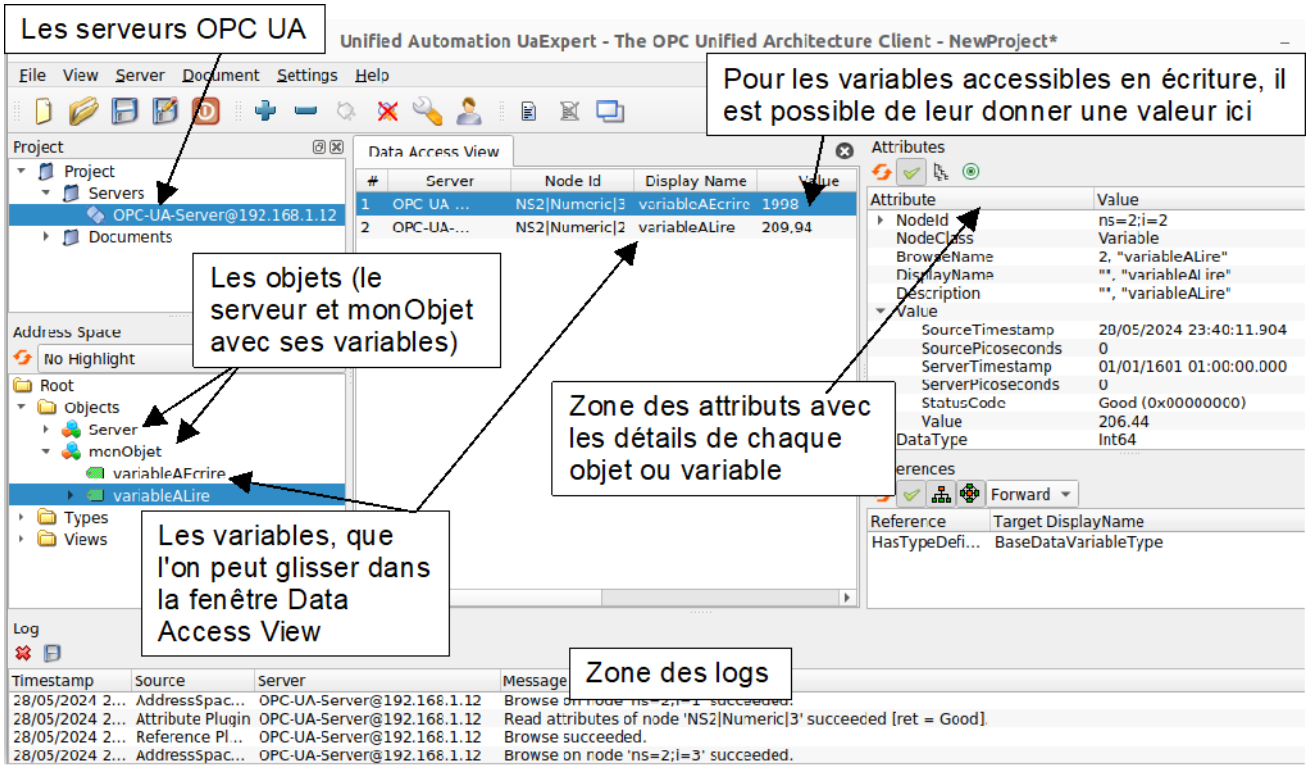


Figure 12 : Utilisation de UA Expert pour afficher et modifier les variables supervisées

Espionnage des échanges (Wireshark sur le PC)

Le logiciel analyseur de réseau Wireshark [10] permet d'observer les échanges entre le client et le serveur. Il est possible de l'installer sur le PC et/ou sur le nano-ordinateur raspberry Pi. L'espionnage de la mise en place de la connexion est intéressant :

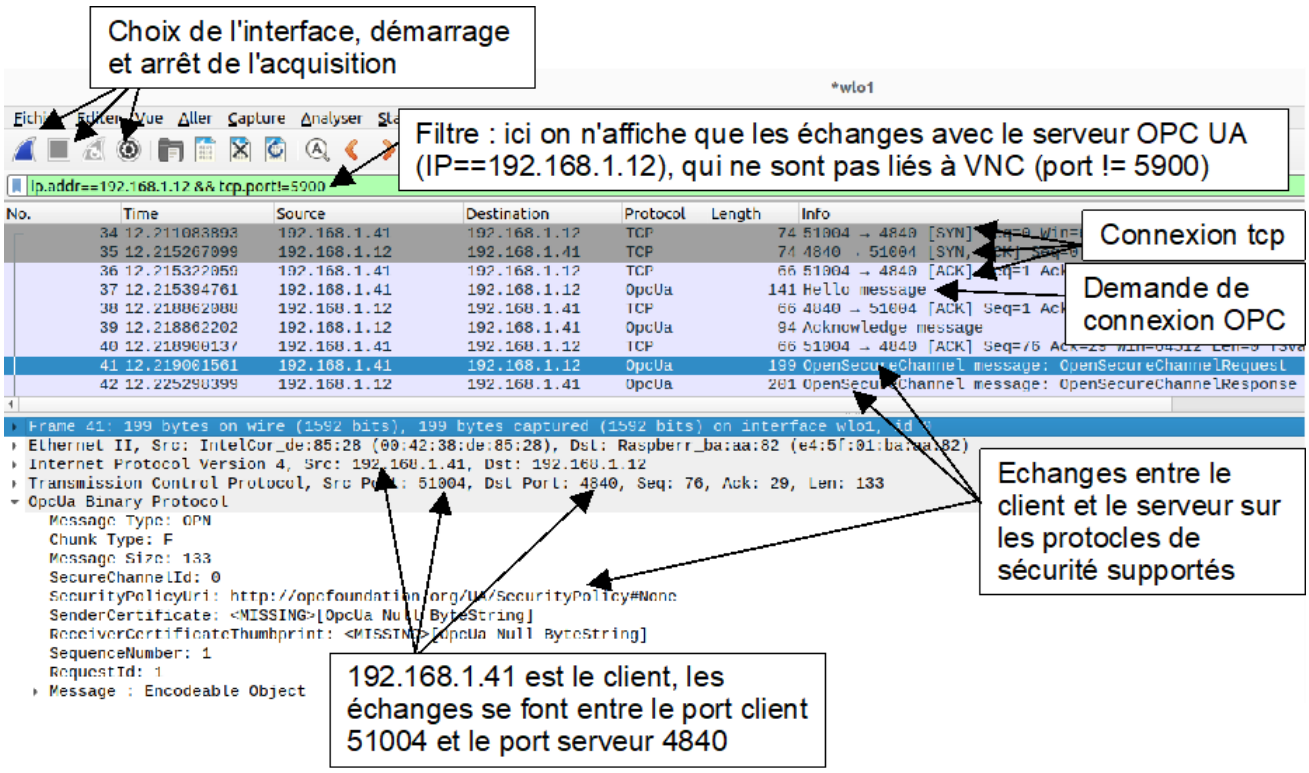


Figure 13 : Espionnage des échanges entre le client et le serveur OPC UA avec Wireshark, lors de la connexion

L'espionnage des échanges une fois la connexion établie permet de mettre en évidence le mode publisher/subscriber.

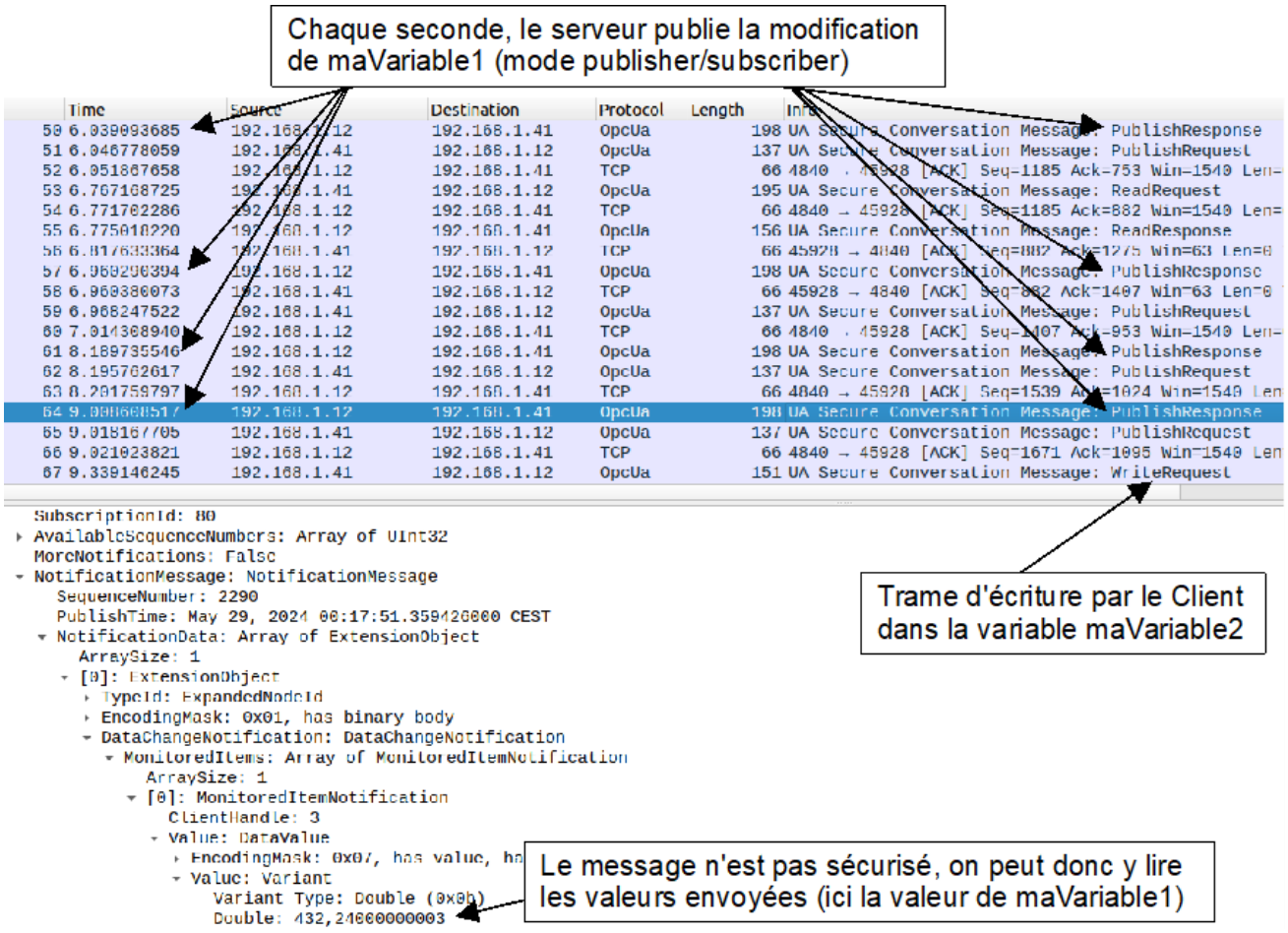


Figure 14 : Espionnage des échanges des valeurs maVariable1 (Serveur -> Client) et maVariable2 (Client->Serveur)

3.4 - Connexion sécurisée OPC UA

Dans ce 2ème exemple, on reprend l'exemple simple, en ajoutant la sécurisation de la connexion. On choisit le chiffrement des échanges et l'authentification (SignAndEncrypt).

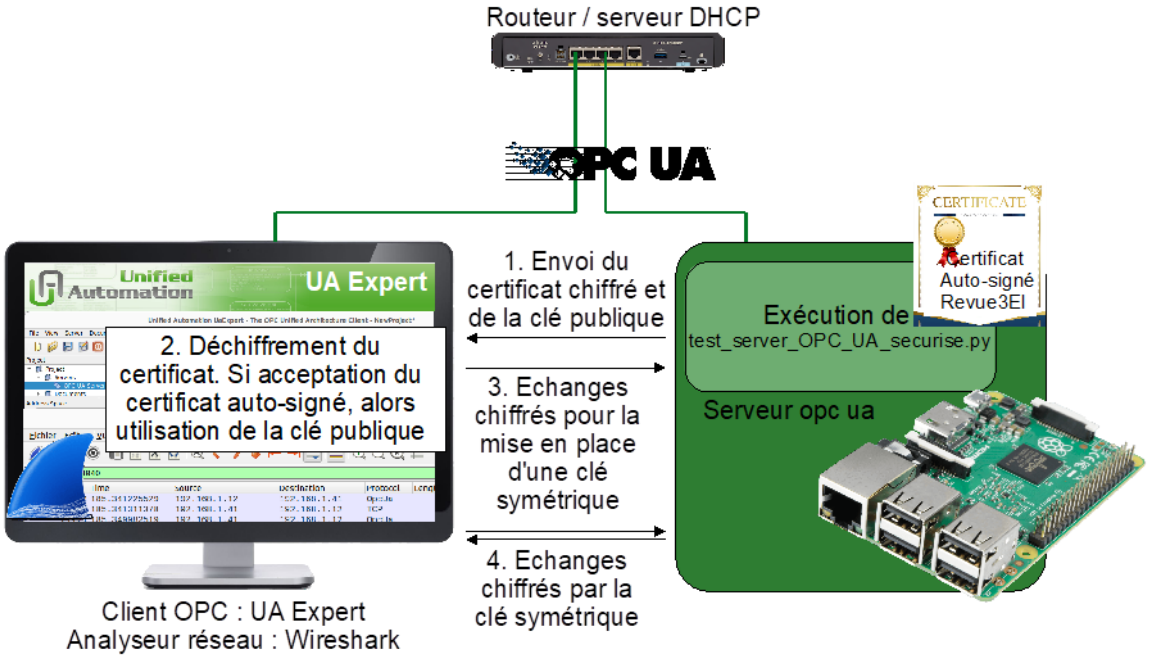


Figure 15 : Mise en place de la connexion OPC-UA sécurisée

La meilleure pratique est de privilégier l'utilisation d'une autorité de certification au lieu d'un certificat autosigné. Cependant, dans un environnement pédagogique, cela est peu accessible. L'authentification est donc basée ici sur des certificats X.509 auto-signés, associés à des clés asymétriques : le serveur envoie sa clé publique et un certificat chiffré par sa clé privée. Le client vérifie que la clé publique déchiffre bien le certificat. Il reste à approuver le certificat (une autre machine pourrait se faire passer pour le serveur), ce qui pour des certificats auto-signés se fait manuellement.

Génération des certificats

La génération de certificats auto-signés est gratuite, mais il faut approuver le certificat serveur manuellement, soit à la demande du client lors de la connexion, soit en ajoutant le certificat dans le dossier des certificats approuvés. Le serveur approuve le certificat du client (il répond à tous les clients).

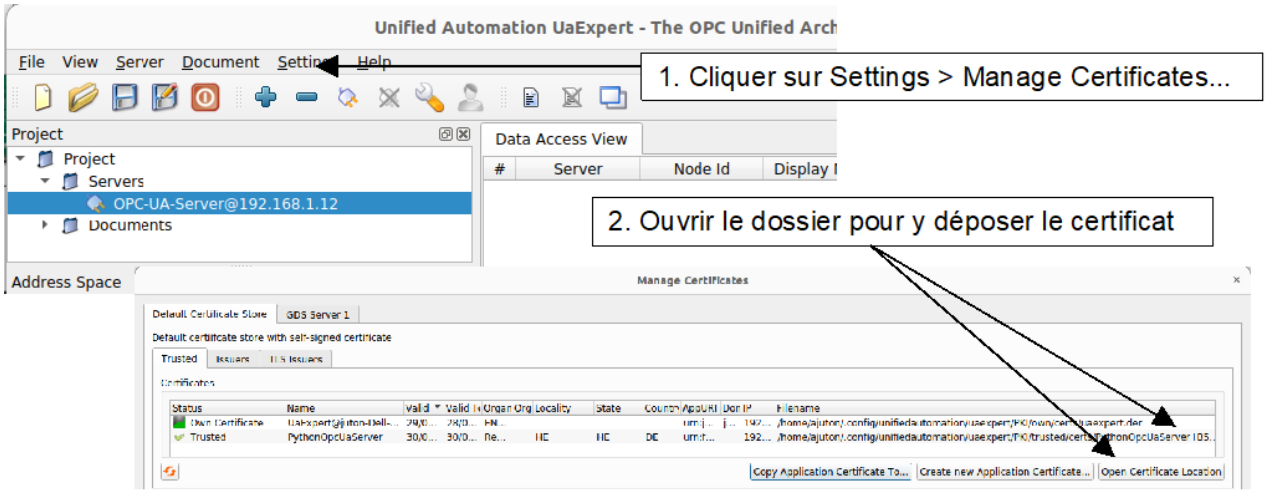


Figure 16 : Management des certificats par UAExpert, avec le dossier de dépôt des certificats

UA Expert crée lui-même le certificat client (*Own Certificate* sur la figure sous-dessous).

Pour le serveur, FreeOPCUA propose un générateur de certificats X.509 auto-signés, avec la configuration pré-remplie, à adapter quelque peu pour que le certificat soit conforme et accepté aussi bien par UAExpert que Panorama (pour la suite) :

On crée le fichier ssl.conf (fourni en pièce jointe à cette ressource). OpenSSL est installé de base sur linux.

```
[ req ]
default_bits = 2048
default_md = sha256
distinguished_name = subject
req_extensions = req_ext
x509_extensions = req_ext
string_mask = utf8only
prompt = no

[ req_ext ]
basicConstraints = CA:FALSE
nsCertType = client, server
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment, keyCertSign
extendedKeyUsage= serverAuth, clientAuth
nsComment = "OpenSSL Generated Certificat"
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
subjectAltName = URI:urn:freeopcua:python:server, IP: 192.168.1.12

[ subject ]
countryName = FR
stateOrProvinceName = IDF
localityName = Saclay
organizationName = Revue3EI
commonName = PythonOpcUaServer
```


Figure 17 : Fichier ssl.conf pour la génération du certificat

On génère alors la clé avec OpenSSL :

```
openssl genrsa -out key2.pem 2048
```

On génère ensuite le certificat X.509, aux formats pem (ascii) et der (binaire) :

```
openssl req -x509 -days 365 -new -out certificate2.pem -key key2.pem -config ssl.conf
openssl x509 -outform der -in certificate2.pem -out certificate2.der
```

Linux permet d'afficher le certificat X.509 généré :

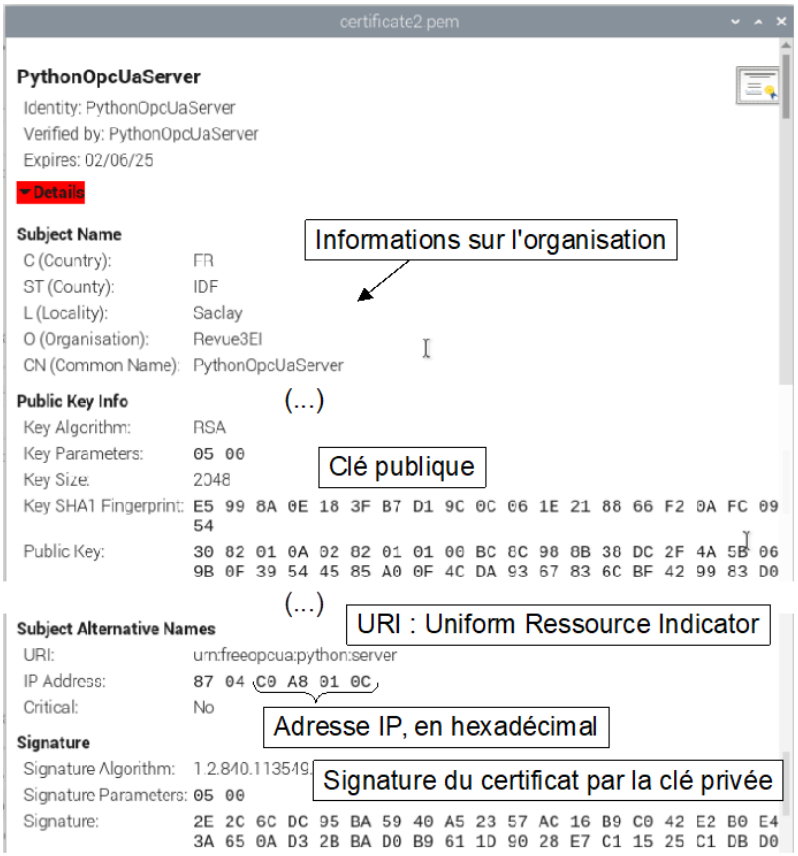


Figure 18 : Affichage du certificat généré

Démarrage du serveur

Une fois certificate2 et key2 copiés dans le dossier du fichier *test_server_OPC_UA_securise.py*, on lance le serveur, avec la sécurité activée :

test_server_OPC_UA_securise.py

```
4
5  def main():
6      # Instanciation du serveur
7      server = Server()
8      # URL de la Raspberrypi qui héberge le serveur
9      server.set_endpoint("opc.tcp://192.168.1.12:4840/UA/SampleServer")
10     # nom du serveur
11     server.set_server_name("OPC-UA-Server")
12     # Sécurité
13     server.set_security_policy([ua.SecurityPolicyType.Basic256Sha256_SignAndEncrypt])
14     server.load_certificate("certificate2.der")
15     server.load_private_key("key2.pem")
16     # Nom de l'espace d'adresse pour éviter les ambiguïtés de noms de noeuds
17     name = "progDeTest_OPCUA"
```

Figure 19 : Début du fichier test_server_OPC_UA_securise.py avec l'intégration du certificat et de la clé

Démarrage du client

Il est possible de se connecter au serveur. Celui-ci n'acceptant que les connexions sécurisées désormais, il propose son certificat au client, que celui-ci doit accepter. 2 possibilités avec UAExpert :

Soit on accepte manuellement le certificat au moment de la première connexion :

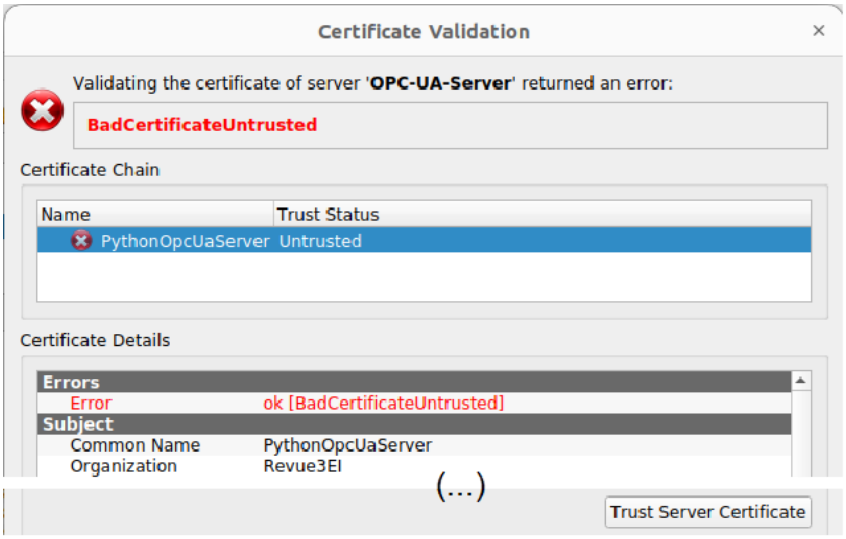


Figure 20 : Fenêtre UAExpert de demande l'acception du certificat

Il est aussi possible de copier à l'avance le certificat dans le dossier des certificats de confiance (voir **Erreur ! Source du renvoi introuvable.**).

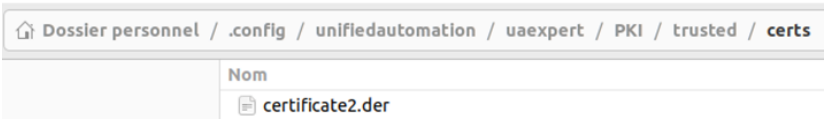


Figure 21 : Fichier certificat copié dans le dossier des certificats de confiance de UAExpert

La communication sécurisée est alors opérationnelle :

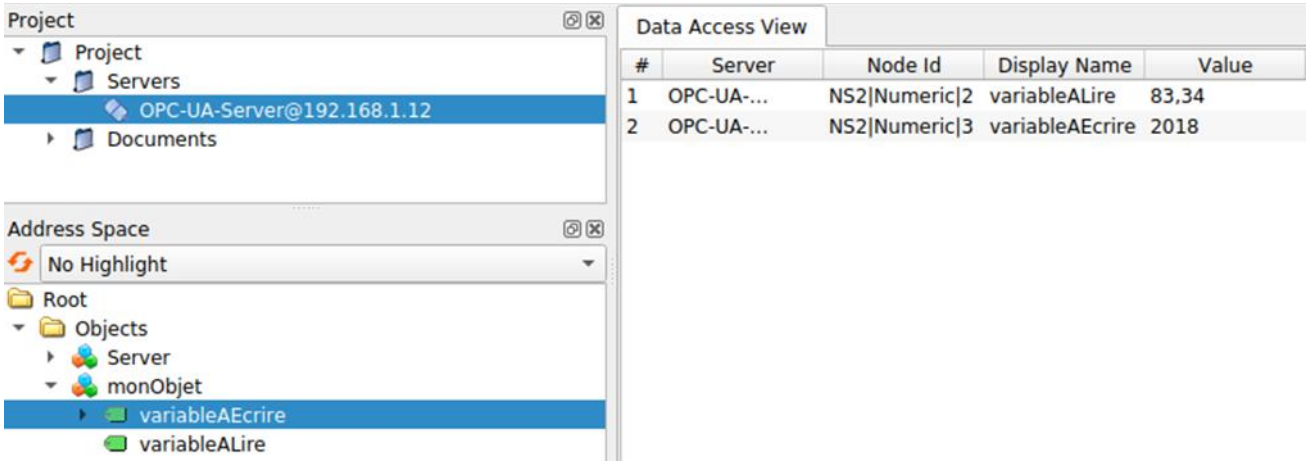


Figure 22 : Communication du client UAExpert avec serveur OPC UA

Observation des échanges OPC UA sécurisés par wireshark

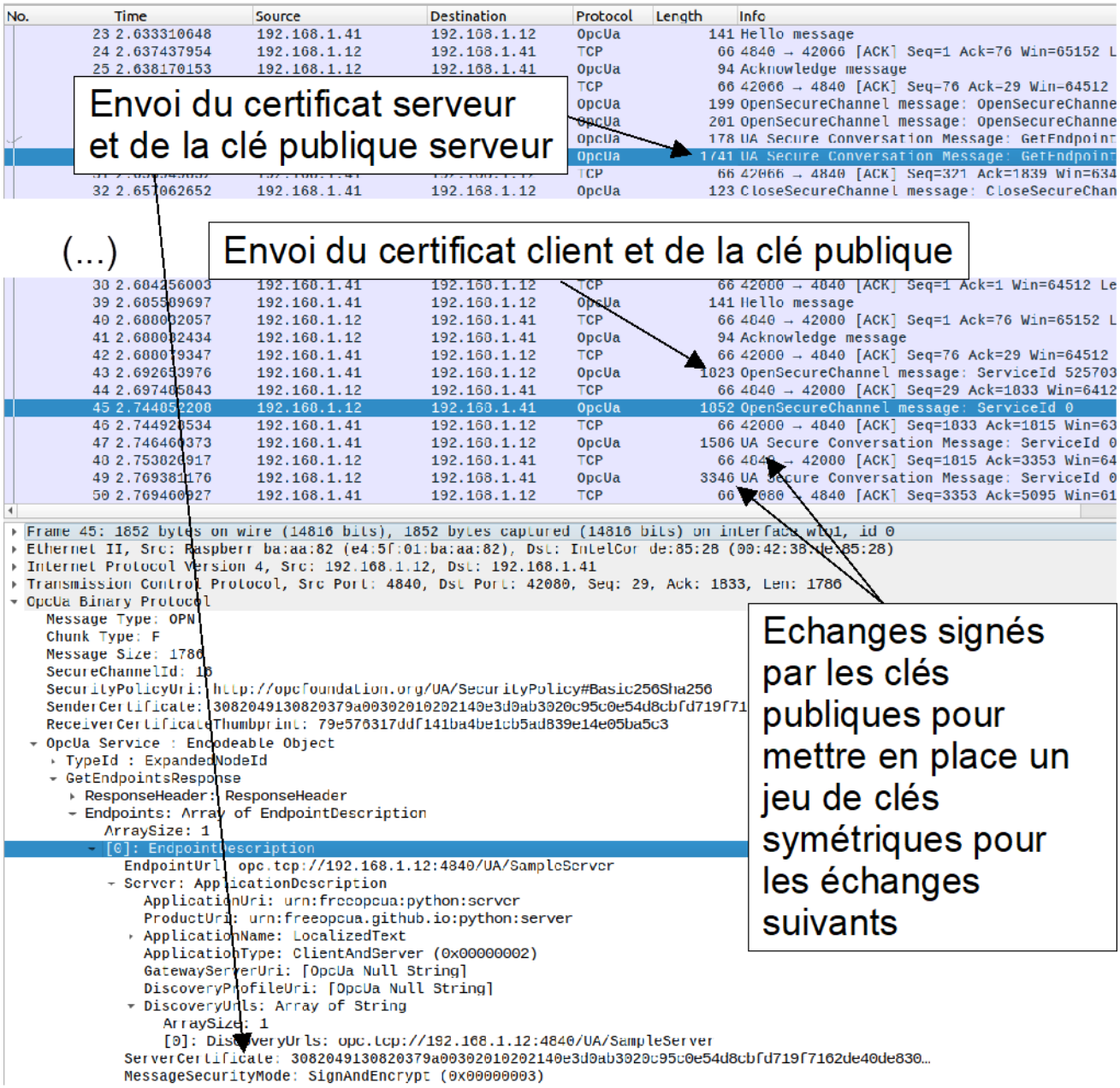


Figure 23 : Mise en place des échanges sécurisés lors de la connexion du client UA Expert au serveur OPC UA

3.5 - Supervision d'un château d'eau simulé via OPC UA

Les échanges supervisés étant en place entre UAExpert et le serveur OPC UA, il est possible de lancer le serveur OPC UA du château d'eau et le client OPC UA sur le superviseur Panorama.

Démarrage du serveur

Sur le serveur, l'ensemble des fichiers nécessaires est fourni en pièce jointe à cette ressource.

Figure 24 : Fichiers du dossier ScriptsPython, pour l'exécution du simulateur de château d'eau

Il faut lancer d'abord le fichier *simulation_usine_eau.py* qui gère le fonctionnement simulé du château d'eau. Ensuite, on exécute *server_OPC_UA.py* qui, comme son nom l'indique gère la mise à disposition des variables par OPC. Les échanges entre les 2 process (simulation_usine_eau et server_OPC_UA) se font par l'intermédiaire de 3 fichiers texte : *fichier_commande_OPC.txt*, *fichier_retour_pompes.txt* et *fichier_retour_reservoirs.txt*.

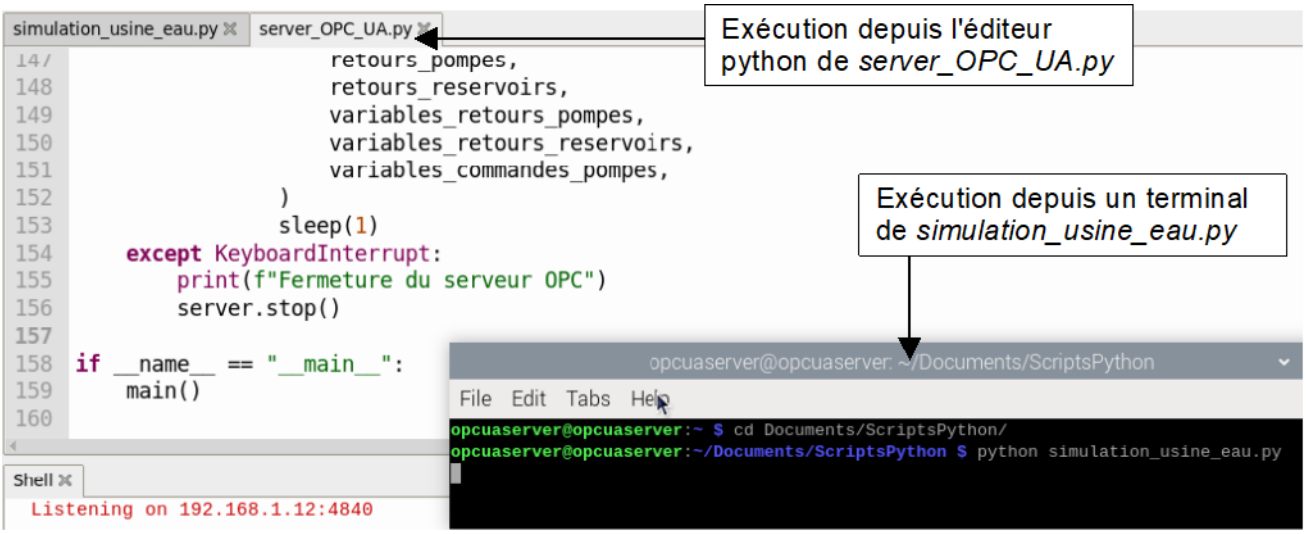


Figure 25 : Copie d'écran du serveur Raspberry Pi lors de la simulation du château d'eau

Vérification du bon fonctionnement du serveur depuis un client UAExpert

Depuis UAExpert, il est possible de se connecter au simulateur de château d'eau pour vérifier son bon fonctionnement.

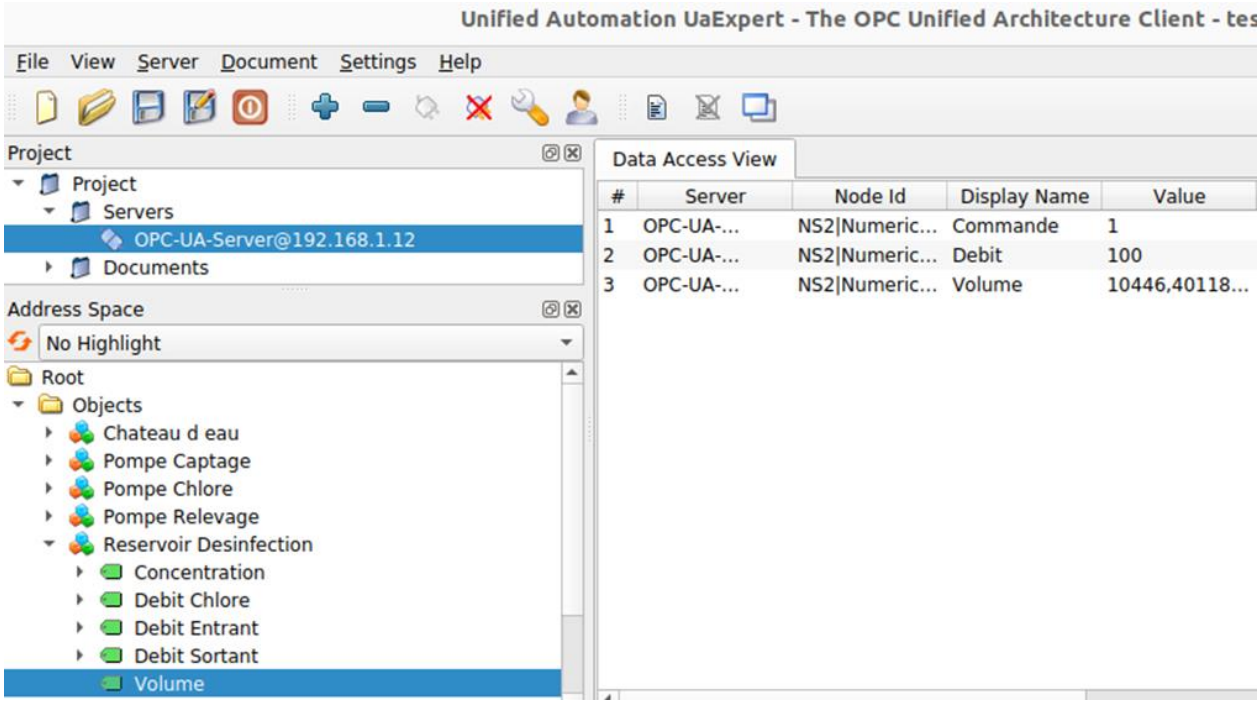


Figure 26 : Supervision des variables du simulateur de château d'eau depuis UAExpert

Cela permet de vérifier le bon fonctionnement mais aussi de mettre en évidence l'intérêt du service Discovery de OPC-UA, fournissant l'ensemble des variables accessibles.

Supervision par Panorama

Le populaire logiciel de supervision Panorama de Codra [3] inclut la possibilité de créer des clients et des serveurs OPC UA. Comme Codra fait partie de la fondation OPC, leur produit est régulièrement validé et mis aux normes. Le logiciel est disponible en version d'essai pour faire des projets de taille raisonnable pour l'enseignement (moins de 50 variables / 4h de connexion) [3]. Panorama ne fonctionne que sous Windows.

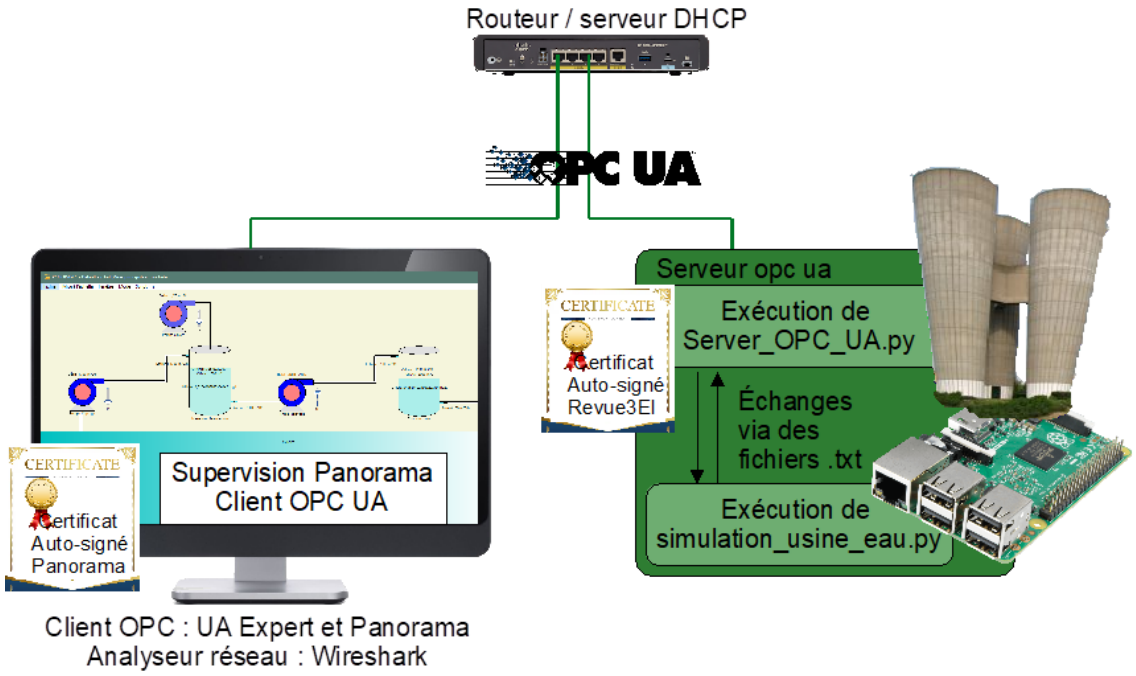


Figure 27 : Schéma de la configuration réseau pour la supervision du château d'eau simulé

Depuis Panorama Studio, ouvrir le fichier Panorama.ini du dossier SimuUsineMars2024 fourni avec cette ressource. SimuUsineMars2024 a été créé avec Panorama version 2023, il pourra donc être

ouvert avec une version plus récente. Aucune garantie par contre pour son ouverture avec une version plus ancienne.

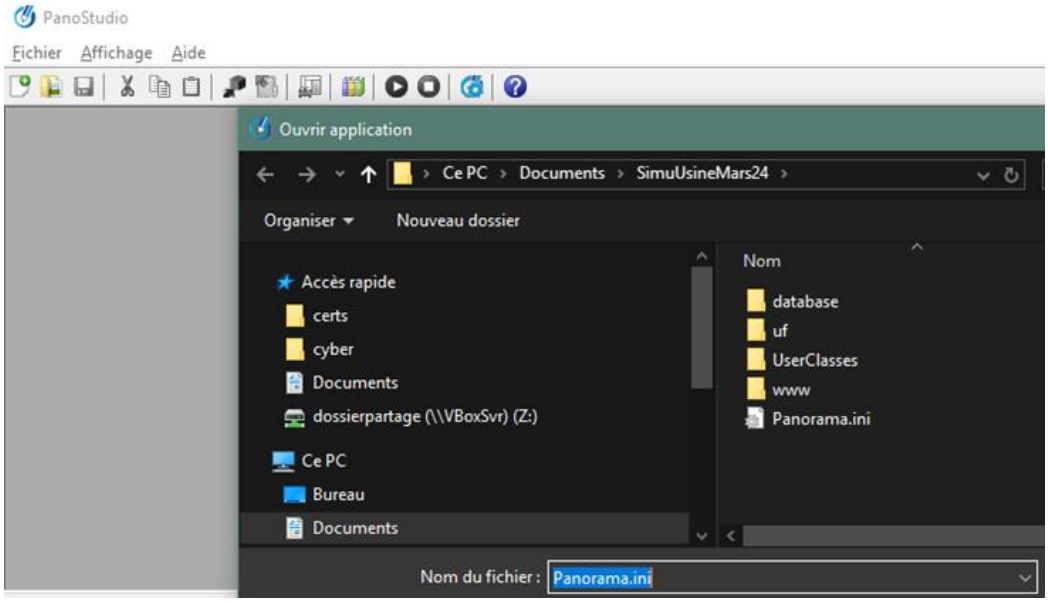


Figure 28 : Ouverture du projet

Ajuster la configuration OPC UA du projet.

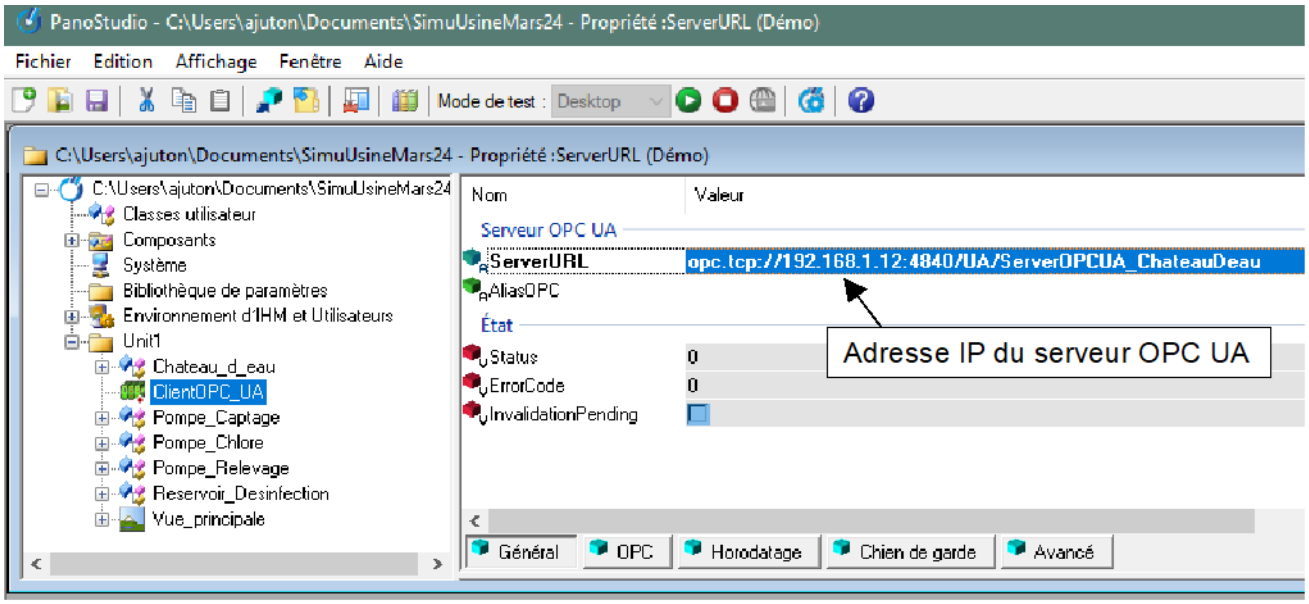


Figure 29 : Configuration de l'adresse IP du serveur OPC-UA / simulateur de château d'eau

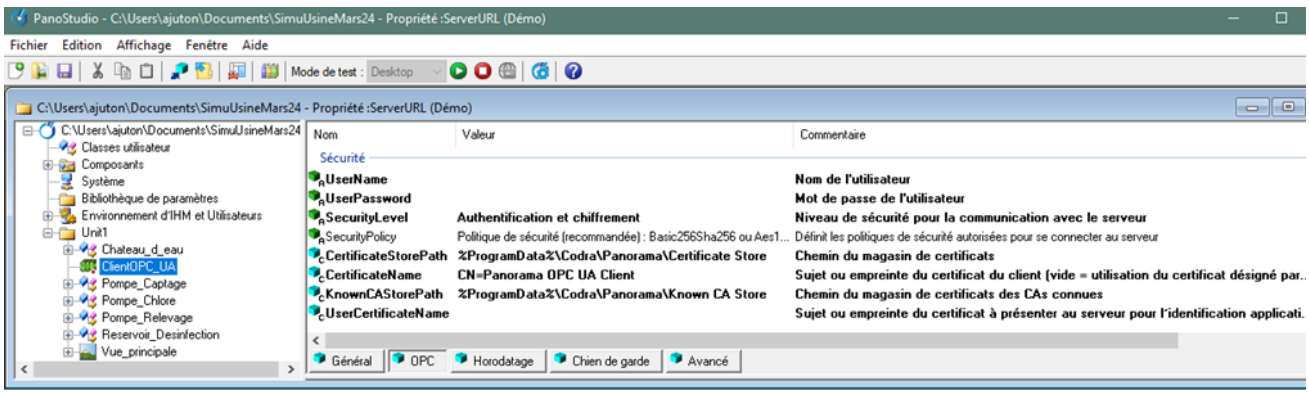


Figure 30 : Configuration des dépôts de certificat du client OPC UA

Pour se connecter de manière sécurisée, il ne manque plus à OPC-UA qu'un certificat SSL (X.509). Pour le créer depuis Windows (OpenSSL ne fonctionne que sous linux), Codra propose une solution (aide de Panorama rubriques *Création et installation du certificat Client OPC UA* et *Configuration de la sécurité*) ainsi qu'une fiche technique *FAQ080-V2.1 Création de certificats pour les fonctions Panorama* [12].

1. Installer [11] et démarrer PowerShell en administrateur

2. Exécuter les commandes suivantes (un peu différente de celles proposées par Panorama pour gérer l'utilisation de certificat auto-signé), en adaptant le chemin de destination du couple certificat/clé. Attention, la clé et le certificat doivent avoir le même nom, seule l'extension les différenciant.

```
PS C:\Users\ajuton\Downloads> $Cert = New-SelfSignedCertificate `
>> -Type Custom `
>> -Subject "CN=Panorama OPC UA Client" `
>> -HashAlgorithm sha256 `
>> -KeyAlgorithm RSA `
>> -KeyLength 4096 `
>> -KeyExportPolicy Exportable `
>> -Provider "Microsoft Enhanced RSA and AES Cryptographic Provider" `
>> -KeySpec KeyExchange `
>> -NotAfter (Get-Date).AddYears(5) `
>> -CertStoreLocation "Cert:\LocalMachine\My" `
>> -KeyUsage DigitalSignature,NonRepudiation,KeyEncipherment,DataEncipherment `
>> -TextExtension("2.5.29.37={text}1.3.6.1.5.5.7.3.2",
"2.5.29.17={text}URL=urn:localhost:CODRA:Panorama OPC UA Client")

PS C:\Users\ajuton\Downloads> Export-Certificate -Cert $Cert -FilePath
C:\Users\ajuton\Downloads\MyOpcUaClientCert5.der

PS C:\Users\ajuton\Downloads> $thumb = $Cert.Thumbprint

PS C:\Users\ajuton\Downloads> $CertPasswordEmpty = new-object System.Security.SecureString

PS C:\Users\ajuton\Downloads> Export-PfxCertificate -Cert "Cert:\LocalMachine\My\$thumb" -FilePath
C:\Users\ajuton\Downloads\MyOpcUaClientCert5.pfx -Password $CertPasswordEmpty
```

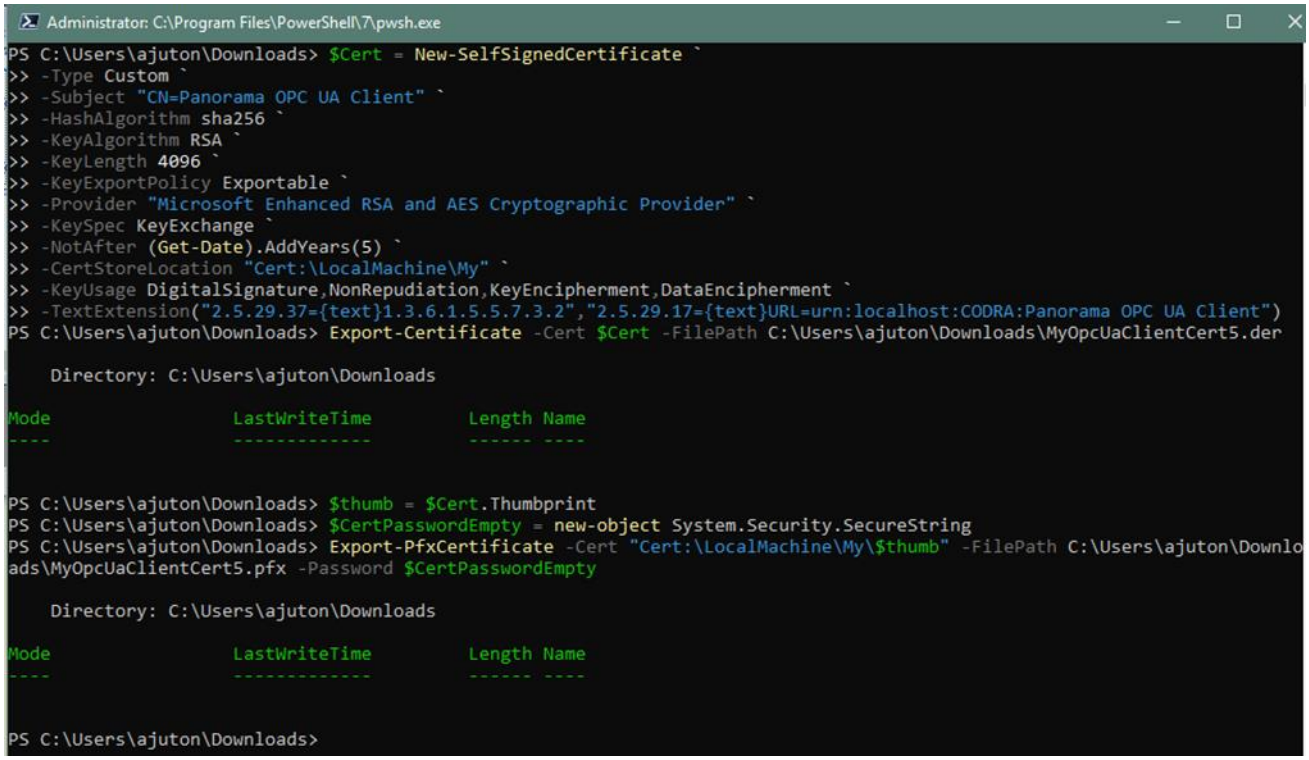


Figure 31 : Copie d'écran des commandes PowerShell de création du certificat X.509 et d'exportation de la clé privée associée

Copier le certificat généré, ainsi que le certificat du serveur OPC UA dans le dossier des certificats de Panorama :

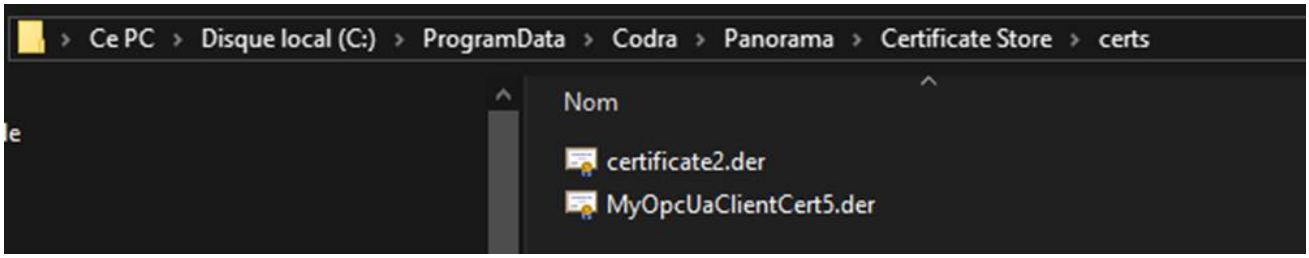


Figure 32 : Dossier certificats de Panorama

Faire de même avec la clé privée associée au certificat Client.

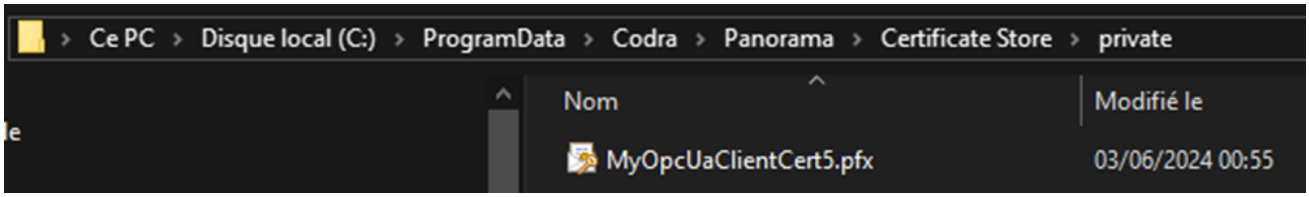


Figure 33 : Dossier clés privées de Panorama

Lancer le logiciel Codra Traceur (installé en même temps que Panorama) et éxécuter la supervision, en choisissant la vue principale. Le château d'eau est alors visualisé et contrôlable (on peut contrôler le débit de chaque pompe).

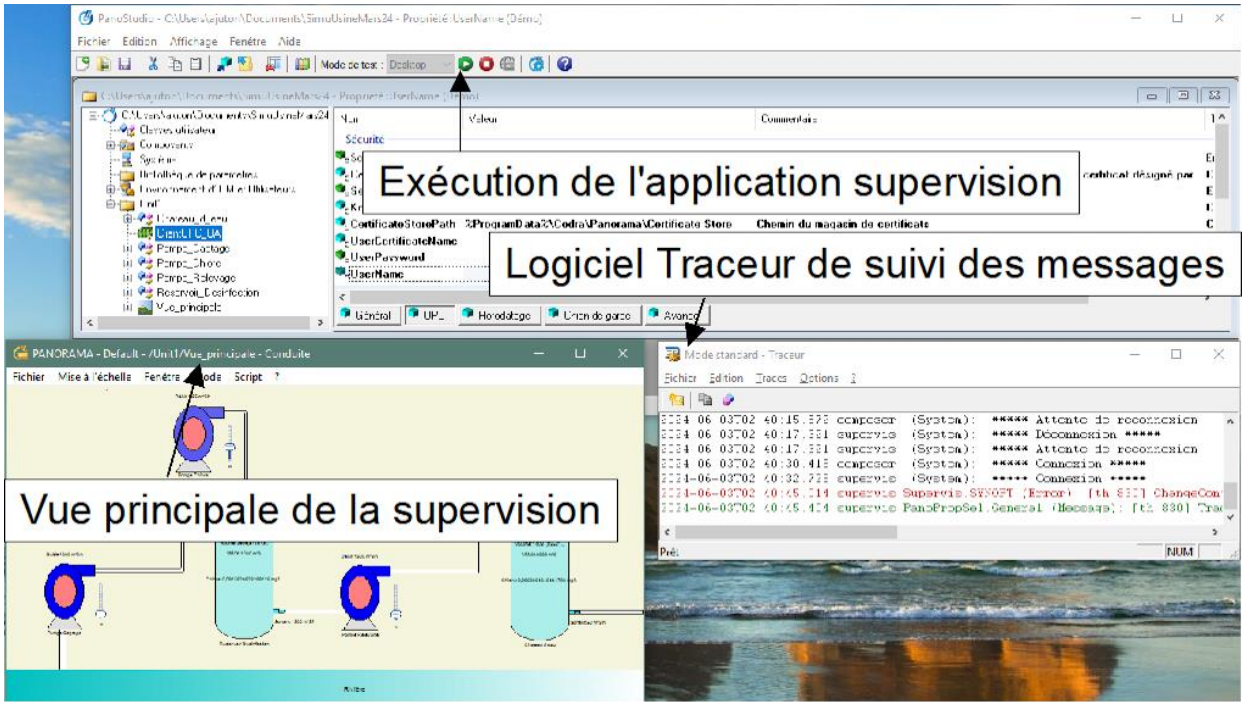


Figure 34 : Vue de la fenêtre de supervision du client OPC UA Panorama

Il est possible de connecter les clients OPC UA UAExpert et Panorama en même temps, montrant par là-même que le serveur accepte plusieurs clients simultanés. Wireshark peut être utilisé pour mettre en évidence la mise en place de la connexion sécurisée et les échanges (peu parlants lorsque la communication est chiffrée)

4 - Conclusion

Cette ressource s'est intéressée essentiellement à l'aspect sécurité d'OPC UA. Une des raisons du succès d'OPC UA est lié à l'utilisation et à la réputation des mécanismes SSL. Le monde de

l'automatisme industriel (OT – Operationnal Technology) tire parti des technologies développées et fiabilisées par l'informatique (IT – Information Technology). Le mode publisher/subscriber et les possibilités temps réels (basées sur la couche réseau TSN Time Sensitive Network) sont d'autres atouts d'OPC UA qui mériteraient également d'être présentés.

Les constructeurs intègrent de plus en plus souvent des serveurs OPC UA à leurs automates programmables. Par exemple, le S7-1200 de Siemens embarque un serveur OPC UA permettant d'accéder aux variables internes de l'automate. Hervé Discours, professeur à l'IUT de Cachan, a fait quelques vidéos très intéressantes sur le sujet [6].

Références :

[1] https://opcfoundation.org

[2] OPCUAcademics propose des ressources pédagogiques sur OPC UA. L'accès gratuit à ces ressources se demande sur la page https://opcfoundation.org/resources/opcuacademic/

[3] Codra, page de présentation de Panorama : https://codra.net/fr/offre-logiciel/plateforme-supervision/logiciel-panorama-suite/ et serveur web de téléchargement de Panorama - https://my.codra.net/

[4] Installation de Raspberry OS sur raspberry Pi4 et mise en place du bureau à distance.

https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay/blob/main/Bibliotheques_logicielles/Installation_RaspberryOS_CoVAPSy_v1re2.pdf

[5] Documentation de FreeOPCUA

https://github.com/FreeOpcUa/opcua-asyncio

https://opcua-asyncio.readthedocs.io/en/latest

[6] Chaîne Youtube de Hervé Discours :

OPC UA - Initiation : https://www.youtube.com/watch?v=iN4qKm5W35g

OPC UA – Cybersécurité : https://www.youtube.com/watch?v=58FUQzWxs3Y

[7] Using the BME280 I2C Temperature and Pressure Sensor in Python, Matt Hawkins

https://www.raspberrypi-spy.co.uk/2016/07/using-bme280-i2c-temperature-pressure-sensor-in-python/

[8] Guide Installation de Raspberry OS sur raspberry Pi 4 :

https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay/blob/main/Bibliotheques_logicielles/Installation_RaspberryOS_CoVAPSy_v1re3.pdf

[9] Site officiel de Unified Automation pour le téléchargement de UA Expert

https://www.unified-automation.com/products/development-tools/uaexpert.html

[10] Site officiel de Wireshark : https://www.wireshark.org/

[11] Installation de PowerShell : https://learn.microsoft.com/fr-fr/powershell/scripting/install/installing-powershell

[12] Fiche technique FAQ080-V2.1 Création de certificats pour les fonctions Panorama :

https://my.codra.net/fr/productreleases?productrelease=PS-2023&selection=technicalfiles

[13] Fondamentaux de la sécurité réseau, M. Sechehaye, A. Juton, M. Sauvergeat, février 2024, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/fondamentaux-dela-securite-reseau

Ce document est accompagné **d'une annexe zip** dont le lien est https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/opcua-protocole-securise-pour-automatisme-industriel

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

Sécurité du protocole LoRaWAN

La Revue 3E.I
Culture Sciences de l'Ingénieur

Édité le 11/06/2024

école normale supérieure paris-saclay

Maxime SECHEHAYE[1]

[1] *ENS Paris-Saclay – DER Nikola Tesla*

Cette ressource fait partie du N°112 de La Revue 3EI du 2ème trimestre 2024.

Cette ressource présente les mécanismes de sécurité existants dans le protocole de communication sans fil LoRaWAN. Il ne s'agit pas d'un rapport complet sur le protocole LoRaWAN et de la technologie de modulation LoRa (déjà présentés dans le numéro 96 [5]) qu'il utilise mais plutôt d'un exposé sur les outils mis à disposition par ce protocole pour sécuriser les échanges de données. On insistera ainsi sur les choix importants qu'un développeur d'application IoT souhaitant utiliser LoRaWAN devra effectuer afin d'assurer la sécurité de son application.

Après avoir introduit le protocole LoRaWAN et rappelé son architecture, nous listerons les éléments de sécurité proposés par ce protocole puis nous étudierons dans le détail le mécanisme de connexion d'un nouveau terminal dans un réseau LoRaWAN déjà existant.

1 - Introduction

LoRaWAN est un protocole de communication radio qui permet à différents terminaux d'établir une communication sans fil et de constituer un LPWAN (*Low Power Wide Area Network*, réseau étendu à basse consommation). Ce protocole utilise la technologie de modulation LoRa pour la communication entre les terminaux et les passerelles, d'où son nom.

Sans rentrer dans les détails de fonctionnement de LoRaWAN et de la technologie LoRa, car ce n'est pas l'objectif de cette ressource, nous allons ici brièvement rappeler les éléments d'un tel réseau LoRaWAN.

L'architecture d'un réseau LoRaWAN suit une topologie de réseau en étoile. Les différents éléments de ce réseau sont :

- Les terminaux (*End Devices*) : objets connectés (capteurs, actionneurs…) qui communiquent avec les passerelles en utilisant la technologie de modulation LoRa ;
- Les passerelles (*Gateways*) : appareils faisant le lien entre les terminaux et les serveurs. Les passerelles ne réalisent aucun traitement sur l'information reçue, elles ne servent que de relais ;
- Le ou les serveurs réseau (*Network Server NS*) : pièce centrale du réseau LoRaWAN qui en assure la gestion et qui vérifie l'intégrité des échanges s'y déroulant ;
- Les serveurs d'application (*Application Servers*) : serveurs qui se chargent de traiter l'information envoyée par les terminaux et si nécessaire d'envoyer une réponse.

Sur un **réseau propriétaire**, tous les équipements appartiennent à la même entreprise, dans une zone limitée (une gare par exemple).

Sur un **réseau opéré**, les terminaux appartiennent à une entreprise A et les passerelles et serveurs réseau à un opérateur B. Le serveur d'application peut appartenir à l'entreprise A ou être hébergé dans un datacenter « cloud » C (OVH, AWS, Azure…). Notons que l'entreprise A peut-être un *fournisseur de service, les données appartenant alors à une entreprise D.*

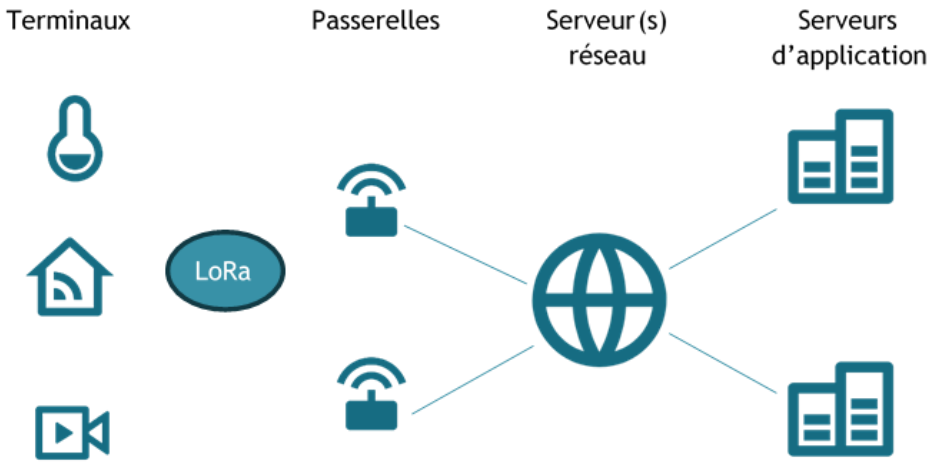


Figure 1 : Schéma de l'architecture d'un réseau LoRaWAN

Il est à noter qu'un terminal n'est pas associé à une unique passerelle. Lorsqu'un terminal souhaite envoyer une information à son serveur d'application, il transmet son message via LoRa et toutes les passerelles à proximité suffisante pour capter ce message le relaieront au serveur réseau. Si le serveur réseau s'aperçoit alors que plusieurs messages identiques arrivent, il n'en gardera qu'un. On a donc une multiplication du message qui permet de réduire la probabilité de devoir l'émettre à nouveau en cas de perte.

LoRaWan ayant une grande portée, « toutes les passerelles à proximité » reçoivent le message signifie que même les messages émis par les terminaux d'un réseau propriétaire sont reçus par les passerelles des opérateurs environnant.

2 - Éléments de sécurité d'un réseau LoRaWAN

Lorsqu'un terminal souhaite envoyer une donnée, son message transitera par au moins trois appareils différents : au moins une passerelle, un serveur réseau et un serveur d'application. Sur un réseau opéré, les 2 premiers appartiennent à l'opérateur, dont on ne maitrise pas la politique de sécurité. Il faut donc que la connexion entre chacun de ces appareils soit sécurisée pour assurer la sécurité de l'ensemble de la chaîne de transmission d'information, tout en restant compatible avec la faible consommation souhaitée et la faible puissance des processeurs.

Voici les éléments de sécurité présents dans le réseau LoRaWAN qui permettent d'augmenter la sécurité des communications établies :

- Des clés de sécurité sont générées et utilisées pour chiffrer les communications depuis les terminaux jusqu'aux serveurs d'application grâce à l'algorithme *AES-128* mais aussi pour vérifier qu'une trame n'a pas été altérée entre son émission et sa réception. Ces clés de sécurité servent donc à assurer l'intégrité et la confidentialité des échanges. Nous reviendrons en détail sur l'établissement et l'utilisation de ces clés de sécurité.
- Deux compteurs de trames (*Frame Counter*) sont utilisés lors de chaque nouvelle connexion entre le terminal et le réseau LoRaWAN et s'incrémentent chaque fois qu'un message ascendant est envoyé (depuis le terminal) ou bien chaque fois qu'un message descendant est envoyé (vers le terminal). On ne peut donc pas retransmettre une trame déjà envoyée.

3 - Établissement d'une connexion sécurisée sur LoRaWAN

Nous allons ici décrire le processus d'établissement d'une connexion sécurisée sur LoRaWAN. Comme expliqué précédemment, ce sont les clés de sécurité qui permettent d'assurer la confidentialité et l'intégrité des messages échangés. Il faut donc s'assurer que l'établissement et le partage de ces clés soient sécurisés.

Il existe deux types d'ajout d'un nouveau terminal sur un réseau LoRaWAN :

- L'activation sans fil (*Over The Air activation*, *OTAA*) où toutes les données liées à la sécurité des communications futures sont établies de manière sans fil entre le serveur du réseau (NS) et le terminal lors de l'établissement de la connexion.
- L'activation par personnalisation (*Activation By Personalization*, *ABP*) où toutes les données liées à la sécurité sont déjà stockées sur le terminal et le réseau LoRaWAN avant les premiers échanges.

Avec la méthode ABP, les mêmes clés de sécurité sont donc utilisées à chaque session et il faut les changer manuellement si besoin. C'est donc la méthode la moins sécurisée des deux.

On va donc s'intéresser ici à l'activation sans fil qui est la méthode d'activation la plus sécurisée.

4 - Activation sans fil d'un nouveau terminal sur LoRaWAN

Avant l'activation, le terminal doit posséder les informations suivantes :

- DevEUI : identifiant unique sur 64 bits attribué dès la fabrication, similaire à une adresse MAC
- AppEUI : identifiant unique d'une application sur le réseau LoRaWAN considéré, sur 64 bits et modifiable
- AppKey : clé sur 128 bits partagée par le terminal et le serveur réseau

La clé AppKey est spécifique à chaque nouveau terminal sur le point de rejoindre un réseau LoRaWAN. Elle n'est *jamais transmise sur le* réseau LoRaWAN pour des raisons évidentes de sécurité et doit donc être transmise entre le terminal et le serveur réseau par un moyen extérieur (provisionnement physique, connexion sécurisée HTTPS…)

4.1 - Demande de connexion (*Join Request*)

Comme son nom l'indique, la première étape consiste à une demande de connexion du terminal au serveur. Cette demande contient :

Le *DevNonce* est un nombre aléatoire et unique. En effet, le serveur réseau conserve les *DevNonce* précédemment utilisés par chaque terminal et rejette toutes les demandes de connexion avec un *DevNonce* déjà utilisé. Cela empêche les attaques par rejeu.

Il y a aussi un *Message Integrity Code (MIC)* qui est calculé à partir des trois champs et de l'AppKey. Il permet de s'assurer, comme son nom l'indique, de l'intégrité du message reçu et de s'assurer de l'expéditeur du message.

Le MIC est systématiquement le résultat d'un chiffrement CMAC-AES sur 128 bits sur les données du message et avec AppKey

La demande de connexion n'est pas chiffrée car elle ne contient pas d'information sensible

4.2 - Acceptation de connexion (Join Accept)

Ce message est envoyé par le serveur réseau au terminal. Il contient les champs suivants :

On va rapidement expliquer ces différents champs :

- AppNonce : Nombre aléatoire fournie par le serveur réseau
- NetID : Ses 7 bits de poids les plus forts représentent l'identifiant du réseau (NwkID) qui est unique dans une zone géographique donnée. Les autres bits donnent l'adresse du terminal dans le réseau.
- DevAddr : Adresse attribuée par le serveur réseau au terminal (similaire à une adresse IP sur un réseau local qui serait fournie par un serveur DHCP)
- DLSettings : Paramètres à utiliser par le terminal pour le *downlink* (lorsque le terminal va recevoir des messages du serveur réseau)
- RXDelay : Délai entre l'émission d'un message par le serveur réseau et le début de sa réception par le terminal
- CFList (optionnel) : Fréquences de canaux autorisées pour les communications entre le terminal et les passerelles

On calcule de nouveau un MIC et cette fois les données sont chiffrées à l'aide d'AppKey pour les protéger

4.3 - Calcul des clés de session

Le serveur réseau et le terminal peuvent alors tous les deux calculer les clés spécifiques à cette nouvelle session : NwkSKey (*Network Session Key*) et AppSKey (*Application Session Key*).

NwkSKey est utilisée pour calculer le MIC des messages mais aussi pour chiffrer les commandes MAC (*Medium Access Control*). Ces commandes sont uniquement entre le serveur réseau et le terminal et permettent de configurer et de contrôler le comportement de ce terminal LoRaWAN. La demande de connexion est un exemple de commande MAC.

AppSKey permet quant à elle de déchiffrer les données (*payload*) des messages entre le terminal et le serveur d'application.

Le serveur réseau conserve donc la clé NwkSKeyet transfère la clé AppSkey au serveur d'application. Comme le lien entre le serveur réseau et le serveur d'application peut être réalisé via n'importe quel protocole de communication, il faut s'assurer que la connexion entre ces deux serveurs soit sécurisée.

Les formules pour obtenir ces deux clés sont les suivantes :

NwkSkey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad16)

AppSkey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad16)

Où *pad16* signifie qu'on ajoute le nombre de zéros suffisant pour que le message chiffré avec la clé AppKey ait une taille de 16 octets.

On retrouve dans ces formules, entre autres, des données envoyées par le serveur réseau dans le message d'acceptation de connexion, ce qui justifie le chiffrement de ce dernier.

4.4 - Le terminal est intégré au réseau LoRaWAN

Maintenant que le terminal est intégré au réseau LoRaWAN, il devra garder pour cette session :

- DevAddr qui lui permet de s'identifier sur ce réseau
- NwkSKey utilisée pour calculer le MIC des messages ainsi que pour chiffrer les messages MAC
- AppSKey utilisée pour chiffrer les données utiles envoyées au serveur d'application

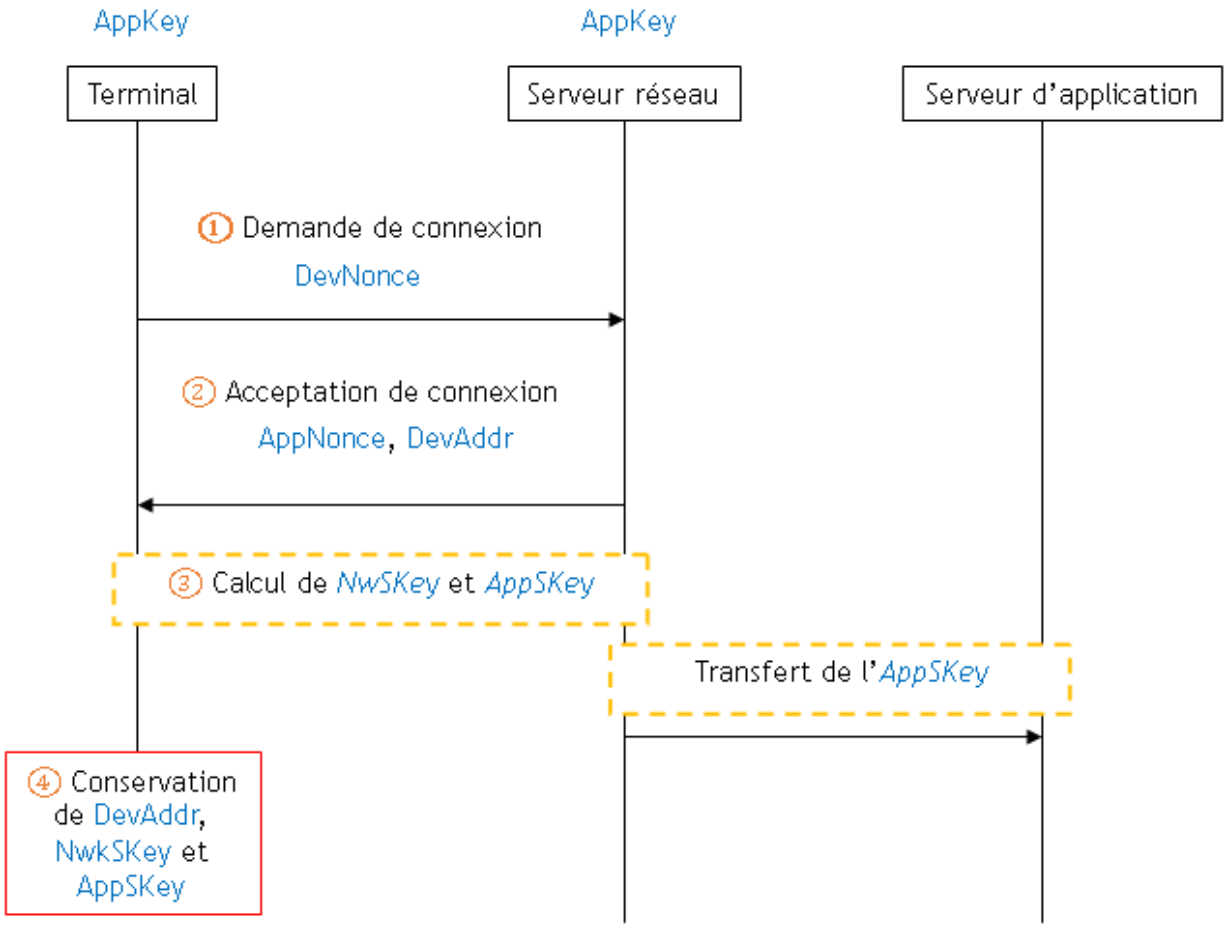


Figure 2 : Diagramme résumant les différentes étapes de l'activation sans fil d'un terminal sur un réseau LoRaWAN

5 - Différences entre LoRaWAN 1.0 et LoRaWAN 1.1

Dans LoRaWAN 1.1, tout ce qui a été exposé précédemment reste valable mais il y a quelques ajouts.

Il y a des améliorations dans la sécurité (changement du Devnonce en compteur, …).

Enfin, il y a davantage de clés de session qui permettent de déléguer au réseau visité certaines fonctions et de piloter le terminal quand le terminal est en roaming.

6 - Conclusion

Nous avons observé que le protocole LoRaWAN propose une architecture séparant les informations liées au réseau de celles liées à l'application spécifique. Pour cela, deux serveurs distincts sont utilisés dans une session : le serveur réseau et le serveur d'application. Il y aura donc une sécurité liée au serveur réseau et une autre liée au serveur d'application.

Ce protocole propose deux méthodes d'activation d'un nouveau terminal mais seule l'activation sans fil permet d'établir une session sécurisée car elle permet la création de nouvelles clés spécifiques à chaque session.

Enfin, le protocole LoRaWAN propose un mécanisme de génération et d'approvisionnement des clés de session à partir d'une clé spécifique au terminal considéré, la clé *AppKey*. Ce mécanisme est assez simple et semble sécurisé dès lors que les messages sont chiffrés, comportent un code d'intégrité ainsi qu'un compteur de trames pour éviter des attaques par rejeu.

Cependant, cette sécurité repose sur une unique clé : la clé AppKey. Ce sera toujours cette même clé qui sera utilisée lors de toute nouvelle session démarrée par le terminal considéré. Il faut donc s'assurer que cette clé est stockée de manière sécurisée et que son partage avec le serveur réseau est aussi sécurisé. Le protocole LoRaWAN n'indique rien à ce sujet, c'est à la responsabilité de l'utilisateur de ce protocole de s'assurer de la sécurité de l'approvisionnement de l'AppKey. Il en est de même pour le transfert de la clé AppSKey du serveur réseau au serveur d'application.

Voici un extrait de la spécification de LoRaWAN 1.1 [2] à ce propos :

> *Secure provisioning, storage, and usage of root keys NwkKey and AppKey on the end device and the backend are intrinsic to the overall security of the solution. These are left to implementation and out of scope of this document.*

Le protocole LoRaWAN propose donc des mécanismes de sécurité classiques et efficaces à condition d'utiliser les bonnes méthodes et d'assurer la sécurité des éléments non pris en charge par le protocole.

Références :

[1]: *LoRaWAN Specification*, LoRa Alliance Technical Committee, 2015

[2]: *LoRaWAN 1.1 Specification*, LoRa Alliance Technical Committee, 2017

[3]: *LoRaWAN Security, Full end-to-end encryption for IoT application providers*, Gemalto, Actility, Semtech, 2017

https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf

[4]: *End Device Activation*, The Things Network

https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/

[5]: Réseaux très basse consommation, longue portée, bas débit, l'exemple de LoRaWAN, A. Juton, septembre 2019, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/reseau-tres-basse-consommation-longue-portee-bas-debit-exemple-lorawan

La Revue 3E.I
Culture Sciences de l'Ingénieur

Sécurité de ZigBee

Maxime SECHEHAYE[1]

Édité le 10/06/2024

[1] *ENS Paris-Saclay – DER Nikola tesla*

Cette ressource fait partie du N°112 de La Revue 3EI du 2ème trimestre 2024.

Cette ressource présente les mécanismes de sécurité présents dans le protocole de communication sans fil ZigBee. On y rappelle brièvement la structure d'un réseau ZigBee avec les différents éléments le composant avant de détailler les différents éléments qui permettent d'établir des communications sécurisées. Ces éléments mis à disposition par le protocole ne sont toutefois pas obligatoirement utilisés par des fournisseurs de solution ZigBee. On s'attachera donc à proposer des exemples de choix judicieux pour améliorer la sécurité d'un réseau ZigBee.

1 - Introduction

ZigBee est un protocole de communication qui permet à différents équipements géographiquement proches de communiquer sans fil. Il s'agit d'un protocole d'utilisation simple et peu chère, particulièrement adapté pour des réseaux sans fil personnels (*Wireless Personal Area Network*, *WPAN*) avec des applications de domotiques par exemple.

1.1 - Rappels sur l'architecture d'un réseau ZigBee

Pour pouvoir aborder sereinement le reste de cette ressource, voici un bref rappel sur l'architecture d'un réseau ZigBee avec ses différents composants.

Un réseau ZigBee est composé de trois types d'appareils :

- Coordinateur : c'est le premier membre d'un réseau ZigBee. C'est donc lui qui va le configurer. Il attribue les adresses des nouveaux membres du réseau, surveille l'état du réseau et participe au routage des messages.
- Routeur : appareil permettant de transmettre les messages en choisissant le chemin optimal pour atteindre le terminal de destination.
- Terminal : dispositif émetteur et/ou récepteur qui échange des informations avec d'autres nœuds du réseau ZigBee.

Un réseau ZigBee peut être réalisé suivant une des trois topologies suivantes : en étoile, maillée ou en arbre.

Dans la topologie en étoile (voir la figure 1), il y a un coordinateur qui se charge de l'ensemble du routage. Tous les autres nœuds du réseau sont des terminaux. C'est une topologie simple et facile à mettre en place mais dont le bon fonctionnement repose uniquement sur un nœud, le coordinateur.

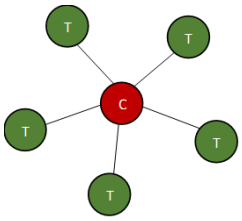


Figure 1 : Topologie en étoile d'un réseau ZigBee (C : Coordinateur, T : Terminal)

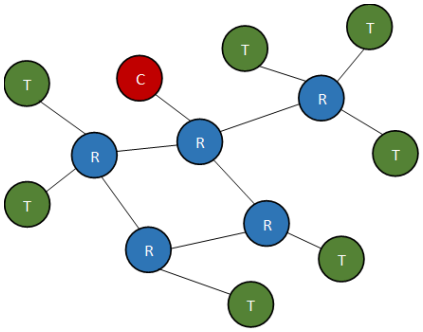


Figure 2 : Topologie maillée d'un réseau ZigBee (C : Coordinateur, R : Routeur, T : Terminal)

Dans les topologies maillée (*mesh*) et en arbre (voir les figures 2 et 3), le réseau composte des routeurs. La topologie maillée offre plus de redondance en cas de panne d'un routeur.

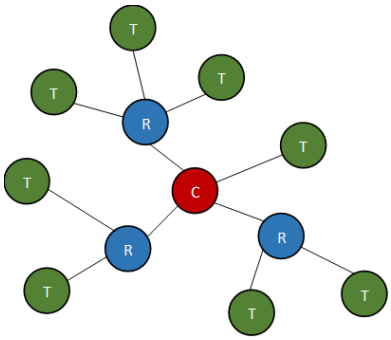


Figure 3 : Topologie en arbre d'un réseau ZigBee (C : Coordinateur, R : Routeur, T : Terminal)

1.2 - Structure d'une trame ZigBee

Voici la structure d'une trame ZigBee qui nous sera utile pour comprendre certains éléments de sécurité par la suite :

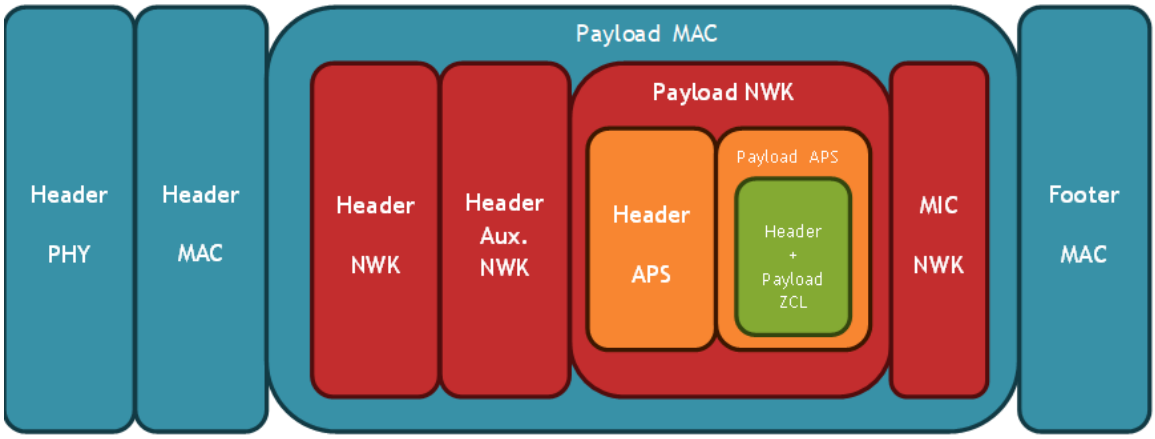


Figure 4 : Structure d'une trame ZigBee

Où PHY correspond à la couche physique, MAC signifie *Medium Access Control*, NWK représente la couche réseau, APS la couche d'application et ZCL la *Zigbee Cluster Library* (fonctionnalités courantes standardisées pour un développement plus rapide).

2 - Éléments de sécurité du protocole ZigBee

Le protocole de communication ZigBee prévoit un certain nombre d'éléments qui, si mis en place et utilisés correctement comme nous le verrons dans la suite de cet article, permettent d'améliorer la sécurité du réseau. On va ici détailler chacun de ces éléments.

2.1 - Compteur de trames

Lorsqu'un nœud du réseau ZigBee envoie un message, la trame comporte un champ correspond à son compteur de trames. Ce compteur permet d'éviter les attaques par rejeu.

Chaque nœud maintient une liste des compteurs de trames de ses voisins et de ses enfants (dans le cas d'un routeur). Ainsi, lorsqu'il reçoit un message d'un de ses voisins, il peut facilement vérifier si le compteur de trames présent dans ce message est supérieur au dernier dont il a eu connaissance. Si ce n'est pas le cas, le message est ignoré.

2.2 - Centre de confiance centralité (*Centralized Trust Center*)

Le centre de confiance centralisé est le nœud du réseau ZigBee où est centralisée la gestion de la sécurité de ce réseau. Ce nœud :

- Authentifie les nouveaux arrivants sur le réseau.
- Autorise ou non l'accès au réseau.
- Distribue les clés de sécurité.

Le choix du centre de confiance ainsi que sa politique de fonctionnement sont des choix cruciaux lors de la création d'un réseau ZigBee

Le plus souvent, le centre de confiance centralisé est confondu avec le coordinateur.

2.3 - Liste d'accès contrôlé (*Access Control List*)

Cette liste permet au centre de confiance de définir des règles d'accès au réseau ZigBee pour les nouveaux nœuds. On peut y renseigner une liste d'appareils autorisés (*white list*) ou bien une liste d'appareils interdits (*black list*). On peut également y définir des permissions accordées ou non à certains nœuds. L'implémentation de cette liste varie d'une solution ZigBee à l'autre suivant la politique de sécurité désirée.

2.4 - Clés de sécurité

Clé de sécurité du réseau (*Network Key*)

Il s'agit d'une clé de chiffrement sur 128 bits commune à tous les nœuds du réseau ZigBee et transmise lors de la procédure d'intégration au réseau. Cette clé sert à chiffrer les messages de maintenance du protocole ZigBee au sein du réseau mais aussi à chiffrer les informations réseau contenues dans les messages envoyés. C'est généralement cette clé qui est utilisée lors du calcul du code d'intégrité (*Message Integrity Code*, *MIC*) de chaque message (voir *MIC NWK* sur la figure 4).

Quand elle est transmise à un nouveau nœud, cette clé est elle-même chiffrée avec une clé préconfigurée déjà connue du centre de confiance et du nouveau nœud.

Clé de sécurité de la liaison (*Link Key*)

Cette clé de chiffrement est créée pour chiffrer les données échangées entre deux nœuds spécifiques. Les messages entre ces deux nœuds sont donc chiffrés à la fois avec la clé de liaison et la clé de réseau.

C'est le centre de confiance qui se charge de générer aléatoirement cette clé et de la transmettre aux deux nœuds concernés. Pour ce faire, une clé préconfigurée peut être partagée entre le nœud et le centre de confiance. C'est avec cette clé que le centre de confiance partagera de manière sécurisée la clé de réseau et pourra aussi générer une clé de liaison spécifique à cette session entre lui et le nouveau nœud. Ainsi, lorsque le nœud demande une clé de liaison pour chiffrer ses communications avec un autre nœud, le centre de confiance lui transmettra une clé aléatoire chiffrée avec la clé de liaison qu'il partage avec ce nœud.

Provisionnement de clé par vérification de certificat (*Certificate-Based Key Establishment*)

Il est possible d'utiliser des certificats numériques fournis par des organisations reconnues pour identifier de manière sécurisée un nouveau nœud sur le réseau ZigBee. Ce certificat permet aussi de procéder à un échange sécurisé d'une clé de liaison entre le centre de confiance et le nouveau nœud.

Politique de sécurité décentralisée (*Distributed Security*)

Depuis ZigBee 3.0, il est possible de décentraliser la sécurité du réseau. Lorsque cette politique de sécurité est choisie, ce sont les routeurs par lesquels les nouveaux nœuds entrent sur le réseau ZigBee qui sont en charge d'authentifier ces nouveaux arrivants et de distribuer les clés de sécurité. Il n'y a alors plus de nœud central qui a connaissance de tous les nœuds authentifiés du réseau.

La documentation officielle de ZigBee n'impose pas une politique de sécurité en particulier. Le choix est laissé aux fournisseurs de solutions ZigBee.

La seule obligation donnée par le protocole ZigBee est l'utilisation d'un chiffrement par bloc AES avec une clé sur 128 bits.

Sécurité « saut par saut »

Dans le protocole ZigBee, la sécurité est effectuée en mode « saut par saut » (*hop-by-hop*). Cela signifie que chaque fois qu'un paquet ZigBee passe par un routeur, ce dernier vérifie l'intégrité du paquet avec le MIC et empêche toute attaque par rejeu en vérifiant le compteur de trames.

Une fois ces vérifications effectuées, s'il n'y a aucun problème, le routeur va rechiffrer la trame avec la clé de réseau et modifier les champs du header *Aux. NWK* de la trame (voir figure 4), notamment l'adresse source ainsi que le compteur de trames. Ainsi, lorsqu'un autre routeur ou le terminal de destination recevra la trame, il pourra vérifier à nouveau qu'elle n'a pas été altérée ou rejouée.

Cette politique de sécurité permet de ne pas encombrer le réseau avec des messages corrompus ou rejoués car ils seront rapidement identifiés et ignorés. Cependant, les routeurs sont davantage

sollicités par rapport à du simple routage. C'est une des raisons pour lesquelles le protocole ZigBee n'est pas adapté pour des réseaux de grande échelle.

3 - Sécurisation de l'intégration d'un nouveau nœud au réseau

On va décrire ici les principaux éléments qui ont une influence sur la sécurité du réseau lorsqu'un nouveau nœud le rejoint.

3.1 - Protection de la clé de réseau lors de son transfert

Il est crucial de sécuriser le transfert de la clé de réseau au nouveau nœud. Le choix de la clé préconfigurée en est grandement responsable. Il existe différents choix que nous allons détailler.

Clé par défaut - « *Well-known* »

Il s'agit d'une clé préconfigurée par défaut et qui est donc connue de tous, sur tous les réseaux ZigBee. Cette clé a pour valeur hexadécimale 5A 69 67 42 65 65 41 6C 6C 69 61 6E 63 65 30 39, ce qui donne après décodage ASCII « ZigBeeAlliance09 ».

Il est évident qu'il faut absolument éviter ce choix de clé préconfigurée car il n'apporte aucune sécurité au transfert de la clé réseau.

Clé préconfigurée spécifique au réseau

On peut configurer une clé qui sera choisie comme clé de lien préconfigurée pour tous les nouveaux nœuds sur le réseau.

Le déploiement de cette solution est facile car la clé est commune à tous les nœuds sur le réseau. La sécurité du transfert de la clé de réseau est alors améliorée par rapport à la clé par défaut qui est connue même à l'extérieur du réseau ZigBee concerné.

Clé de liaison dérivée à partir d'un code d'installation

Depuis ZigBee 3.0, un code d'installation (*install code*) peut être utilisé par le nœud entrant et le centre de confiance pour générer une même clé permettant de chiffrer la clé de réseau. Ce code est généré aléatoirement lors de la conception du nœud et doit être communiqué au centre de confiance du réseau avant l'arrivée du nœud dans ce réseau.

Ce même code d'installation peut ensuite être utilisé pour générer une clé de liaison entre le centre de confiance et le nouveau nœud.

Il est vivement recommandé de transmettre hors-réseau le code d'installation au centre de confiance (par QR code par exemple).

Le code d'installation permet l'authentification du nouveau nœud et garantit l'unicité de la clé au sein du réseau. C'est donc le choix apportant le plus de sécurité au transfert de la clé de réseau.

3.2 - Mise en service hors-réseau

Il est possible de transférer hors-réseau toutes les informations nécessaires à l'intégration au réseau. On évite donc de passer par le réseau où ces informations sensibles pourraient être interceptées. Différentes solutions sont envisageables :

Pré configuration lors de la conception de l'appareil

Lorsqu'on fait ce choix, la clé de réseau est présente en permanence sur l'appareil. Il n'y a alors pas le droit à l'erreur sinon l'appareil sera inutilisable. De plus, un piratage d'un tel appareil rendrait la clé de réseau accessible depuis l'extérieur, ce qui compromettrait le réseau concerné.

NFC / QR code

On peut utiliser une communication NFC ou encore un QR code pour transmettre la clé de réseau. Ces mécanismes de communication sont très faciles d'utilisation mais il ne suffit sur place que d'un lecteur pour obtenir les informations sensibles.

Site internet

On peut enfin stocker toutes les informations de sécurité dans une base de données qui est reliée à un site internet. Si on utilise les codes d'installation pour le transport de la clé de réseau, cela peut nécessiter une base de données conséquente. De plus, on ne fait que transmettre la responsabilité de la sécurité au site internet.

4 - Sécurisation après l'intégration d'un nouveau nœud au réseau

4.1 - Empêcher les attaques par reconnexion

Si la clé préconfigurée peut être réutilisée à chaque reconnexion, il est possible de simuler une tentative de reconnexion en usurpant l'identité de l'appareil et en utilisant cette clé préconfigurée. On peut alors obtenir la clé de réseau et donc compromettre l'ensemble du réseau ZigBee.

Pour empêcher ce type d'attaque, ZigBee 3.0 propose un mécanisme de négociation de clé de liaison avec le centre de confiance. Lors de la première connexion de l'appareil au réseau ZigBee avec la clé préconfigurée, le centre de confiance génère une nouvelle clé de liaison qui sera utilisée lors des reconnexions futures.

Il est recommandé de désactiver la reconnexion avec la clé préconfigurée dans la politique du centre de confiance. Si un appareil tente de se reconnecter avec la clé préconfigurée, il faudrait mettre en place une intervention manuelle au centre de confiance pour autoriser ou non cette reconnexion.

4.2 - Changer régulièrement la clé de réseau

Puisque la clé de réseau est l'élément de sécurité majeur du réseau ZigBee, il est souhaitable de la changer régulièrement afin de s'assurer qu'aucun piratage d'un ancien appareil du réseau ne vienne compromettre ce réseau. On procède ainsi :

- Le centre de confiance diffuse la nouvelle clé de réseau à tous les nœuds en la chiffrant avec la clé actuelle qui est sur le point de de devenir obsolète

- Le centre de confiance diffuse un message *Switch Key* comprenant le numéro permettant d'identifier la nouvelle clé de réseau. C'est dorénavant cette clé qui sera utilisée sur le réseau.

Pour ne pas surcharger le réseau ZigBee de messages de diffusion, il faut trouver un juste intervalle de temps au bout duquel on change la clé de réseau. C'est un compromis entre sécurité et performance du réseau ZigBee. On pourrait le faire à chaque fois qu'un appareil quitte le réseau pour s'assurer qu'un piratage de cet appareil ne compromette pas la sécurité du réseau.

5 - Conclusion

Nous avons ici observé les éléments de sécurité présents dans le protocole de communication ZigBee. Ce protocole propose différents mécanismes de sécurité et laisse beaucoup de liberté sur son implémentation. Il revient donc à l'utilisateur de ce protocole de faire les bons choix pour assurer la sécurité de son réseau. Cette sécurité repose principalement sur celle de la clé de réseau.

Nous avons pu constater que les choix sont nombreux et qu'il faut donc être conscient de leur importance, de leurs enjeux et des conséquences que tel ou tel choix peut avoir.

Cette ressource ne se veut pas exhaustive sur les politiques de sécurité possible et sur les vulnérabilités de ce protocole. Le lecteur souhaitant approfondir ces connaissances à ce propos pourra se référer au papier de NXP sur la sécurité de ZigBee [2] ainsi qu'à la présentation lors de la conférence BlackHat des failles de sécurité possibles de ZigBee et leur exploitation [5].

Références :

[1]: *ZigBee Specification*, ZigBee Alliance, 2015
https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf

[2]: *Maximizing security in ZigBee networks*, NXP Laboratories UK, 2017
https://www.nxp.com/docs/en/supporting-information/MAXSECZBNETART.pdf

[3]: *ZigBee 3.0 Security*, Digi, 2018
https://www.digi.com/support/knowledge-base/zigbee-3-0-security

[4]: *AN1233 : ZigBee Security*, Silicon Laboratories, 2022

[5]: *ZigBee exploited – The good, the bad and the ugly*, Tobias Zillner, Sebastian Strobl, black hat USA 2015
https://www.youtube.com/watch?v=9xzXp-zPkjU&ab_channel=BlackHat
https://www.blackhat.com/docs/us-15/materials/us-15-Zillner-ZigBee-Exploited-The-Good-The-Bad-And-The-Ugly.pdf

Choix d'une topologie de conversion adaptée à un système de dégivrage piézoélectrique

La Revue 3E.I
Culture Sciences de l'Ingénieur

Modar JOMAA[1,3] – Pierre-Etienne LÉVY[1] – Dejan VASIC[1,2] - Marwan ALI[4]
François COSTA[1,3]

Édité le
23/04/2024

école normale supérieure paris-saclay

[1]*Université Paris-Saclay, ENS Paris-Saclay, CNRS, SATIE, 91190 Gif-sur-Yvette, France*
[2]*Université de Cergy-Pontoise, 95031 Cergy-Pontoise, France*
[3]*Université Paris Est Créteil, INSPE, 94000 Créteil, France*
[4]*Safran Tech, groupe de recherche E&E, 78117 Magny-Les-Hameaux, France*

Cette ressource fait partie du N°112 de La Revue 3EI du 2ème trimestre 2024.

Les contraintes environnementales, ainsi que leur impact sur l'opinion publique ont conduit les équipementiers aéronautiques à accélérer la transition énergétique en aéronautique à travers l'avion plus électrique. Nous assistons donc à une augmentation progressive de la place de l'énergie électrique dans les applications embarquées. Ceci se traduit par une tendance à remplacer les systèmes non propulsifs (hydrauliques et pneumatiques) par des chaines de conversion électromécanique. Ces sous-systèmes sont en effet souvent plus performants, dynamiques et précis avec des délais de maintenance plus courts que leurs équivalents hydrauliques.

Le système de dégivrage est un candidat de choix pour cette transition. Il existe plusieurs méthodes de dégivrage qui varient selon la nature de l'énergie de conversion utilisée. On trouve entre autres le flux d'air, les boudins déformables, le système électrothermique ou ETIPS (Electro-thermal Ice Protection Systems), le fluide chimique et les systèmes électromécaniques (système électro-impulsif, système électromécanique expulsif et système piézoélectrique) [1]. Bien que certaines de ces méthodes de dégivrage soient déjà certifiées et équipent certains avions, elles restent très énergivores et ne sont pas adaptées pour tous les types d'avion. La solution envisagée ici de dégivrage piézoélectrique semble être une alternative efficace et plus économique en termes de coût, masse et encombrement.

Cet article expose le fonctionnement d'un système de dégivrage piézoélectrique aéronautique, en particulier ce qui a motivé le choix de son alimentation de puissance. Plusieurs topologies de convertisseurs statiques sont présentées afin de sélectionner la mieux adaptée aux actionneurs piézoélectriques dans le cadre du dégivrage. Une étude par simulation de ces topologies est conduite afin d'identifier les avantages et les inconvénients de chacune d'entre elles pour cette application particulière. Un démonstrateur de la solution retenue a été développé pour valider notre choix.

1 - Principe de fonctionnement

Le principe de fonctionnement du système de dégivrage à base d'actionneurs piézoélectriques consiste à appliquer des vibrations à une certaine fréquence à la structure cible pour fracturer le givre et le faire se décoller de la surface (bords d'attaque). Ceci s'obtient en excitant les fréquences propres de la structure, ce qui permet d'atteindre des amplitudes de vibrations

suffisantes pour assurer la casse et le détachement du givre avec une faible consommation. Les actionneurs piézoélectriques sont fixés (collage ou vissage, Figure 1) sur la face interne d'un tronçon de bord d'attaque (nacelle ou aile) et créent des vibrations quand on les alimente avec une tension alternative.

Dans la littérature, plusieurs études ont été menées sur ce système de dégivrage, avec différents types d'actionneurs piézoélectriques et sur des plages de fréquences variées. Deux configurations d'actionneurs piézoélectriques ont été décrites. La première configuration utilise des transducteurs Langevin, choisis pour leur installation simple à l'aide de boulons et leur risque réduit de défaillance mécanique grâce à leur structure précontrainte qui leur permet de supporter des contraintes plus élevées pendant le fonctionnement. Cependant, l'utilisation de céramiques PZT précontraintes, principalement destinées à exciter des modes de flexion structurelle, peut, dans des meilleurs scénarios, entraîner un délaminage partiel de la glace [2]. La deuxième configuration implique l'utilisation des patchs piézoélectriques, qui peuvent être collés à la structure mécanique. Cette méthode a été plus largement testée dans la littérature et semble présenter des perspectives plus prometteuses. Pour exciter uniquement un mode spécifique, un générateur de tension sinusoïdale correctement conçu doit être utilisé [3].

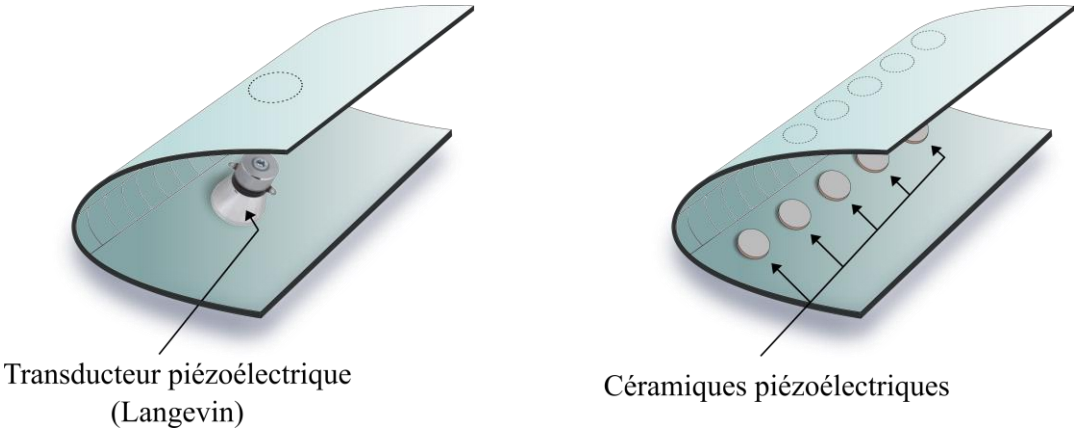


Figure 1 : Configuration d'un système de dégivrage piézoélectrique avec des transducteurs et des patchs nus.

L'alimentation des actionneurs piézoélectriques pose un défi majeur du fait de leur comportement capacitif sur quasiment toute leur plage fréquentielle d'usage et surtout dans le cas du dégivrage par vibrations qui nécessite une fréquence de fonctionnement élevée. Ainsi, afin d'améliorer la performance des actionneurs piézoélectriques et générer leur signal d'alimentation optimal, il est primordial de connaître leur comportement physique. Le circuit équivalent le plus courant qui caractérise un actionneur piézoélectrique autour de sa fréquence de résonance est celui de Van Dyke. Dans le modèle présenté à la figure 2, on identifie, la capacité statique Cs en parallèle avec la branche motionnelle (Lm, Cm, Rm représentatif d'un mode de vibration mécanique). On observe sur la courbe d'impédance un comportement globalement capacitif dû à Cs et de multiples résonances-antirésonances correspondant aux différents modes de résonance mécaniques possibles. Un seul est effectivement utile pour le dégivrage du tronçon présenté, un agrandissement autour de celui-ci à 56 kHz est donné à la Figure 2.

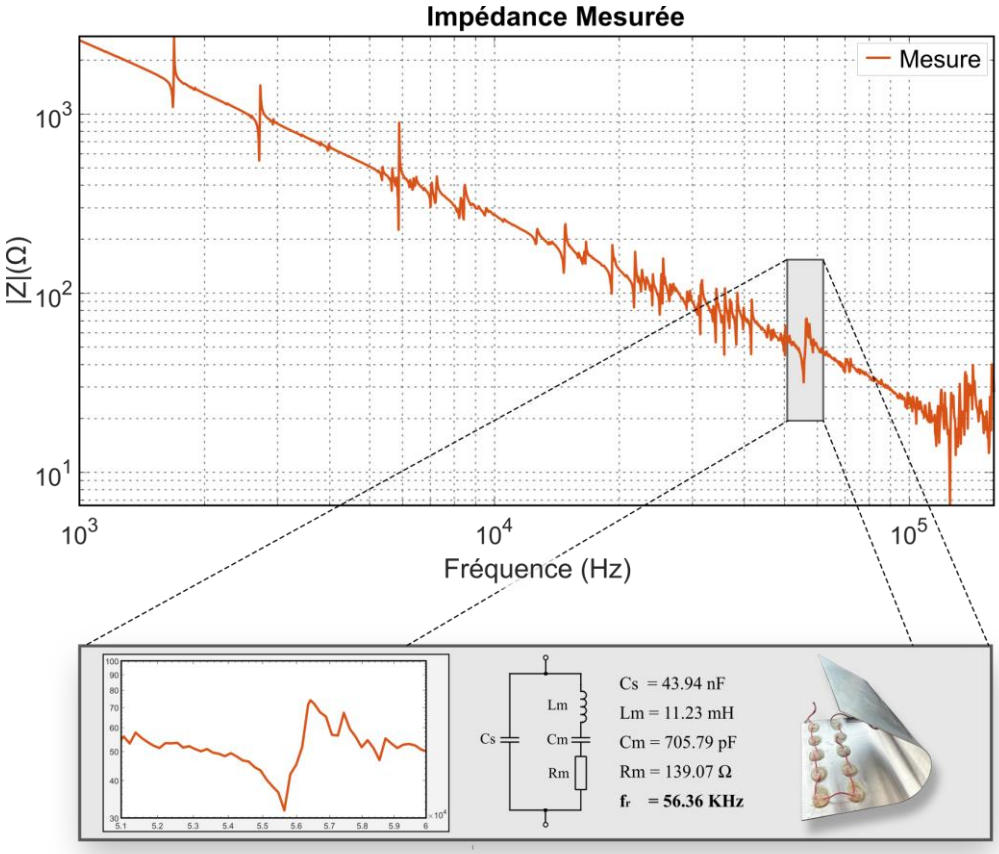


Figure 2 : Caractérisation des actionneurs piézoélectriques par le modèle de Van Dyke autour de la fréquence de résonance à 56.36 kHz.

Ainsi, les actionneurs piézoélectriques collés sur la structure mécanique comportent plusieurs modes de résonances, le mode choisi est celui dont l'amplitude de vibration est maximale, ce qui correspond dans notre cas à la fréquence de résonance à 56.36 kHz. À cette fréquence-là, les paramètres du circuit électrique équivalent sont donnés dans la Figure 2.

2 - Cahier des charges électrique

Le système piézoélectrique ainsi que son alimentation doivent respecter les normes aéronautiques (DO 160), ainsi que les contraintes d'installation de l'équipement (sécurité, encombrement) tout en permettant son bon fonctionnement.

Le système doit pouvoir délivrer une tension sinusoïdale au transducteur afin de n'exciter que le mode désiré et ne pas dégrader l'efficacité de dégivrage du système. Ce dernier étant alimenté par le réseau avionique, la qualité des signaux en termes de pollution harmonique ainsi que de puissance réactive appelée doit être assurée afin de respecter les contraintes CEM du système ainsi que d'éviter un surdimensionnement des éléments du système (transformateur d'isolement, filtres de sortie, filtres CEM).

Enfin, la structure de la nacelle ou de l'aile ne permet pas, au vu des contraintes d'encombrement, de placer le convertisseur au plus près de la charge. Celle-ci sera donc alimentée à travers des câbles de 2 mètres de longueur environ. La modification de l'impédance du système complet aussi bien en termes de valeur qu'en terme de nature (les câbles étant inductifs) est donc un élément important à prendre en compte pour le choix de la structure.

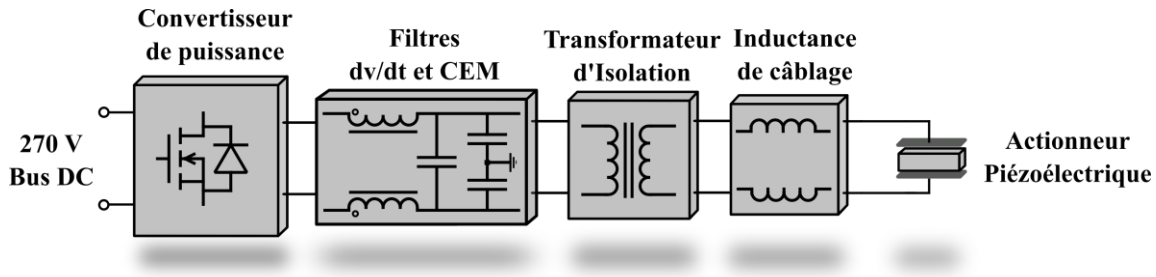


Figure 3 : Synoptique du système complet.

3 - Choix de la topologie

3.1 - État de l'art

Étant donné que le transducteur piézoélectrique a un comportement électrique sensible à la charge mécanique et à la température, il est essentiel de prendre en compte quelques aspects lors de la conception de l'alimentation. Parmi ceux-ci, on trouve la fréquence d'alimentation qui doit correspondre à la fréquence de résonance mécanique de l'actionneur fixé à son support. En effet, à la résonance, la consommation d'énergie réactive est réduite et le transfert de puissance est meilleur. Un autre aspect important, est la qualité du signal d'excitation qui joue un rôle important sur la performance du transducteur piézoélectrique et sa durée de vie [4].

Dans cette problématique, plusieurs techniques sont proposées dans la littérature [5],[6],[7]. Les amplificateurs de puissance linéaires (A, B, AB, …) sont utilisés pour alimenter des charges piézoélectriques car ils permettent de générer des signaux avec des taux de distorsion faibles. Cependant, ils ont un faible rendement et sont souvent volumineux et lourds. Par conséquent, les convertisseurs à découpage sont de plus en plus utilisés et dominent le marché du fait de leur bon rendement et de leur densité de puissance élevée.

Dans la littérature, on trouve plusieurs travaux qui traitent de l'alimentation des actionneurs piézoélectriques à l'aide des onduleurs. Les onduleurs résonants (LC ou LLCC) et les onduleurs à MLI (LC ou LLCC) sont les plus couramment utilisés. D'autres topologies ont été utilisées telles que l'onduleur NPC trois niveaux et l'onduleur de courant [8].

Les inconvénients principaux des onduleurs résonants sont le volume et le poids des éléments magnétiques du filtre résonant et une variation très limitée de la fréquence de fonctionnement.

Afin de pallier ces inconvénients, des onduleurs (LC ou LLCC) à commande MLI ont été proposés [9]. Les inconvénients de la commande MLI sont toujours liés à la fréquence de découpage qui génère des pertes par commutation élevées et des contraintes CEM.

3.2 - Topologies candidates

Onduleur de courant

Pour l'alimentation des actionneurs piézoélectriques, l'utilisation des filtres LC peut détériorer la performance du transducteur en déplaçant la fréquence de résonance [10]. En effet, un actionneur piézoélectrique a un comportement capacitif sur quasiment toute sa plage de fréquences, par conséquence un onduleur de courant peut être une bonne solution pour alimenter le transducteur et avoir un effet favorable sur ses performances.

Étant donné que le domaine d'application est l'aéronautique, l'onduleur est alimenté par le réseau DC à 270 V à travers un hacheur abaisseur (Buck) qui sert à contrôler le courant d'entrée. Le schéma du système complet avec la charge piézoélectrique est illustré sur la Figure 4.

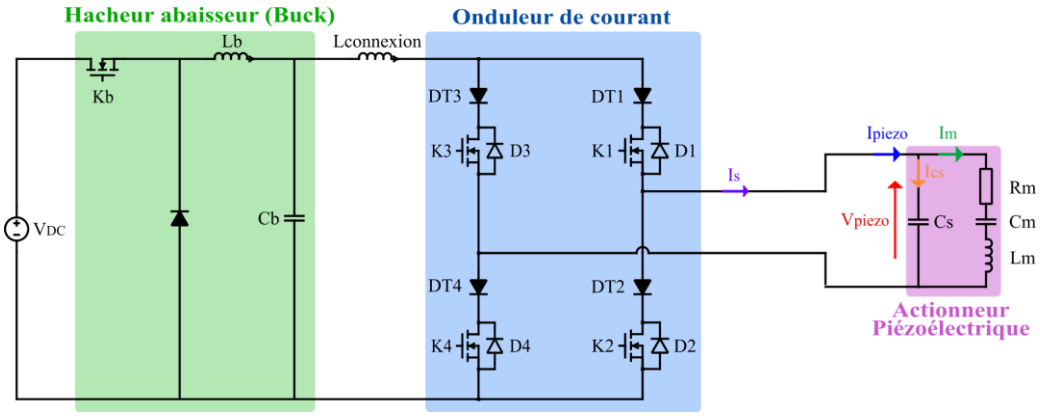


Figure 4 : Schéma du convertisseur complet avec sa charge.

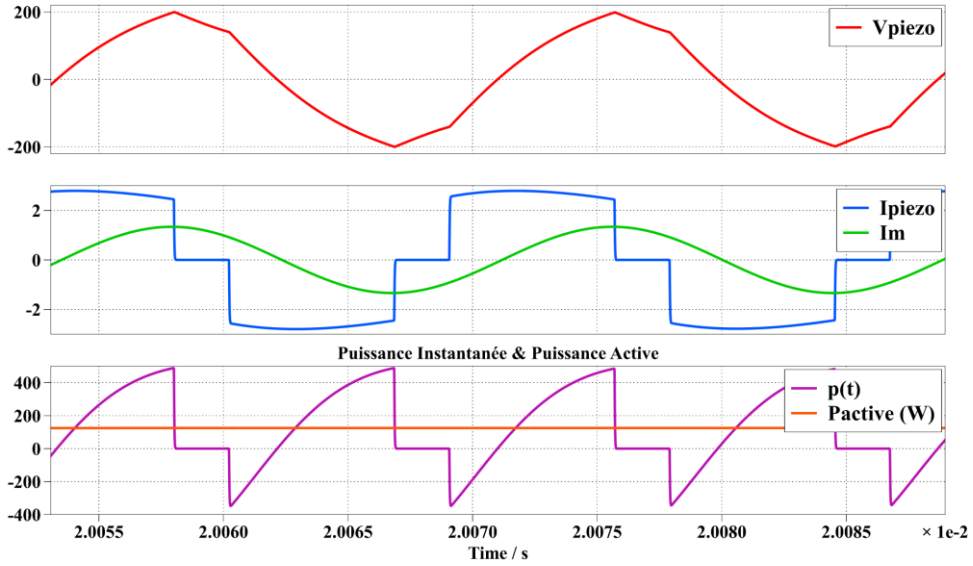


Figure 5 : Formes d'onde en sortie.

L'onduleur est commandé en pleine onde unipolaire à la fréquence de résonance mécanique (résonance série). Puisque l'onduleur impose un courant carré et que la branche motionnelle absorbe un courant I_m sinusoïdal, la capacité statique C_s absorbe la différence I_{Cs} définissant la forme d'onde V_{piezo} (Figure 5).

En effet, afin d'avoir une tension sinusoïdale, il faudrait ajouter une inductance en parallèle avec la capacité C_s et les piloter à leur propre fréquence de résonance. Cependant, cette inductance formerait avec C_s une sorte de filtre passe-bande très sélectif qui limiterait la plage de fréquence de fonctionnement en plus de l'atténuation très forte du fondamental du courant.

Au regard du cahier des charges qui impose une tension de sortie sinusoïdale et une variabilité de la fréquence de commande l'actionneur piézoélectrique, cette topologie ne sera pas retenue. De plus, l'ajout de l'inductance de câblage et le transformateur d'isolement, la charge vue par l'onduleur sera inductive provoquant un conflit de sources et l'apparition de surtensions aux bornes des interrupteurs difficilement contrôlables.

Convertisseur "résonant" avec récupération de l'énergie réactive

Le comportement capacitif des actionneurs piézoélectriques rend leur utilisation délicate car la puissance instantanée peut être beaucoup plus importante que la puissance active. De ce fait, les convertisseurs conventionnels intègrent une inductance qui est, dans la majorité des cas, lourde et volumineuse ce qui rend le système inadapté à certaines applications en particulier en aéronautique. Cette problématique a abouti à de nombreuses études dont l'objectif vise à l'optimisation du facteur de puissance [11], [12]. Le volume du convertisseur peut également être impacté par un surdimensionnement des convertisseurs de puissance dû aux pertes par commutation. Il est alors nécessaire d'introduire des techniques de commutation à zéro de tension qui utilisent dans certains cas des inductances volumineuses [13] .

Dans ce contexte, et afin de pallier ces inconvénients, une topologie consistant en l'ajout d'un circuit shunt auxiliaire sur un bras d'onduleur a retenu notre attention [14]. Le circuit est composé d'un bras de commutation et d'une petite inductance auxiliaire. Celle-ci n'est pas en série dans le circuit de puissance principal et donc elle n'est pas dimensionnée en fonction de la fréquence de résonance de l'actionneur piézoélectrique.

Afin d'adapter cette structure à notre cas d'application, il faut que la tension de sortie soit bipolaire. Pour cela, nous proposons la structure représentée à la Figure 6.

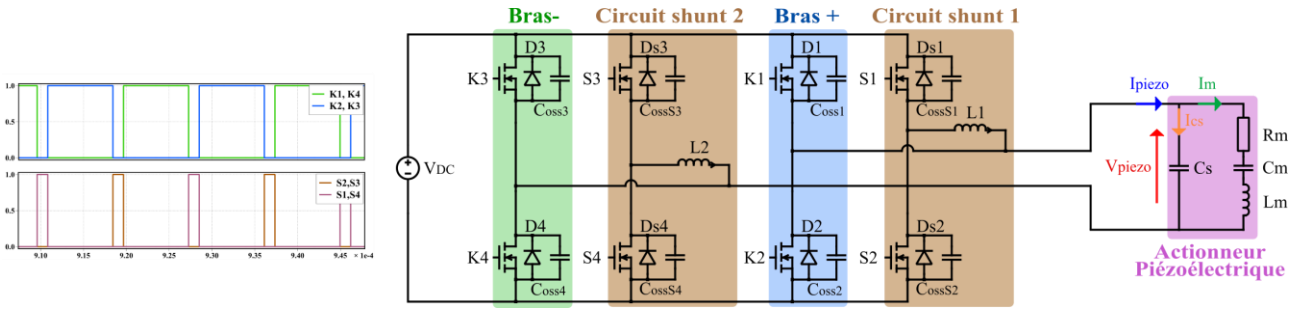


Figure 6 : Configuration en structure bipolaire avec les signaux de commande.

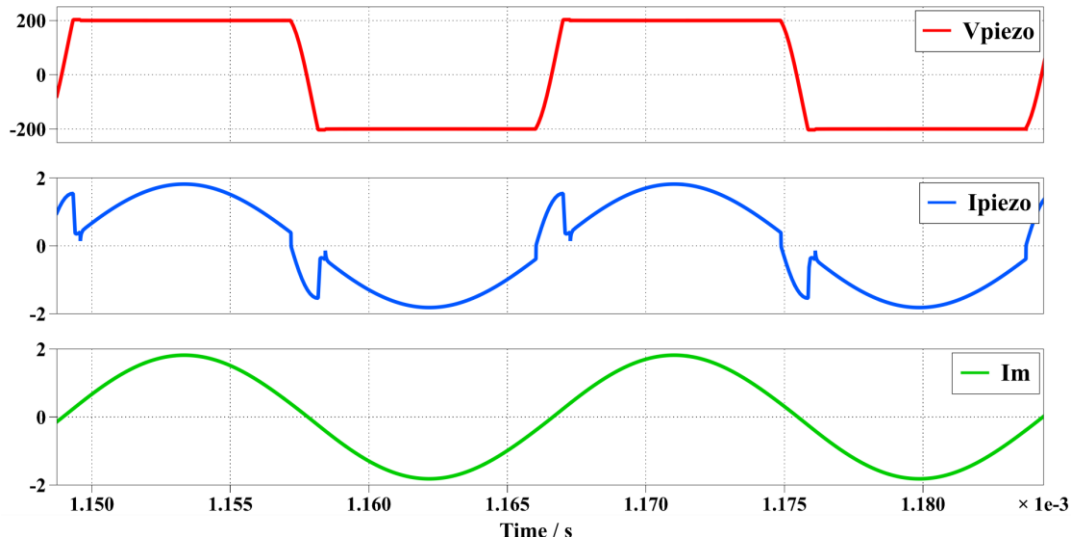


Figure 7 : Formes d'onde en sortie du convertisseur.

Cette structure nous permet de réaliser des commutations douces en mode ZVS dans les quatre cellules de commutation. Elle permet également de récupérer l'énergie réactive provenant de la capacité Cs et ainsi augmenter le rendement du convertisseur. Cependant, si on alimente le transducteur avec une tension carrée, plusieurs modes de résonance mécanique sont excités, conduisant à une possible casse des céramiques piézoélectriques. Afin de résoudre ce problème, il

faudrait mettre en place un filtre passe-bas. Or, avec l'ajout d'une inductance de filtrage, on perd la condition de la commutation en mode ZVS qui dépend de la résonance entre $(L_1 + L_2)$ et C_s. Cela impose un ajustement important (élargissement) du temps mort qui doit tenir compte de la présence d'inductances potentiellement présentes dans la charge (inductance de câblage, inductance de fuite du transformateur d'isolement …) impliquant une transmission peu optimale de la puissance vers la charge.

Cette structure est intéressante pour alimenter directement une charge piézoélectrique à une fréquence fixe et dans le cas où les modes mécaniques parasites sont peu gênants. Or, avec toutes les contraintes de notre cahier des charges, cette structure avec ses inconvénients (tension non sinusoïdale, sensibilité aux inductances parasites de connexion) se montre inadaptée ou de mise en œuvre trop complexe.

Onduleur de tension à circuit résonant auxiliaire (ARCPI)

Plusieurs topologies d'onduleur à commutation douce ont été proposées dans la littérature [15],[16],[17]. Le but d'utiliser ce type d'onduleur est de diminuer les pertes par commutation et également les interférences électromagnétiques que l'on trouve dans les onduleurs traditionnels à commutation dure. L'exemple intéressant des onduleurs à commutation douce de type "Resonant Pole Inverter (RPI)" est l'onduleur avec circuit résonant auxiliaire (ARCPI) [18],[19]. La topologie de cet onduleur, ainsi que ses formes d'ondes théoriques sont illustrées aux Figure 8 et Figure 9 respectivement.

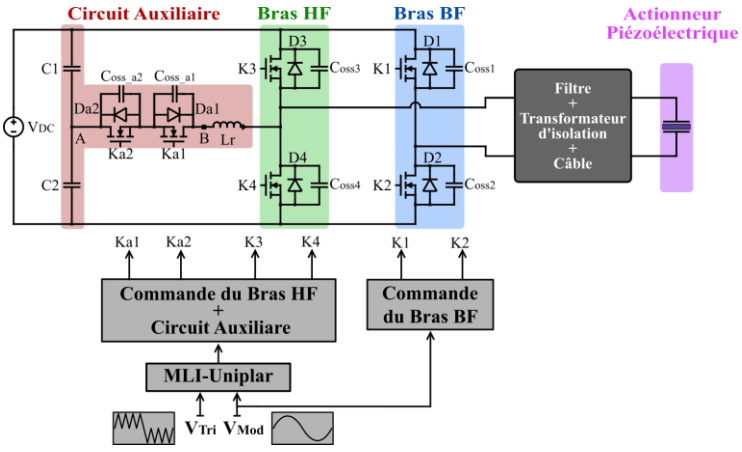


Figure 8 : Topologie de l'onduleur ARCPI.

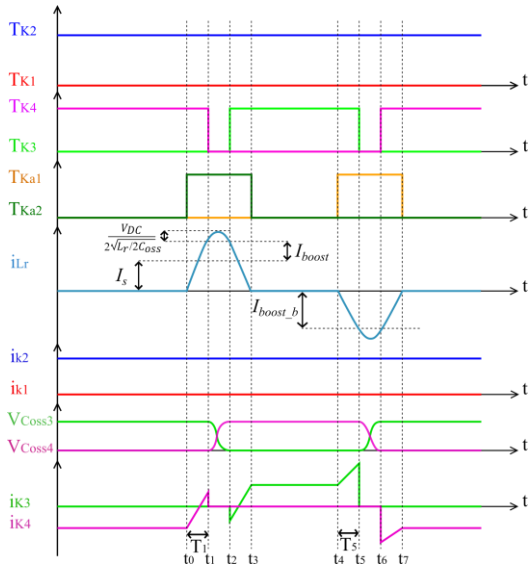


Figure 9 : Formes d'ondes théoriques.

L'onduleur est constitué de deux bras principaux et d'un circuit auxiliaire connecté à un pont diviseur capacitif. Dans le but de limiter au maximum les pertes ainsi que le nombre de composants, un bras est commuté à Basse Fréquence (BF~56 kHz) synchronisé à la fréquence du transducteur et l'autre bras est commuté à Haute Fréquence (HF~1.7 MHz) sur lequel se connecte le circuit auxiliaire. Ce circuit a pour rôle de charger et décharger les capacités parasites C_{oss} afin de faire commuter les transistors du bras HF à ZVS. Ainsi, la loi de commande du circuit auxiliaire implique une commutation à zéro de courant (ZCS) de ses transistors. De plus, puisque le circuit auxiliaire n'est pas dans le circuit de transfert de puissance principal, le calibre en puissance de ses interrupteurs est réduit par rapport à celui des interrupteurs principaux. La commande appliquée est une MLI unipolaire ce qui réduit les harmoniques de la tension de sortie.

Dimensionnement de l'inductance résonante L_r :

Afin d'obtenir des commutations douces en ZVS au niveau du bras HF, il faut que l'énergie stockée dans l'inductance résonante L_r soit suffisante pour charger et décharger les capacités parasites des transistors concernés. De ce fait, les conditions suivantes doivent être respectées :

$$\frac{V_{DC}}{2L_r}T_1 \geq I_{s_{max}} \quad et \quad \frac{V_{DC}}{2L_r}T_5 \geq I_{s_{max}} \tag{1}$$

Avec : I_{Smax} le courant de sortie BF maximal, $T_1 = t_1 - t_0$ et $T_5 = t_5 - t_4$.Ces deux intervalles de temps doivent être toujours inférieurs à $D_{min}T_s$ et $(1 - D_{max})T_s$, où T_s et D sont la période de découpage et le rapport cyclique respectivement.

À partir des inégalités (1), l'inductance résonante L_r doit respecter :

$$L_r \leq \frac{V_{DC}.T_1.V_{s_{max}}}{4P_{s_{max}}} \tag{2}$$

Avec : $P_{s_{max}} = \frac{1}{2}V_{s_{max}}I_{s_{max}}$.

Simulation :

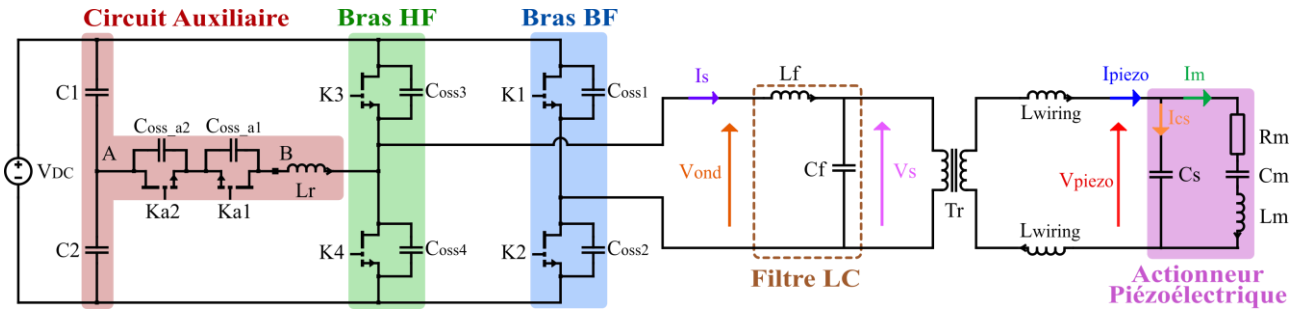


Figure 10 : Schéma complet du système.

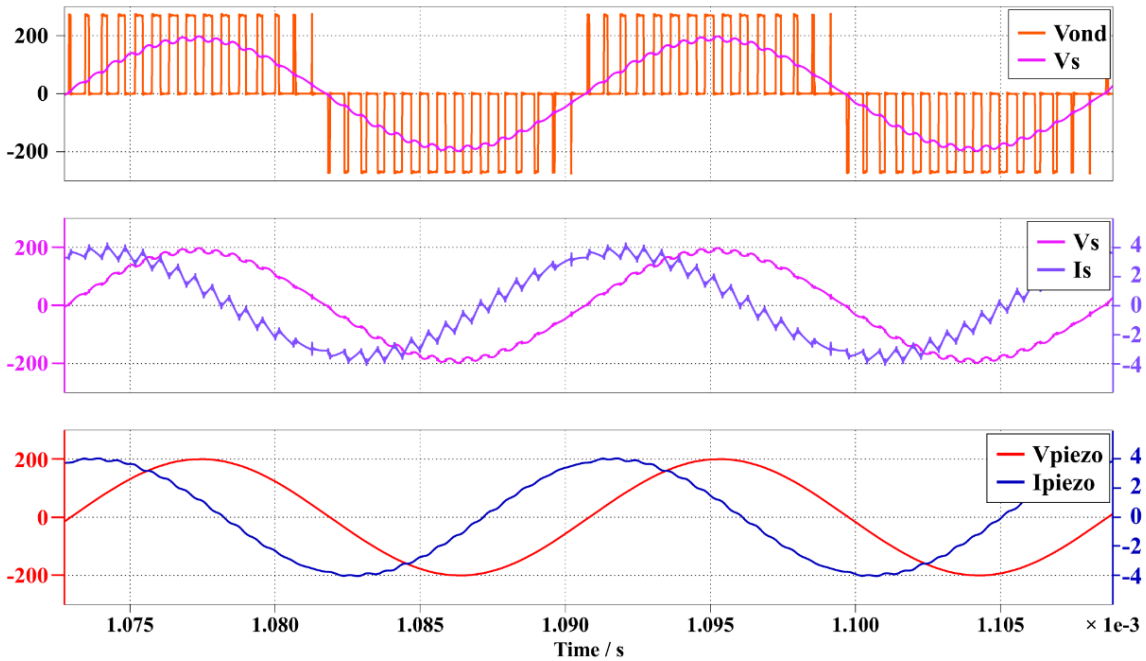


Figure 11 : Formes d'ondes en sortie de l'onduleur.

Le filtre LC de sortie de l'onduleur a été dimensionné de façon à avoir une ondulation de courant à 1.7 MHz inférieur à $20\%.\widehat{I_s}$ et une ondulation de tension inférieur à $10\%.\widehat{V_s}$. Pour cela, une valeur

de $L_f = 53\ \mu\text{H}$ et de $C_f = 5\ \text{nF}$ sont nécessaires, imposant la fréquence de coupure du filtre à $f_{LC} = 300\ kHz$. Ce choix confirme que le signal à 56 kHz n'est pas atténué mais qu'en revanche, la composante de découpage à 1.7 MHz est correctement filtrée. Le transformateur d'isolement utilisé pour la simulation a un rapport de transformation unitaire, une inductance de fuite de $10\ \mu H$ et une inductance magnétisante de $572\ \mu H$.

La simulation est réalisée en prenant en compte l'inductance de câblage ($\text{L}_{\text{câb}_{\text{tot}}} = 2\mu\text{H}$). Comme illustré sur la **Erreur ! Source du renvoi introuvable.**, nous obtenons des formes d'onde sinusoïdales avec un taux de distorsion harmonique (THD) inférieur à 10% pour V_s et I_s, et inférieur à 2% pour V_{piezo} et I_{piezo} . Cependant, un déphasage proche de 70° est observé entre V_s et I_s, ce qui implique une énergie réactive très importante et par conséquent un facteur de puissance faible. Dans le but de compenser l'énergie réactive, un filtre LLCC pourrait être proposé.

Enfin, les avantages que l'onduleur ARCPI présente par rapport aux topologies mentionnées ci-avant (voir Tableau 1), le rendent idéal pour notre application.

Tableau 1 : Comparaison entre les différentes topologies

Topologie	Onduleur de courant	Structure "résonante"	ARCPI
Balayage fréquentiel	☹	😐	☺
Sensibilité à la variation de la charge	☹	😐	☺
Commutation douce	☺	☺	☺
Gestion de l'énergie réactive	☹	☺	😐
Qualité des signaux / Filtrage	☹	😐	☺
Facilité de la commande	☺	😐	😐

4 - Validation expérimentale

Afin de valider les résultats de simulation ainsi que le concept de dégivrage piézoélectrique, une maquette de l'onduleur ARCP a été réalisée. Le choix des semiconducteurs s'est porté sur des composants HEMT GaN bien adaptés à la commutation à très haute fréquence (1.7 MHz) et aux calibres nécessaires pour les interrupteurs (V_{ds} < 300 V, I_{ds} < 10 A). La commande de l'onduleur a été implantée sur une cible FPGA (AMD Xilink ZCU104, cadencé à 250 MHz) ce qui permet un contrôle très précis des instants et durées de commande des interrupteurs (~4 ns). Le test expérimental du système complet a été effectué avec un transformateur d'isolation galvanique (d'un rapport de transformation unitaire, une inductance de fuite de $10\ \mu H$ et une inductance magnétisante de $572\ \mu H$) et un câble de connexion de 2 mètres de longueur ($\sim 2\ \mu H$) (Figure 12).

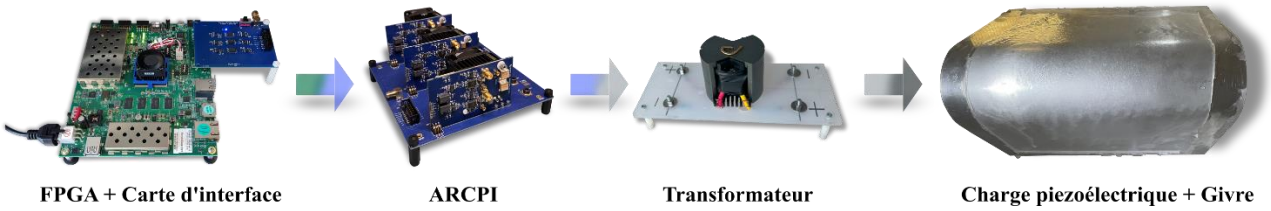


Figure 12 : Illustration du banc expérimental avec la charge piézoélectrique.

Les paramètres des essais expérimentaux de l'ARCPI avec le filtre LC et la charge piézoélectrique sont présentés dans le Tableau 2 suivant :

Tableau 2 : Paramètres du système avec le filtre LC.

Paramètres expérimentaux		
Inductance de filtrage	L_f	$53\ \mu H$
Capacité de filtrage	C_f	$5\ nF$
Inductance de fuite du transformateur	L_{fuite}	$10\ \mu H$
Capacité statique des actionneurs	C_s	$42.7\ nF$
Fréquence de résonance mécanique	f_{rm}	$56.36\ kHz$
Fréquence de découpage pour la MLI	f_{dec}	$1.7\ MHz$
Tension d'alimentation du bus DC	V_{DC}	$270\ V$

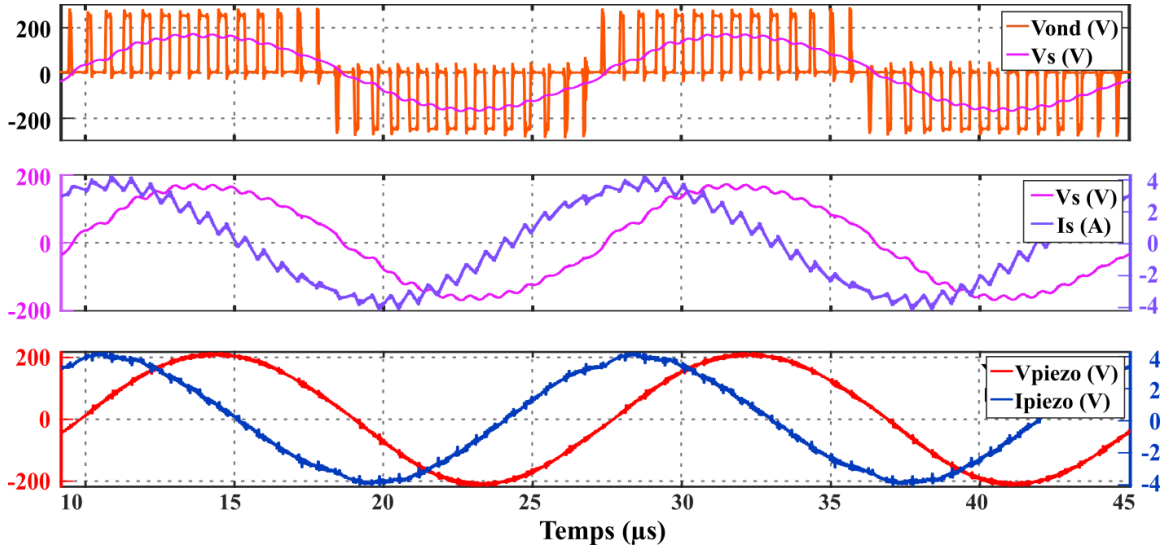


Figure 13 : Résultats d'expérimentation de l'ARCPI avec le filtre LC

Les tests de dégivrage ont été réalisés en utilisant le filtre LC connecté à la charge piézoélectrique autour de la fréquence de résonance à 56.36 kHz pour garantir un niveau de contrainte suffisant pour le délaminage de la glace et avoir un dégivrage complet. Les essais étant réalisés en boucle ouverte, un balayage fréquentiel de ±3 kHz autour de la résonance a été effectué afin de s'assurer d'exciter le mode visé. Les résultats expérimentaux ont montré un dégivrage complet en moins de 5 secondes avec une densité de puissance d'entrée (aux actionneurs) de 74 mW/cm² et un ratio de surface de 0.07 actionneur par cm².

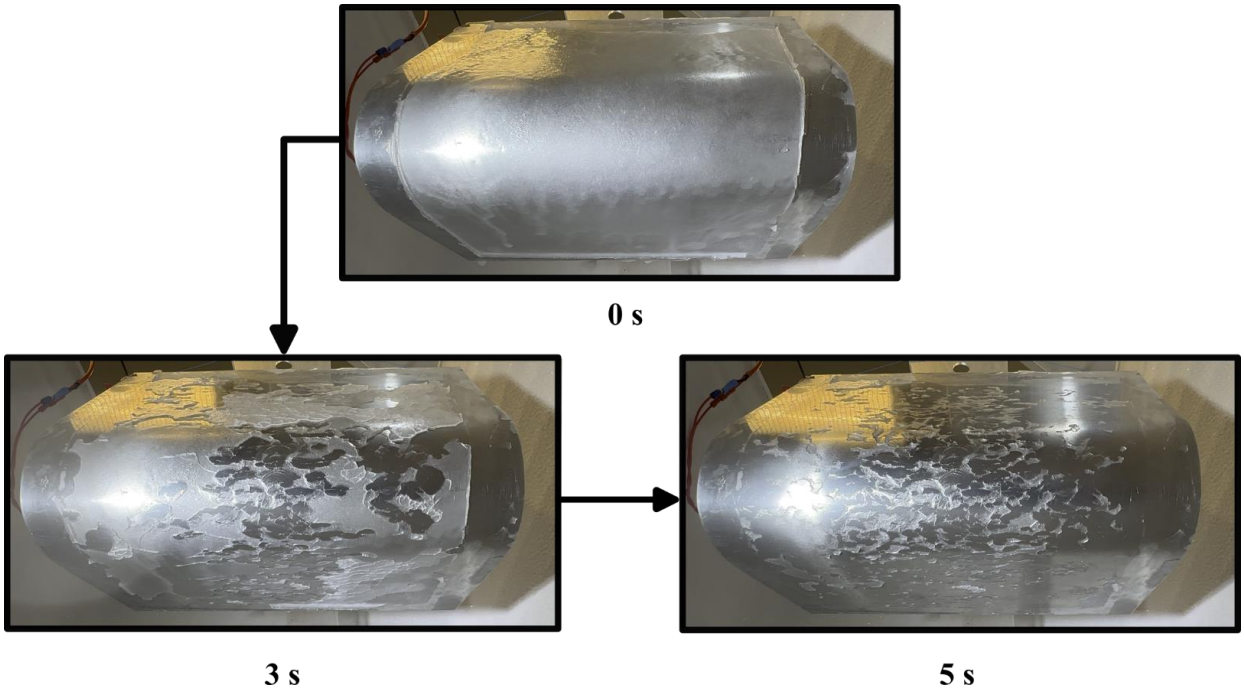


Figure 14 : Résultat d'expérimentation de dégivrage d'un tronçon de bord d'attaque, images extraites de cette vidéo [Laboratoire SATIE].

5 - Conclusion

Dans le cadre du projet de l'avion plus électrique dont cette étude fait partie, une étude de différentes topologies de convertisseurs de puissance pour l'alimentation d'un système de dégivrage piézoélectriques a été menée. Le choix final s'est porté sur l'onduleur ARCP en raison d'un certain nombre d'avantages présentés ci-avant. Une maquette du convertisseur a été réalisée afin de valider expérimentalement les résultats de simulation. L'obtention des performances visées a été rendue possible par le choix de la topologie de conversion la mieux adaptée à la charge et ses contraintes de pilotage, grâce à l'usage de semiconducteurs GaN et grâce à l'usage d'un circuit FPGA très rapide. Le concept de dégivrage piézoélectrique a ainsi été démontré avec une densité de puissance de 74 mW/cm² soit un ratio de surface de 0.07 actionneur par cm², ceci représente un gain d'un ordre de grandeur sur la puissance/cm² par rapport aux solutions actuelles. Il reste toutefois d'autres étapes à franchir pour aboutir à un dispositif réellement embarquable : tests de vieillissement, tests à la foudre et tests CEM.

Références :

[1] Z. Goraj, « An Overview of the De-Icing and Anti-icing Technologies with Prospects for the Future », p. 11.

[2] M. Budinger, V. Pommier-Budinger, G. Napias, et A. Costa da Silva, « Ultrasonic Ice Protection Systems: Analytical and Numerical Models for Architecture Tradeoff », *J. Aircr.*, vol. 53, n° 3, p. 680-690, mai 2016, doi: 10.2514/1.C033625.

[3] M. Jomaa, F. Costa, D. Vasic, P.-E. Lévy, et M. Ali, « Driving Power Supply for Ultrasound Piezoelectric Transducers », in *2023 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*, Venice, Italy: IEEE, mars 2023, p. 1-5. doi: 10.1109/ESARS-ITEC57127.2023.10114888.

[4] Rongyuan Li, N. Frohleke, et J. Bocker, « LLCC-PWM inverter for driving high-power piezoelectric actuators », in *2008 13th International Power Electronics and Motion Control Conference*, Poznan, Poland: IEEE, sept. 2008, p. 159-164. doi: 10.1109/EPEPEMC.2008.4635261.

[5] K. Agbossou, J.-L. Dion, S. Carignan, M. Abdelkrim, et A. Cheriti, « Class D amplifier for a power piezoelectric load », *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 47, nº 4, p. 1036-1041, juill. 2000, doi: 10.1109/58.852087.

[6] H. L. Cheng, C. A. Cheng, C. C. Fang, et H. C. Yen, « Single-switch high power factor inverter for driving piezoelectric ceramic transducer », in *2009 International Conference on Power Electronics and Drive Systems (PEDS)*, Taipei: IEEE, nov. 2009, p. 1571-1576. doi: 10.1109/PEDS.2009.5385732.

[7] Sai Chun Tang et G. T. Clement, « A harmonic cancellation technique for an ultrasound transducer excited by a switched-mode power converter », in *2008 IEEE Ultrasonics Symposium*, Beijing, China: IEEE, nov. 2008, p. 2076-2079. doi: 10.1109/ULTSYM.2008.0513.

[8] S. M. R. Sadriyeh, M. R. Zolghadri, et J. Mahdavi, « Application of a current source inverter for a linear piezoelectric step motor drive », in *4th IEEE International Conference on Power Electronics and Drive Systems. IEEE PEDS 2001 - Indonesia. Proceedings (Cat. No.01TH8594)*, Denpasar, Indonesia: IEEE, 2001, p. 892-897. doi: 10.1109/PEDS.2001.975438.

[9] C. Kauczor et N. Frohleke, « Inverter topologies for ultrasonic piezoelectric transducers with high mechanical Q-factor », in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, Aachen, Germany: IEEE, 2004, p. 2736-2741. doi: 10.1109/PESC.2004.1355265.

[10] N. Ghasemi, F. Zare, A. Ghosh, et C. Langton, « A high frequency current source converter with adjustable magnitude to drive high power piezoelectric transducers », in *2012 15th International Power Electronics and Motion Control Conference (EPE/PEMC)*, Novi Sad, Serbia: IEEE, sept. 2012, p. DS1b.5-1-DS1b.5-4. doi: 10.1109/EPEPEMC.2012.6397206.

[11] B. Ducharne, L. Garbuio, M. Lallart, D. Guyomar, G. Sebald, et J.-Y. Gauthier, « Nonlinear Technique for Energy Exchange Optimization in Piezoelectric Actuators », *IEEE Trans. Power Electron.*, vol. 28, nº 8, p. 3941-3948, août 2013, doi: 10.1109/TPEL.2012.2227813.

[12] Y.-P. Liu et D. Vasic, « Small power step-up converter for driving flapping wings of the micro robotic insects », in *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, Raleigh, NC, USA: IEEE, sept. 2012, p. 41-46. doi: 10.1109/ECCE.2012.6342414.

[13] W.-C. Su et C.-L. Chen, « ZVS for PT Backlight Inverter Utilizing High-Order Current Harmonic », *IEEE Trans. Power Electron.*, vol. 23, nº 1, p. 4-10, janv. 2008, doi: 10.1109/TPEL.2007.911831.

[14] D. Vasic et F. Costa, « Energy recovery power supply for piezoelectric actuator », in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, Dallas, TX, USA: IEEE, oct. 2014, p. 1440-1445. doi: 10.1109/IECON.2014.7048691.

[15] D. M. Divan, « The resonant DC link converter-a new concept in static power conversion », *IEEE Trans. Ind. Appl.*, vol. 25, nº 2, p. 317-325, avr. 1989, doi: 10.1109/28.25548.

[16] D. C. Katsis, J.-Y. Choi, D. Boroyevich, et F. C. Lee, « Drive Cycle Evaluation of A Soft-Switched Electric Vehicle Inverter », p. 6.

[17] J.-Y. Lim, J. Soh, et R.-Y. Kim, « An Improved Single-Phase Zero-Voltage Transition Soft-Switching Inverter with a Subtractive Coupled Inductor Auxiliary Circuit », in *2016 IEEE Vehicle Power and Propulsion Conference (VPPC)*, Hangzhou, China: IEEE, oct. 2016, p. 1-6. doi: 10.1109/VPPC.2016.7791610.

[18] R. W. De Doncker et J. P. Lyons, « The auxiliary resonant commutated pole converter », in *Conference Record of the 1990 IEEE Industry Applications Society Annual Meeting*, Seattle, WA, USA: IEEE, 1990, p. 1228-1235. doi: 10.1109/IAS.1990.152341.

[19] M.-C. Jiang, W.-S. Wang, H.-K. Fu, et K. Wu-Chang, « A novel single-phase soft-switching unipolar PWM inverter », in *8th International Conference on Power Electronics - ECCE Asia*, Jeju, Korea (South): IEEE, mai 2011, p. 2874-2879. doi: 10.1109/ICPE.2011.5944785.

Conception des onduleurs de tension : Comparaison entre une structure classique et une structure multiniveau NPC

La Revue 3E.I
Culture Sciences de l'Ingénieur

Édité le
06/05/2024

école normale supérieure paris—saclay

Adrien VOLDOIRE[1]

[1] *Université Paris-Saclay, CentraleSupélec, CNRS, Laboratoire de Génie Electrique et Electronique de Paris (GeePs)*

Cette ressource fait partie du N°112 de La Revue 3EI du 2ème trimestre 2024.

Cet article résume des considérations essentielles pour la conception des onduleurs de tension, qu'ils soient utilisés pour l'entrainement de machines triphasées ou pour la connexion à des réseaux de distribution ou embarqués. L'électrification massive des usages amenant un besoin d'efficacité énergétique accru, l'utilisation d'une structure d'onduleur classique 2 niveaux est comparée à une structure multiniveau dite Neutral Point Clamped (NPC). La comparaison devant reposer sur des indicateurs de rendement et de densité de puissance, des modèles analytiques sont proposés pour les deux structures. Ces modèles présentent des éléments clés pour le dimensionnement des transistors, diodes, inductances et condensateurs. Finalement, une analyse comparative sur trois dimensionnements permet de comprendre les enjeux sous-tendant le dimensionnement des onduleurs de tension et l'usage de structures multiniveaux.

1 - Introduction

Les onduleurs de tension sont indispensables pour réaliser de nombreuses fonctions centrales dans le cadre de l'électrification de usages et de la transition énergétique. On les retrouve par exemple dans l'intégration des énergies renouvelables sur le réseau, pour le pilotage des machines électriques synchrones et asynchrones, dans les alimentations sans interruption (ASI en français, UPS en anglais) des serveurs, ou encore dans la gestion de l'énergie électrique embarquée. L'électrification des usages amène à l'augmentation des puissances converties, en environnement souvent contraint, tout en améliorant les rendements énergétiques.

La réalisation de ces onduleurs devient alors critique, ce qui demande au concepteur d'innover en proposant des structures plus performantes et en choisissant rigoureusement les composants actifs et passifs. Ainsi, cet article propose une comparaison entre les performances d'un onduleur 2 niveaux, structure la plus classique, et celles d'un onduleur 3 niveaux Neutral Point Clamped (NPC), une des structures multiniveaux la plus mature dans l'industrie. Cette comparaison est réalisée à partir de modèles de pertes des composants actifs et passifs, dans le but d'évaluer correctement le rendement de chaque structure.

2 - Structures d'onduleur 2 et 3 niveaux

La Figure 1Figure 1 : Onduleur de tension triphasé avec filtrage**Erreur ! Source du renvoi introuvable.** présente le schéma de l'onduleur de tension triphasé à 2 niveaux, intégrant 6 interrupteurs, 3 inductances de lissage et deux condensateurs de découplage. L'inductance de lissage n'est pas nécessaire dans le cas de l'entrainement d'une machine. Elle peut être complétée

par des condensateurs et d'autres inductances pour augmenter l'ordre du filtre dans le cas d'une connexion au réseau avec des contraintes de pollution harmonique sévère.

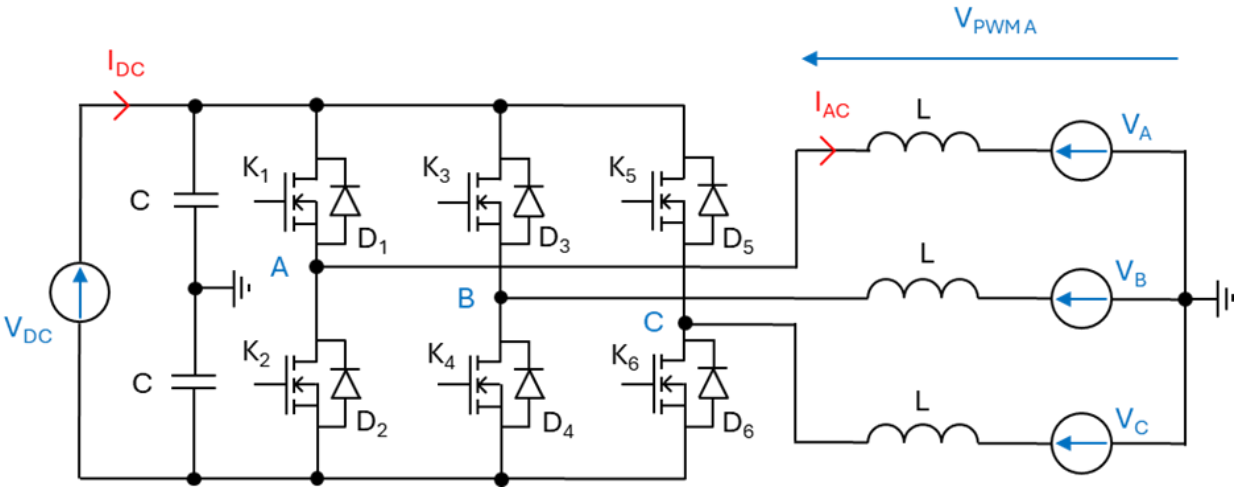


Figure 1 : Onduleur de tension triphasé avec filtrage

Les onduleurs multiniveaux sont apparus ces dernières décennies pour répondre aux cahiers des charges mettant en jeu des moyennes et fortes tensions. Le principe des structures multiniveaux [1] consiste à offrir plusieurs valeurs de tensions commutées, qui sont limitées à $+V_{DC}$ et $-V_{DC}$ dans le cas des onduleurs à 2 niveaux. Ainsi, une troisième valeur de tension, de valeur 0, est disponible en 3 niveaux. Cette valeur supplémentaire permet d'améliorer la qualité du signal côté alternatif, et donc de relâcher les contraintes sur le dimensionnement du filtre.

Un autre avantage des structures multiniveaux est de pouvoir répartir la tension du bus DC sur plusieurs interrupteurs au lieu d'un seul. Cela permet de choisir des interrupteurs de plus petit calibre en tension, ayant généralement de meilleures propriétés tout en étant plus facilement approvisionnables. Parmi les structures à 3 niveaux disponibles [2], [3], [4], la structure Neutral Point Clamped (NPC), représentée sur la Figure 2, est étudiée ici.

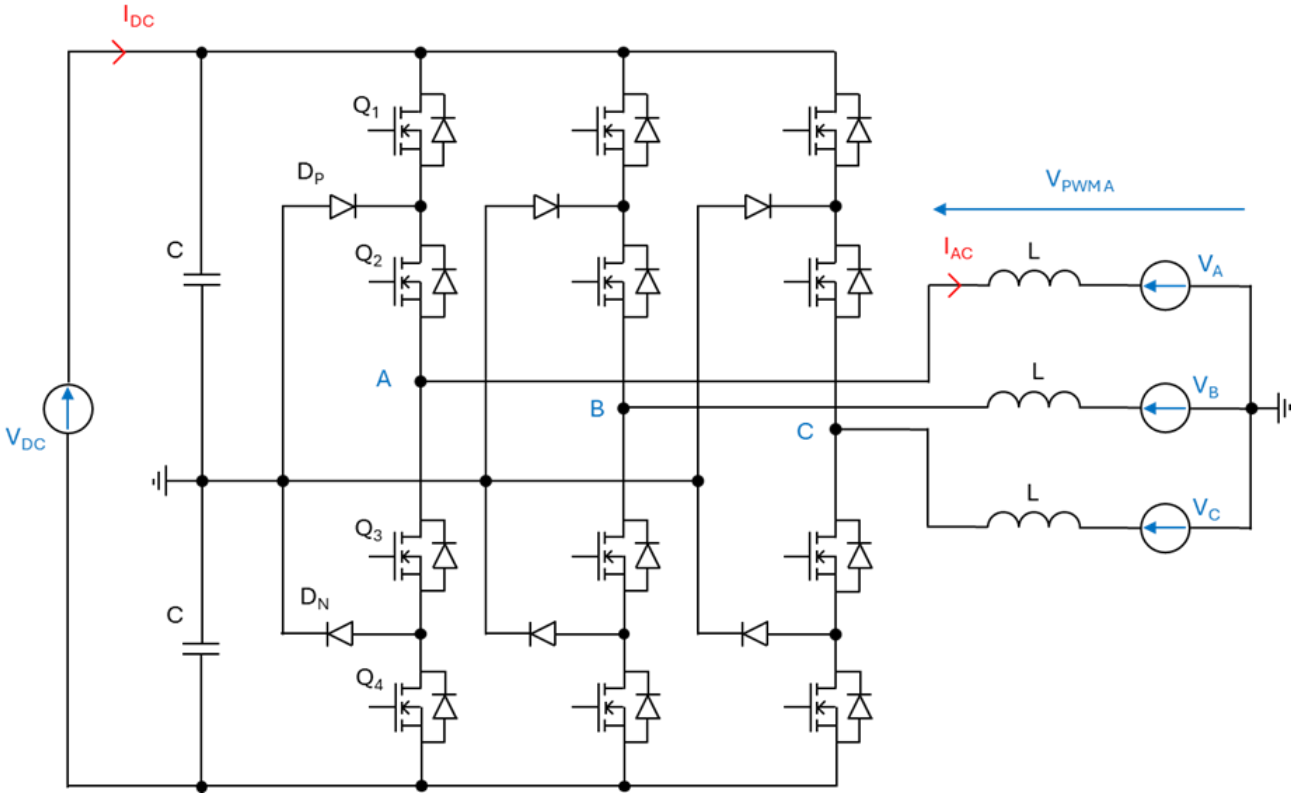


Figure 2 : Structure Neutral Point Clamped 3 niveaux "NPC"

La structure NPC se commande naturellement en "Level Shift" ou "Phase Disposition", comme indiqué sur la Figure 3. La modulante est commune à tous les interrupteurs. La comparaison entre la modulante et la porteuse triangulaire en bleu contrôle les commutations complémentaires des interrupteurs Q_1 et Q_3. La comparaison entre la modulante et la porteuse triangulaire en vert contrôle réciproquement Q_2 et Q_4. L'analyse de l'état des interrupteurs, 1 signifiant passant et 0 bloqué, permet de tracer l'évolution de la tension simple, qui prend bien 3 valeurs : $+V_{DC}/2$, 0 et $-V_{DC}/2$.

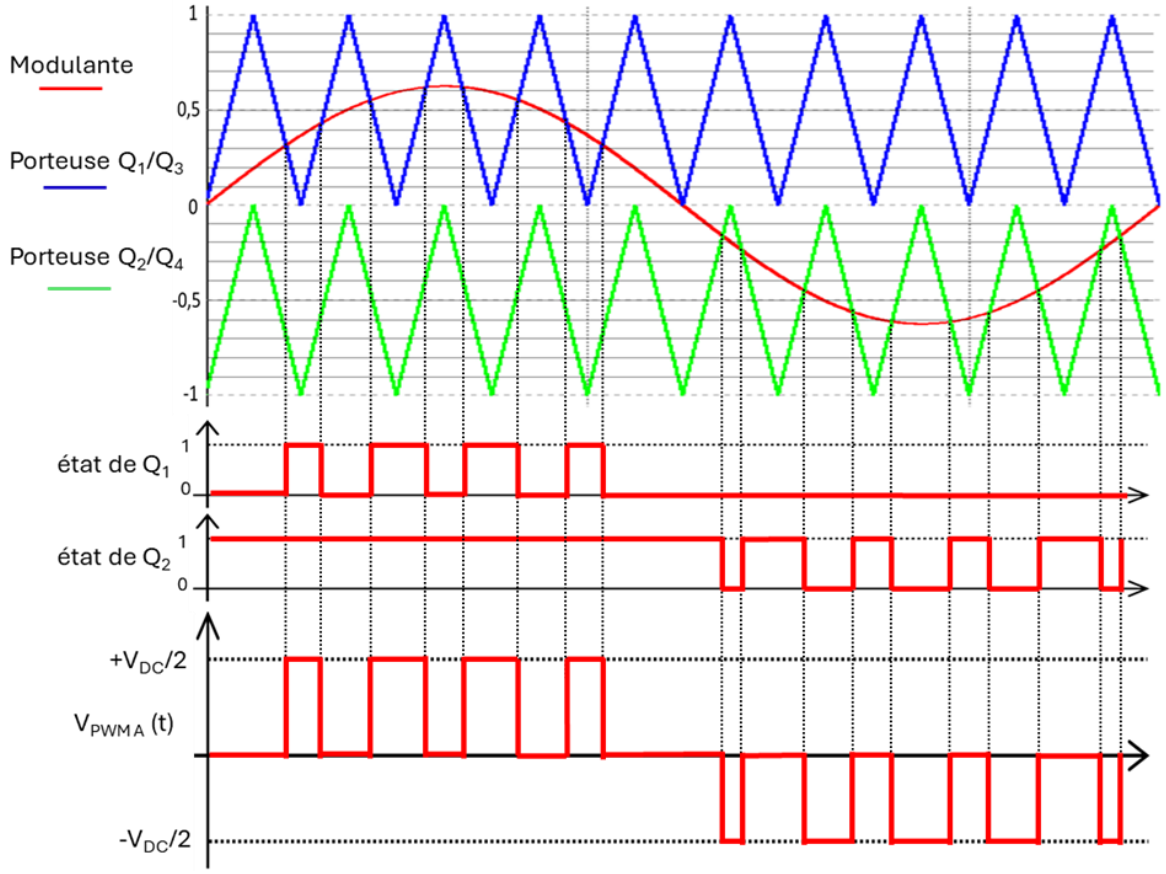


Figure 3 : Illustration de la commande en Phase Disposition, de l'état des interrupteurs et de la tension simple d'un onduleur NPC

3 - Dimensionnement des interrupteurs de puissance

3.1 - Choix du calibre et de la technologie d'interrupteur

La réalisation d'un onduleur de tension impose le choix d'un interrupteur bidirectionnel en courant, unidirectionnel en tension. Pour réaliser cette fonction, une solution simple consiste à choisir un transistor MOSFET ou à associer un IGBT et une diode en antiparallèle. Bien qu'un transistor MOSFET possède déjà intrinsèquement une diode en antiparallèle, aussi appelée diode 'body', on ajoutera physiquement une diode, car la diode intrinsèque est de mauvaise qualité. L'hypothèse est faite que le transistor conduira les courants positifs, et la diode les courant négatifs.

Le choix d'un interrupteur se fait d'abord en regard des tensions mises en jeu. Le calibre en tension de l'interrupteur choisi doit être supérieur à la tension qu'il devra supporter, en raison des surtensions à chaque blocage. Le besoin critique de fiabilité dans de nombreux domaines, comme l'aéronautique ou l'automobile, amène à choisir un calibre en tension au moins égal à deux fois la tension à supporter.

$$V_{calibre} \geq 2V_{DC} \quad (1)$$

Pour une tension nominale de 540 V, le calibre en tension de l'interrupteur choisi est donc de 1200 V. Les transistors 1200 V disponibles sur le marché sont soit des IGBT en silicium, soit des MOSFET en carbure de silicium (SiC). Dans ce contexte, le choix d'interrupteur s'est porté sur les MOSFET SiC, qui ont comme propriétés remarquables des énergies de commutation bien plus faibles que pour les IGBT. Les diodes utilisées en antiparallèle seront des Schottky SiC, dont les pertes par recouvrement seront considérées négligeables devant les autres pertes.

Á la différence du choix du calibre en tension qui s'impose rapidement, le calibre en courant est bien plus difficile à choisir car il dépend de beaucoup de facteurs. Trois contraintes guident ce choix :

- Le rendement : les pertes issues des semi-conducteurs contribuent souvent majoritairement dans le calcul du rendement d'un convertisseur.
- La température de jonction : la température de jonction du transistor ne doit pas dépasser la température préconisée par le constructeur, souvent fixée entre 125°C et 150°C en fonction de la contrainte de fiabilité. Elle dépend de la quantité de pertes générée par les interrupteurs et du système de refroidissement mis en place pour évacuer ces pertes.
- Le prix : le prix est directement lié au calibre en courant d'un transistor et plus généralement à la surface de semiconducteur utilisée pour réaliser l'interrupteur.

Il apparait alors que le choix du calibre en courant résulte d'un compromis impliquant différentes variables d'entrée (dissipateur et fréquence de découpage), et plusieurs contraintes et performances critiques du convertisseur. La figure 4 montre deux technologies d'intégration de transistor : un composant discret pour les faibles courants et un module pour les forts courants. Par défaut, on peut choisir le calibre en courant à une valeur égale ou supérieure au courant efficace circulant dans le transistor.

(a) *(b)*

Figure 4 : MOSFET SiC du fabricant Cree : (a) packaging discret (TO247) et (b) packaging module

3.2 - Pertes par conduction dans la structure 2 niveaux

Les pertes par conduction existent car la résistance à l'état passant d'un interrupteur n'est jamais nulle. Pour évaluer les pertes en conduction, il est nécessaire de connaitre le courant efficace et moyen dans les interrupteurs MOSFET et diode. Ces pertes sont calculées dans un seul interrupteur, elles sont identiques pour les 5 autres interrupteurs par symétrie des formes d'onde. La forme d'onde du courant dans un interrupteur (association d'un MOSFET et d'une diode) est donnée sur la Figure 5. La Figure 5 considère une fréquence de découpage très faible à des fins d'illustration pédagogique. Dans la réalité, on considère une fréquence de découpage au moins 100 fois plus grande que la fréquence fondamentale, ce qui permet un certain nombre d'hypothèses simplificatrices, dont le fait de négliger l'impact des ondulations de courant haute fréquence sur les pertes.

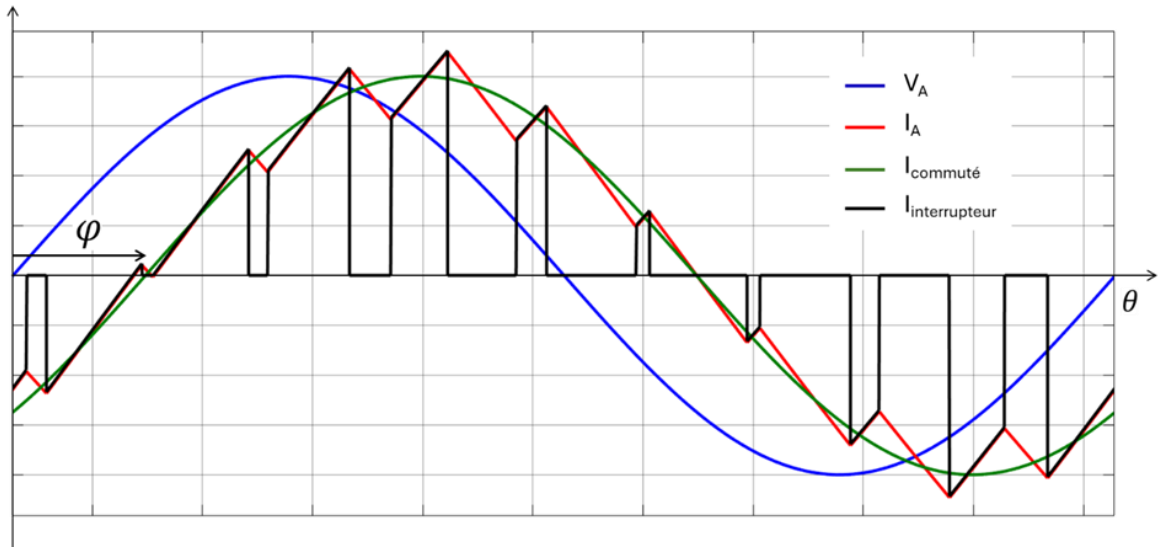


Figure 5 : Forme d'onde du courant dans un interrupteur pour un déphasage faible

Le temps de conduction d'un transistor dépend du rapport cyclique (2), variable au cours du temps en MLI bipolaire. La variable m définie par (3) représente l'indice de modulation en amplitude de l'onduleur, soit le rapport entre la valeur crête de la tension de sortie et la tension commutée, valant la moitié de la tension DC dans le cas de l'onduleur triphasé. Le courant commuté (4) est assimilé à un signal sinusoïdal parfait, avec φ le déphasage courant-tension en sortie d'onduleur entre les interrupteurs et le filtre.

$$\alpha(\theta) = \frac{1 + m.\sin(\theta)}{2} \tag{2}$$

$$m = \frac{2.V_{AC}\sqrt{2}}{V_{DC}} \tag{3}$$

$$I_{commuté}(\theta) = I_{AC}\sqrt{2}\,\sin(\theta - \varphi) \tag{4}$$

Deux cas de figures sont ici possibles. Si l'interrupteur est composé uniquement d'un MOSFET, sans diode additionnel en anti-parallèle, l'expression du courant efficace dans le MOSFET est exprimée par (5) en fonction du courant commuté pondéré par le rapport cyclique $\alpha(\theta)$, et conduit à l'expression (6).

$$I_{T\,efficace} = \sqrt{\frac{1}{2\pi}\int_0^{2\pi} \alpha(\theta)\, I_{commuté}(\theta)^2 d\theta} \tag{5}$$

$$I_{T\,efficace} = \frac{I_{AC}\sqrt{2}}{2} \tag{6}$$

Si on ajoute une diode en antiparallèle au transistor, on suppose alors que le courant positif passe par le MOSFET, et le courant négatif passe par la diode. Le transistor conduit alors sur une demi-période à la fréquence du réseau quand le courant commuté est positif, soit entre φ et $\varphi + \pi$ (7). Une fois l'intégrale calculée, le courant efficace s'exprime analytiquement selon (8). Le même raisonnement est appliqué pour la diode complémentaire, en prenant en compte que son temps de conduction correspond à $1 - \alpha(\theta)$, ce qui conduit à (9) et (10).

$$I_{T\,efficace} = \sqrt{\frac{1}{2\pi}\int_{\varphi}^{\varphi+\pi} \alpha(\theta)\, I_{commuté}(\theta)^2 d\theta} \quad (7)$$

$$I_{T\,efficace} = I_{AC}\sqrt{2}\sqrt{\frac{1}{8} + \frac{m}{3\pi}\cos(\varphi)} \quad (8)$$

$$I_{D\,efficace} = \sqrt{\frac{1}{2\pi}\int_{\varphi}^{\varphi+\pi} (1-\alpha(\theta))\, I_{commuté}(\theta)^2 d\theta} \quad (9)$$

$$I_{D\,efficace} = I_{AC}\sqrt{2}\sqrt{\frac{1}{8} - \frac{m}{3\pi}\cos(\varphi)} \quad (10)$$

Le calcul des pertes dans une diode requiert également la connaissance du courant moyen, qui s'exprime selon (11) et (12).

$$I_{D\,moyen} = \frac{1}{2\pi}\int_{\varphi}^{\varphi+\pi} (1-\alpha(\theta))\, I_{commuté}(\theta) d\theta \quad (11)$$

$$I_{D\,moyen} = \frac{I_{AC}\sqrt{2}}{2\pi}\left(1 - \frac{m.\pi}{4}\cos(\varphi)\right) \quad (12)$$

La résistance à l'état passant $R_{DS\,on}$ d'un transistor MOSFET SiC dépend fortement de la température de jonction de la puce, comme montré sur la Figure 6. La résistance est pratiquement multipliée par 2 en passant de 25°C à 150°C. Il convient alors de choisir la valeur de résistance pour le point de fonctionnement à puissance maximale, ce qui conduit généralement à une température de jonction supérieure à 100°C.

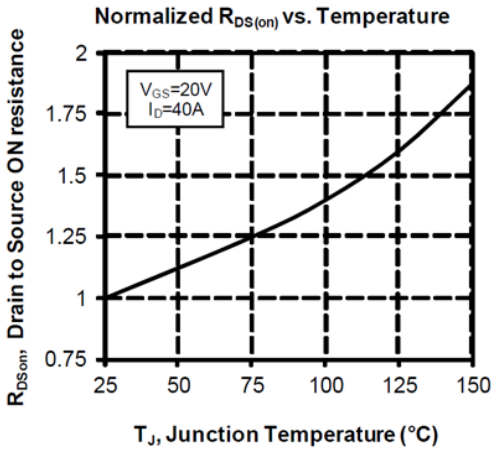


Figure 6 : Résistance normalisée à l'état passant d'un MOSFET en fonction de la température de jonction (données Microsemi)

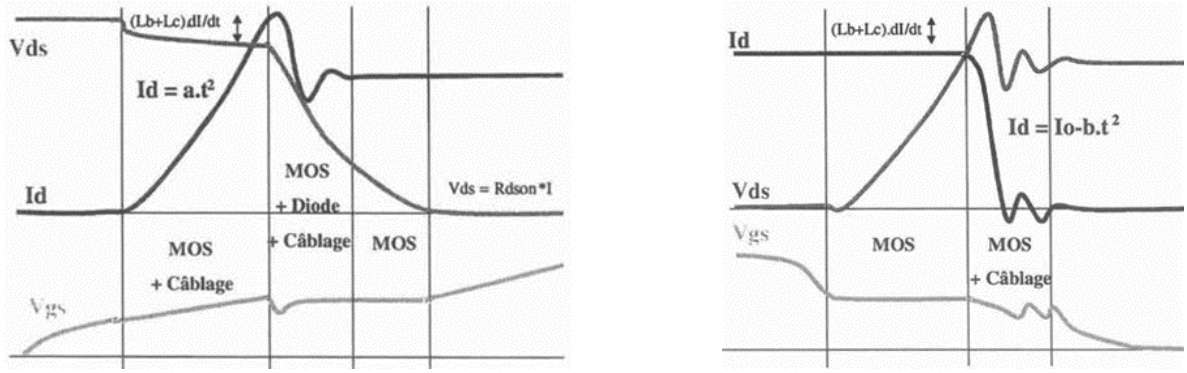
Les pertes par conduction dans un transistor MOSFET dépendent uniquement de la résistance à l'état passant du transistor (13). A la différence des IGBT ou des diodes évoquées précédemment, il n'y a pas d'effet de seuil sur la tension. Les pertes en conduction dans une diode s'expriment selon (14), en prenant cette fois-ci en compte l'effet de tension de seuil V_{D0}. R_{D0} représente le coefficient directeur de la droite caractéristique de la diode dans le plan I-V.

$$P_{T\,conduction} = R_{DS\,on}.{I_{T\,efficace}}^2 \quad (13)$$

$$P_{D\ conduction} = V_{D0}.I_{D\ moyen} + R_{D0}.I_{D\ efficace}{}^{2} \quad (14)$$

3.3 - Pertes par commutation dans la structure 2 niveaux

Chaque commutation, ouverture et fermeture, génère des pertes au sein des transistors. Cette énergie perdue s'explique par le fait que la tension aux bornes de l'interrupteur n'est pas nulle quand le courant s'établit et s'éteint dans le transistor. Les mécanismes de commutation, illustrés par la Figure 7, sont expliqués en détail dans plusieurs travaux [5].



(a) *(b)*

Figure 7 : Mécanismes de commutation d'un MOSFET : (a) fermeture (ON) et (b) ouverture (OFF) [5]

Les énergies de commutation E_{on} et E_{off} dépendent notamment du courant et de la tension commutée. Dans le cas de l'onduleur de tension à 2 niveaux, la tension commutée est constante : c'est la tension V_{DC}. Comme les énergies de commutation sont caractérisées par le constructeur pour une tension d'essai V_{ref}, et qu'elles varient linéairement avec la tension, un coefficient linéaire est pris en compte.

En revanche, le courant commuté varie sinusoïdalement sur une période du réseau. Il est alors nécessaire de modéliser la variation des énergies de commutation en fonction du courant. Cette variation est donnée par le constructeur du module, elle est ici modélisée par une interpolation quadratique au moyen de coefficients a_T, b_T, et c_T. L'expression indiquant la variation des énergies de commutation avec le courant et la tension est indiquée par (15).

$$(E_{on} + E_{off})(I,V) = \frac{V}{V_{ref}}(a_T I^2 + b_T I + c_T) \quad (15)$$

Les énergies de commutation dépendent également de l'environnement de commutation du transistor, qui peut accélérer ou ralentir la commutation. L'environnement de commutation regroupe la résistance de grille, les inductances parasites, le routage, la température, etc. Ces paramètres ne sont pas les mêmes entre l'essai réalisé par le constructeur pour construire la datasheet et le convertisseur réalisé. Il convient alors soit de caractériser ces énergies de commutation dans l'environnement de l'utilisateur, par exemple via un test en Double Pulse [6], soit d'admettre une imprécision réalisée dans l'estimation des pertes par commutation.

Les pertes en commutation dans un transistor se calculent sur une demi-période du réseau car les transistors ne commutent pas quand le courant est négatif. En effet, la diode anti-parallèle (intrinsèque ou ajoutée) conduisant pendant le temps mort précédant le blocage ou suivant l'amorçage, la tension aux bornes de l'interrupteur reste nulle pendant la commutation, ce qui annule les pertes.

Les pertes moyennes par commutation s'expriment alors selon (16). En considérant que le courant commuté est sinusoïdal (4), et en utilisant la variation quadratique de l'énergie en fonction du courant défini en (15), les pertes en commutation s'expriment analytiquement de la manière suivante (17).

$$P_{T\ commuation} = \frac{1}{2\pi}\int_{\varphi}^{\varphi+\pi} \left(E_{on}(\theta) + E_{off}(\theta)\right) f_{dec}\, d\theta \tag{16}$$

$$P_{T\ commuation} = \frac{V_{DC}}{V_{ref}} f_{dec}\left(\frac{a_T}{2} {I_{AC}}^2 + \frac{\sqrt{2}.b_T}{\pi} I_{AC} + \frac{c_T}{2}\right) \tag{17}$$

3.4 - Pertes dans la structure NPC

On appelle Q_1 et Q_4 les interrupteurs externes, Q_2 et Q_3 les interrupteurs internes, D_p et D_n les diodes de clamp. En analysant les formes d'onde présentées sur la **Erreur ! Source du renvoi introuvable.**, il apparait que l'interrupteur interne aura plus de pertes en conduction, alors que l'interrupteur externe aura plus de pertes en commutation.

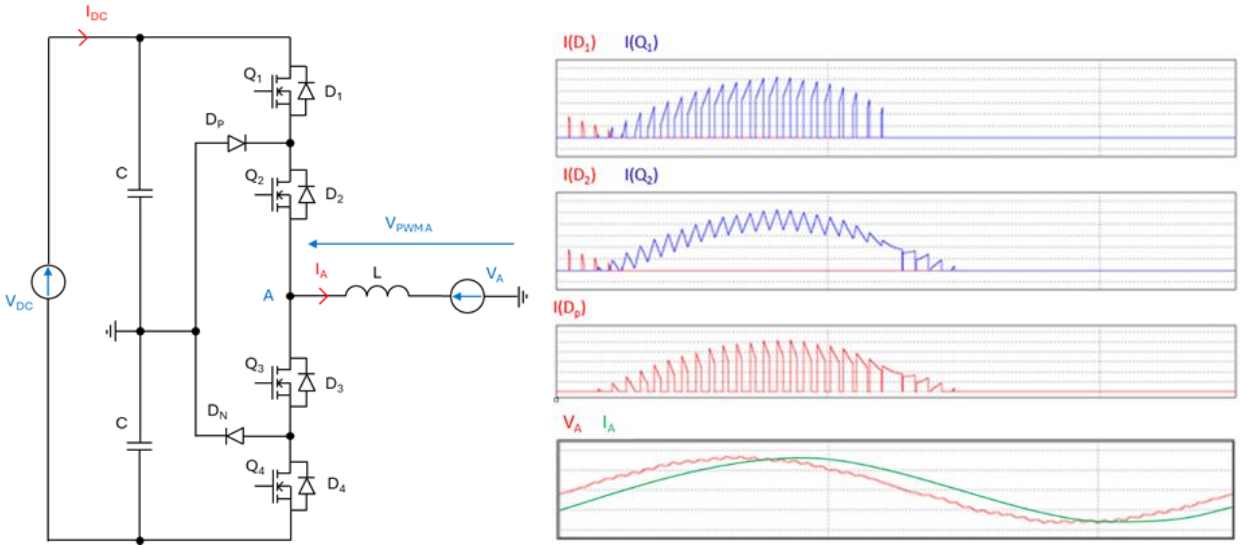


Figure 8 : Formes d'onde en courant dans les interrupteurs d'un onduleur NPC

La méthode de calcul des pertes en conduction et en commutation s'inspire des méthodes développées précédemment. Les courants dans les diodes antiparallèles étant très faibles si le facteur de puissance de la charge est proche de 1, leurs expressions ne seront pas données par soucis de simplification. En configuration NPC, l'expression du rapport cyclique est différente de celle l'onduleur 2 niveaux (18).

$$\alpha(\theta) = m.\sin(\theta) \tag{18}$$

Les équations (19) et (20) indiquent le courant efficace dans un transistor externe (Q_1 et Q_4). Les équations (21) et (22) indiquent le courant efficace dans un transistor interne (Q_2 et Q_3).

Les équations (23), (24), (25) et (26) indiquent le courant dans une diode de clamp (Dp et Dn).

$$I_{RMS\,Q_1\,NPC} = \sqrt{\frac{1}{2\pi}\int_{\varphi}^{\pi} \alpha(\theta)\, I_{commuté}(\theta)^2 d\theta} \tag{19}$$

$$I_{RMS\,Q_1 NPC} = I_{AC}\sqrt{(1+\cos(\varphi))^2\frac{m}{3\pi}} \tag{20}$$

$$I_{RMS\,Q_2\,NPC} = \sqrt{\frac{1}{2\pi}\left(\int_{\varphi}^{\pi} I_{commuté}(\theta)^2 d\theta + \int_{0}^{\varphi}\big(1-\alpha(\theta)\big)I_{commuté}(\theta)^2 d\theta\right)} \tag{21}$$

$$I_{RMS\,Q_2\,NPC} = I_{AC}\sqrt{\left(\frac{1}{2} - (1-\cos(\varphi))^2\frac{m}{3\pi}\right)} \tag{22}$$

$$I_{RMS\,D_p\,NPC} = \sqrt{\begin{matrix}\frac{1}{2\pi}\int_{\varphi}^{\pi}(1-\alpha(\theta))I_{commuté}(\theta)^2 d\theta \\ +\frac{1}{2\pi}\int_{\pi}^{\varphi+\pi}\big(1+\alpha(\theta)\big)I_{commuté}(\theta)^2 d\theta\end{matrix}} \tag{23}$$

$$I_{RMS\,D_p\,NPC} = I_{AC}\sqrt{\left(\frac{1}{2} - \frac{2m}{3\pi}(1+\cos(\varphi))^2\right)} \tag{24}$$

$$I_{moyen\,D_p\,NPC} = \begin{matrix}\frac{1}{2\pi}\int_{\varphi}^{\pi}(1-\alpha(\theta))I_{commuté}(\theta)d\theta \\ +\frac{1}{2\pi}\int_{\pi}^{\varphi+\pi}\big(1+\alpha(\theta)\big)I_{commuté}(\theta)d\theta\end{matrix} \tag{25}$$

$$I_{moyen\,D_p\,NPC} = \frac{I_{AC}\sqrt{2}}{2\pi}\left(2 + m\left((\varphi-\frac{\pi}{2})\cos(\varphi) - \sin(\varphi)\right)\right) \tag{26}$$

Les pertes par commutation sont calculées pour un transistor externe (Q_1 et Q_4) (27) (28), et pour un transistor interne (Q_2 et Q_3) (29) (30), en considérant que la tension commutée est divisée par deux selon les avantages des structures 3 niveaux.

$$P_{Q1\,com} = \frac{V_{DC}}{2.V_{ref}}\frac{1}{2\pi}\int_{\varphi}^{\pi}\big(E_{T\,on} + E_{T\,off}\big).F_{dec}\,d\theta \tag{27}$$

$$P_{Q1\,com} = \frac{V_{DC}}{2.V_{ref}}\frac{f_{dec}}{2\pi}.\begin{pmatrix} a_T\,.\,{I_{AC}}^2\left(\pi-\varphi+\frac{\sin(2\varphi)}{2}\right) \\ +b_T\,.\,I_{AC}\sqrt{2}.(1+\cos(\varphi)) + c_T(\pi-\varphi)\end{pmatrix} \tag{28}$$

$$P_{Q2\,com} = \frac{V_{DC}}{2.V_{ref}}\frac{1}{2\pi}\int_{\pi}^{\pi+\varphi}(E_{T\,on} + +E_{T\,off})F_{dec}\,d\theta \tag{29}$$

$$P_{Q2\,com} = \frac{V_{DC}}{2.V_{ref}}\frac{f_{dec}}{2\pi}.\begin{pmatrix} a_T.{I_{AC}}^2\left(\varphi-\frac{\sin(2\varphi)}{2}\right) \\ +b_T.I_{AC}\sqrt{2}.(1-\cos(\varphi)) + c_T.\varphi\end{pmatrix} \tag{30}$$

3.5 - Modèle thermique

Le fonctionnement des composants actifs de puissance, diode et transistor, génère des pertes qu'il convient d'évacuer. Classiquement, trois solutions sont envisagées : un refroidissement à eau, un refroidissement à air en convection forcée, ou un refroidissement à air en convection naturelle. La solution la plus performante est le refroidissement par eau, mais cela nécessite la mise en place

d'un système hydraulique : pompe, échangeur thermique, régulations. La Figure 9 représente les flux de chaleur partant des interrupteurs jusqu'au dissipateur, ayant une température supposée fixe. Les pertes proviennent de trois boîtiers de puissance intégrant chacun deux transistors et deux diodes dans le cas de l'onduleur 2 niveaux. Le principe reste identique dans le cas du 3 niveaux, en ajoutant des boîtiers.

Classiquement, on considère que le flux de chaleur transportant les pertes de la puce au dissipateur est principalement freiné par deux résistances thermiques : la résistance thermique de contact 'junction to case' et la résistance thermique de contact 'case to heatsink'. La résistance 'junction to case' varie en fonction de la taille de l'interrupteur, la résistance 'case to heatsink' dépend de la qualité de la fixation du module sur le dissipateur et de sa surface. L'hypothèse est faite que la résistance thermique entre le boîtier et le dissipateur est unique, et pas distribuée pour chaque composant. Le schéma équivalent retenu est finalement proposé sur la Figure 10.

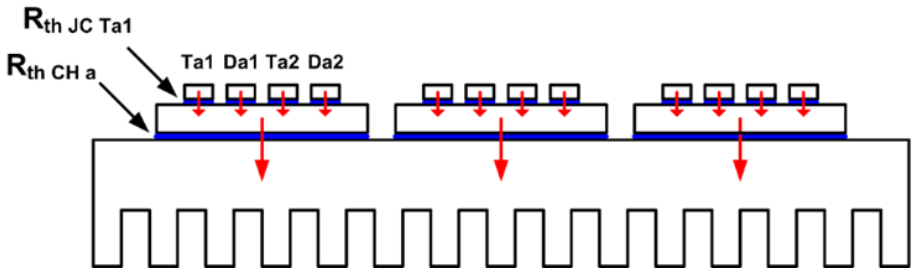


Figure 9 : Schéma thermique représentant 3 boîtiers sur un dissipateur

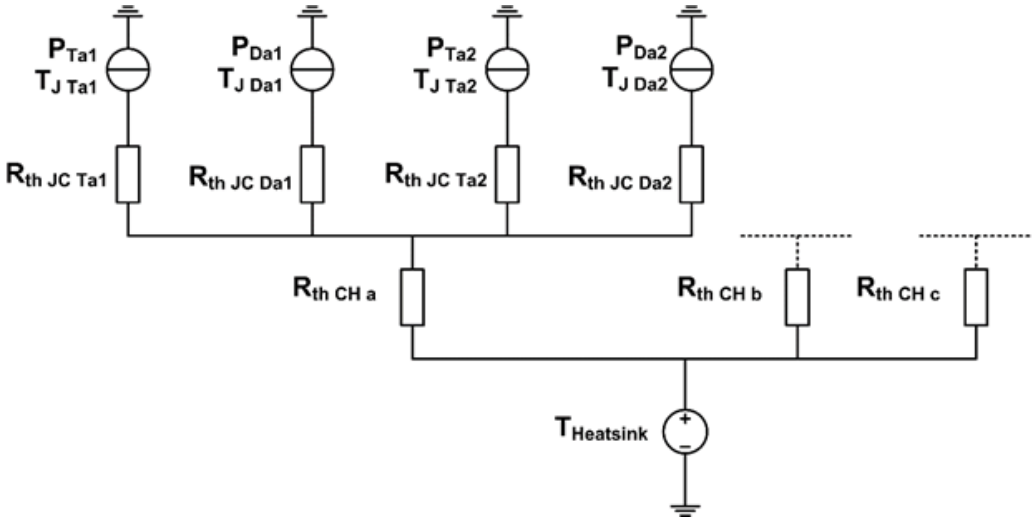


Figure 10 : Réseau de résistance équivalent au schéma thermique

Les pertes dans chaque composant sont calculées comme indiqué précédemment. La valeur des résistances thermiques de contact peut être déduite à partir des données du constructeur. Selon le schéma thermique équivalent, les températures de jonction d'un MOSFET et d'une diode sont calculées selon (31) et (32). Les températures de jonction ne doivent par exemple pas dépasser 125°C pour assurer une bonne fiabilité.

$$T_{J\,MOSFET} = T_{sink} + P_{MOSFET} R_{th\,JC\,MOSFET} + (2P_{MOSFET} + 2P_{diode}) R_{th\,CH} \quad (31)$$

$$T_{J\,diode} = T_{sink} + P_{diode} R_{th\,JC\,diode} + (2P_{MOSFET} + 2P_{diode}) R_{th\,CH} \quad (32)$$

4 - Dimensionnement des composants passifs

4.1 - Dimensionnement des inductances de lissage

Le dimensionnement des inductances de lissage en sortie d'onduleur répond au besoin de filtrage des harmoniques issues du découpage dans le cas de la connexion réseau. De nombreuses contraintes régissent le dimensionnement de ces inductances :

- Garantir une valeur d'inductance suffisante pour la contrainte de Taux de Distorsion Harmonique (THD) du réseau imposé par des normes (EN/CEI 61000-3-2). Cette contrainte n'est pas discutée ici car elle dépend de l'ordre de filtre sélectionné (L, LC, LCL, etc…). Cela nécessite le développement de modèles fréquentiels ou le couplage à une simulation circuit.
- Garantir une valeur d'inductance faible pour ne pas absorber trop de puissance réactive. Une contrainte empirique simple consiste à limiter la chute de tension aux bornes du composant à 10% du fondamental du réseau.
- Garantir une induction crête dans le noyau inférieur à l'induction à saturation du matériau magnétique. Le calcul d'induction est présenté dans les lignes qui suivent.
- Garantit la capacité de bobiner le nombre de spires requis dans la fenêtre de bobinage.
- Garantir une température de composant raisonnable pour ne pas dégrader ses propriétés. Á défaut de modèles thermiques viables quel que soit la géométrie, on prendra comme image de cette contrainte la densité de courant dans le bobinage et les pertes volumiques dans le noyau.
- Garantir des pertes, une masse, un volume ou un prix minimal en fonction du cahier des charges de l'utilisateur.

La méthode présentée ici consiste à calculer en premier lieu la variation de tension aux bornes de l'inductance, qui ne dépend pas de la valeur de cette dernière. Les expressions développées ci-après ne sont valables que dans le cas où les références de potentiel DC et AC sont communes, comme indiqué sur la figure 1. Dans ce cas de figure, à l'échelle d'une période haute fréquence, l'ondulation de tension (32) est calculée entre 0 et αT, avec V_{PWM} valant $V_{DC}/2$ sur cet intervalle, et $\omega_{grid} = 2\pi f_{grid}$. Connaissant cette grandeur, il est possible de calculer la variation d'induction grâce à la loi de Lenz-Faraday énoncée selon (33). Cette écriture suppose que l'induction est uniforme dans le noyau de section S_{mag} autour duquel sont bobinées N_{spires} spires. Ces paramètres sont représentés sur la figure 11, en considérant une inductance à noyau torique.

$$\Delta V(t) = \frac{V_{DC}}{2} - V_{AC}\sqrt{2}\sin\left(\omega_{grid}.t\right) \tag{32}$$

$$\Delta B(t) = \frac{1}{N_{spires}S_{mag}}\int_0^{\alpha T}\Delta V dt \tag{33}$$

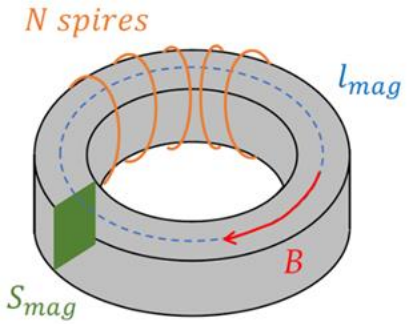


Figure 11 : Définition des paramètres de l'inductance torique

En supposant que l'ondulation de tension ΔV ne dépend pas de la variation de tension du réseau V_{AC} à l'échelle de l'intégration sur une période de découpage, et en considérant l'expression du coefficient de réglage m, l'ondulation d'induction est formulée en (34).

$$\Delta B(t) = \frac{V_{DC}(1 - m^2.(\sin(\omega_{grid}.t))^2)}{4N_{spires}S_{mag}f_{dec}} \tag{34}$$

Cette valeur d'ondulation permet le calcul des pertes volumiques fer dans le noyau. L'équation (35) utilise le modèle de Steinmetz [7], en réduisant l'ondulation à un sinus d'amplitude $\frac{\Delta B(0)}{2}$, ce qui représente la valeur maximale d'ondulation, à la fréquence de découpage F_{dec}. Les coefficients k, a et b se déduisent des abaques fournies par le fabricant du matériaux magnétique [8]. La valeur des pertes volumiques dans le noyau doit être maitrisée, car la température du noyau augmente environ de 10°C par tranche 100 mW par cm^3 de pertes fer en convection naturelle.

$$P_{vol\ noyau} = k.{f_{dec}}^{a}.\frac{\Delta B(0)^b}{2} \tag{35}$$

L'évaluation de l'induction crête (36) est nécessaire pour vérifier qu'on ne dépasse pas l'induction à saturation du noyau. On utilise ici l'hypothèse d'un matériau ayant une perméabilité relative constante μ_r, matériau en régime linéaire non saturé, et un champ parcourant un contour de longueur l_{mag}, sans entrefer.

$$B_{crête} = \frac{N_{spires}.\mu_r.\mu_0}{l_{mag}} I_{L\ AC}\sqrt{2} \tag{36}$$

Les contraintes sur le noyau ayant été énoncées, il reste à écrire celles pour les conducteurs. Une première contrainte de nature géométrique consiste à s'assurer que l'intégralité des N_{spires} spires rentre dans la fenêtre de bobinage du noyau. La surface de bobinage ne doit pas être remplie à plus de 50% pour assurer le passage d'une navette lors du bobinage des spires. Il est également possible d'imposer un bobinage monocouche pour éviter les effets capacitifs entre les spires de différentes couches, ce qui génèrerait davantage de perturbations électromagnétiques.

Enfin, l'évaluation du courant est indispensable pour choisir la section de cuivre, calculer les pertes dans le conducteur, et éviter un échauffement trop important.

L'ondulation de courant se déduit directement de l'ondulation d'induction. La formulation (37) fait apparaître la valeur d'inductance L.

$$\Delta I_{L\ AC}(t) = \frac{l_{mag}.\Delta B(t)}{N_{spires}.\mu_r.\mu_0} = \frac{V_{DC}(1 - m^2.(\sin(\omega_{grid}.t))^2)}{4.L.f_{dec}} \tag{37}$$

Les pertes cuivre résultent du toutes les composantes harmoniques du courant circulant dans l'inductance. En considérant que l'ondulation de courant est de forme triangulaire, en prenant sa valeur maximale à t = 0. Les pertes cuivre s'expriment selon (38).

$$P_{cuivre} = R(f_{grid}).{I_{L\,AC}}^2 + R(f_{dec}).(\frac{\Delta I_{L\,AC}(0)}{2\sqrt{2}})^2 \quad (38)$$

La résistance du bobinage à la fréquence fondamentale du réseau est très proche de la résistance DC, mais ce n'est pas forcément le cas à la fréquence de découpage. Dans les conducteurs massifs, les électriciens ont remarqué depuis longtemps que le courant alternatif préfère circuler à la périphérie du conducteur. Cela s'explique par la formation de courants induits par le champ magnétique généré par le courant alternatif haute fréquence. Ces courants induits s'additionnent au courant alternatif à la périphérie du conducteur, alors qu'ils s'annulent au centre : il s'agit de l'effet de peau. Boucherot définit l'épaisseur de peau δ (39) comme une constante définissant la largeur approximative de la zone où se concentre le courant dans un conducteur en fonction de la fréquence du signal.

$$\delta = \sqrt{\frac{\rho}{\pi\mu f}} \quad (39)$$

avec μ la perméabilité magnétique du conducteur, ρ la résistivité du conducteur et f la fréquence du signal.

La formule de Levasseur [9] donne une bonne approximation de la résistance d'un conducteur en fonction de la fréquence. La formule (40) donne le rapport entre la résistance DC et la résistance à la fréquence f, pour un conducteur de rayon r.

$$\frac{R_{AC}}{R_{DC}}(f) = \frac{1}{4} + \sqrt[6]{0.178 + (\frac{r}{2\delta})^6} \quad (40)$$

Pour finir, la section du conducteur doit être suffisante pour éviter un échauffement excessif. Une bonne pratique consiste à adopter une densité de courant inférieure à 5 A/mm², cette valeur pouvant être modulée par la quantité de pertes par effet de peau et par la technologie de refroidissement mise en œuvre.

Les modèles développés ici dans le cas de l'onduleur 2 niveaux, peuvent être extrapolés au cas de l'onduleur NPC en considérant l'expression du rapport cyclique (18).

4.2 - Dimensionnement des condensateurs de découplage du bus DC

Les condensateurs du bus DC permettent un filtrage efficace des harmoniques générées par les onduleurs MLI pour tenir les contraintes de qualité d'énergie du signal absorbé par le convertisseur. Le volume de ces condensateurs représente souvent une part non négligeable des onduleurs. Le dimensionnement des condensateurs dépend principalement du calibre en tension (imposé par l'application), de la valeur de capacité (nécessaire pour la fonction de filtrage), mais aussi du courant efficace traversant le composant. Très souvent, ce dernier paramètre est contraignant dans le dimensionnement du condensateur. Une approximation du courant efficace dans le condensateur est indiqué en (41) [10].

$$I_{C\,efficace} = \frac{I_{AC}\sqrt{2}}{2} \quad (41)$$

Une particularité de la structures NPC réside dans la présence d'une harmonique basse fréquence dans le spectre du courant dans les condensateurs d'entrée. Cette harmonique basse fréquence circule par les diodes de clamp jusqu'au point milieu capacitif, pour ensuite repartir dans le convertisseur, sans jamais atteindre la source [11]. Il s'agit en quelque sorte d'un courant de mode commun. L'harmonique qui ressort principalement est à trois fois la fréquence du réseau, elle peut s'approximer selon (42).

$$I_{C\ NPC\ h3} = \frac{4m}{5\pi} I_{AC} \tag{42}$$

Il existe une multitude de technologies de condensateur adaptées à la conception des convertisseurs de puissance. La tension du condensateur affichée par le constructeur doit être supérieure à celle que le composant doit tenir pour des raisons de fiabilité. Les plus communes d'entre elles sont représentées sur la Figure 12, en fonction de la valeur de capacité voulue et de la tension à supporter.

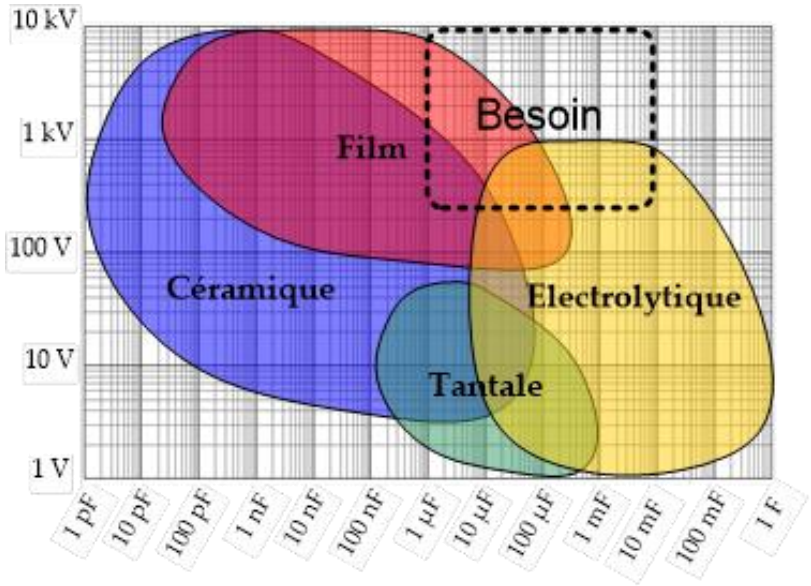


Figure 12 : Choix de la technologie de condensateur en fonction de la tension et de la capacité

En considérant les niveaux de tensions et de courant du cahier des charges, deux technologies de condensateurs se distinguent : les condensateurs film et les condensateurs électrolytiques. Les condensateurs films admettent un courant efficace élevé, mais l'énergie volumique stockable est assez faible, ce qui conduit à un gros volume de composant à iso-tension et iso-capacité par rapport à d'autres technologies. Á l'inverse, les condensateurs aluminium électrolytique ont une moins bonne tenue en courant, vieillissent plus rapidement, mais ils sont plus légers que les condensateurs films.

5 - Comparaison des structures

On souhaite ici étudier la pertinence de la montée en tension dans le domaine aéronautique. En effet, l'électrification des fonctions propulsives et non-propulsives des actionneurs embarqués dans les aéronefs représente une solution prometteuse pour diminuer l'impact environnemental de cette filière [12]. Dans le contexte de l'augmentation des puissances embarquées, la montée en tension et le recours à des architectures multiniveaux est une solution pour diminuer les pertes des systèmes électriques. Le tableau 1 reprend donc un cahier des charges avec des tensions aéronautiques actuellement utilisées dans le Boeing 787 (540 Vdc et 115 Vac), et envisage l'augmentation de ces valeurs pour répondre au besoin d'un actionneur de forte puissance (90 kW).

Cahier des charges			
Description	**Variable**	**Unité**	**Valeur**
Puissance active de la charge	P	W	90 000
Facteur de puissance de la charge	PF	-	0.9
Fréquence réseau	f_{grid}	Hz	400
Fréquence de découpage	f_{dec}	Hz	Variable
Tension du bus DC	V_{DC}	V	540 ou 900
Tension AC efficace phase-neutre	V_{AC}	V	115 ou 230

Tableau 1 : Cahier des charges utilisé pour la comparaison

Les modèles développés plus haut sont intégrés dans un outil de dimensionnement permettant une comparaison des performances. La méthodologie de conception n'est pas développée ici. Un algorithme d'optimisation [13] vient proposer différentes solutions pour les deux cahiers des charges, en présentant les résultats sous forme de front de Pareto avec deux variables : la masse du convertisseur et son rendement. Des modèles de masse de composant sont donc intégrés dans l'outil, avec entre autres la masse d'échangeur thermique nécessaire au refroidissement du convertisseur estimée à 1.5 kg par kW de pertes.

Le cas de référence consiste à utiliser un onduleur 2 niveaux avec les tensions actuelles. Le second cas se résume à élever les tensions mises en jeu. Le troisième cas consiste à remplacer la structure 2 niveaux par une structure NPC dans le cas des tensions plus élevées. L'analyse des signaux commutés amène à changer le calibre en tension et en courant des interrupteurs dans chaque cas :

- Onduleur 2 niveaux avec V_{dc} = 540 V → calibre de 1200 V
- Onduleur 2 niveaux avec V_{dc} = 900 V → calibre de 1700 V
- Onduleur NPC avec V_{dc} = 900 V → calibre de 900 V

La gamme de MOSFET en carbure de silicium (SiC) du fabricant Microchip est utilisée. Le choix du calibre en courant des interrupteurs, la fréquence de découpage ainsi que le dimensionnement des composants passifs sont optimisés par l'algorithme.

La comparaison de trois cas d'étude est disponible sur la figure 13. La répartition des pertes et de la masse pour le point de masse minimale de chaque front de Pareto apparaît sur la figure 14.

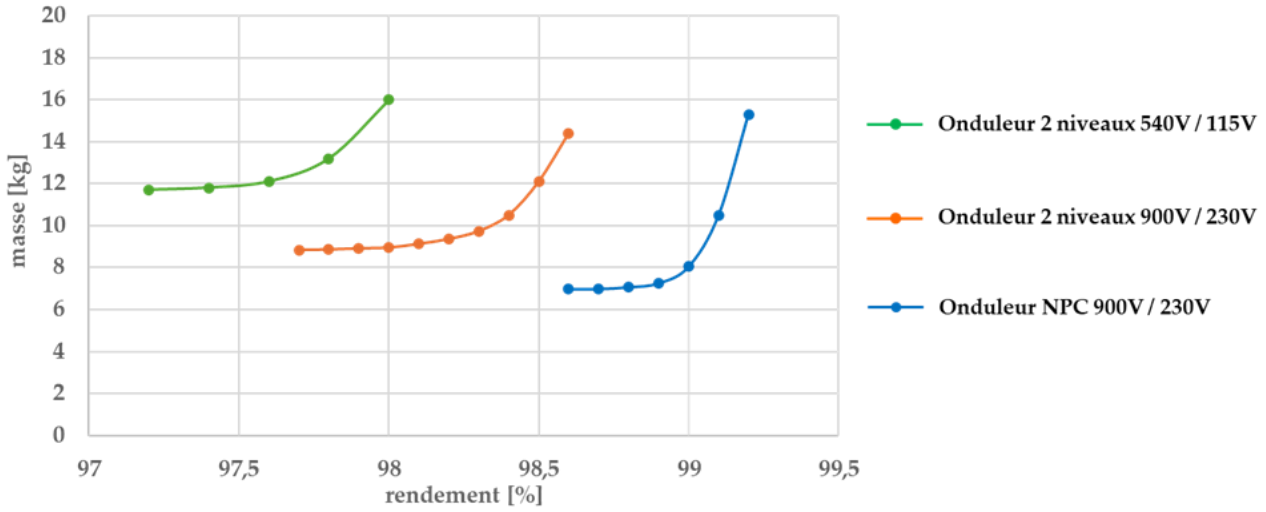


Figure 13 : Fronts de Pareto comparant les masses et rendements optimaux pour les deux onduleurs 2 niveaux et l'onduleur NPC

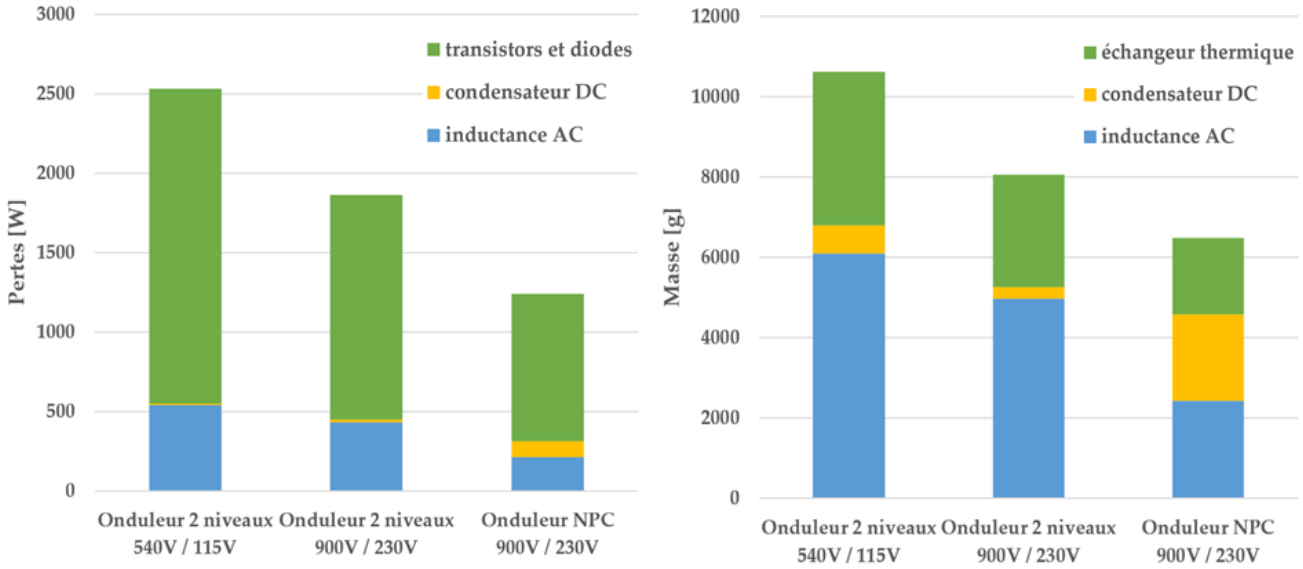


Figure 14 : Répartition des pertes et de la masse pour les différentes solutions considérées

La première conclusion de l'étude réside dans l'avantage de la montée en tension pour des fortes puissances converties. Cela permet de réduire les pertes par conduction, proportionnelles au carré du courant, dans les interrupteurs et les inductances. La masse globale diminue de 25% et le rendement augmente d'un point.

De plus, il apparaît que le passage à une structure NPC soulage à la fois les contraintes sur les interrupteurs et sur les inductances. En effet, la diminution du contenu harmonique introduite par la structure NPC permet de diminuer la taille du filtre AC mais aussi de diminuer la fréquence de découpage du convertisseur, ce qui amène à une baisse des pertes par commutation dans les interrupteurs. En revanche, le dimensionnement des condensateurs DC devient un point clé du problème, car il doivent gérer la circulation de l'harmonique trois en plus des harmoniques haute fréquence.

La présentation des résultats sous forme de front de Pareto permet de proposer plusieurs arbitrages entre deux indicateurs clés de la conception. C'est ensuite au client ou au décideur de choisir les performances qu'il souhaite, tout en ayant conscience des éventuelles particularités technologiques connues du concepteur, et n'apparaissant pas dans l'étude comparative.

6 - Conclusion

Cet article résume des considérations essentielles pour la conception des onduleurs de tension à 2 niveaux et NPC. D'une part, des formules analytiques simples sont présentées pour évaluer la pertinence du choix de calibre en courant des interrupteurs pour respecter les contraintes thermique et les contraintes de rendement. D'autre part, des modèles relatifs au dimensionnement des inductances côté AC et des condensateurs côté DC permettent de tenir les contraintes de qualité d'énergie avec des composants de taille raisonnable. Le choix de la fréquence de découpage résulte d'un compromis entre qualité d'énergie et pertes par commutation. L'intérêt de la structure NPC est ici mise en lumière sur une problématique industrielle actuelle. Le passage d'une structure à 2 niveaux à NPC permet des gains sensibles sur la masse et les pertes du convertisseur. L'utilisation des modèles analytiques permet de chiffrer ce gain en fonction du cas d'application. Les gains de la structure NPC ne sont pas systématiques : cette démarche de modélisation doit être reprise pour chaque cahier des charges pour en tirer des conclusions appropriées.

7 - Ouvrages pour approfondir

- **Electronique de puissance, de la cellule de commutation aux applications industrielles**. Cours et exercices, A. Cunière, G. Feld, M. Lavabre, éditions Casteilla, 544 p. 2012.
- **Mise en œuvre des composants électroniques de puissance**, traité EGEM (électronique-génie électrique-microsystèmes) sous la direction de Robert Perret, 2003, 326 p., Éditions Hermès Lavoisier.
- **Electronique de puissance, méthodologie et convertisseurs élémentaires**, Philippe Barrade, Presses Polytechniques et Universitaires Romandes.
- B.K. Bose, **Power electronics and variable frequency drives**, IEEE press, New York, 1997, 640p. TK 7881 P68 1997.
- R. Bausière, G. Séguier, F. Labrique, **Les convertisseurs de l'électronique de puissance volume 4 : La conversion continu-alternatif**, Tech. & Doc. Lavoisier, Paris, 1998, 561 p.
- Encyclopédie « Les techniques de l'ingénieur », Conversion de l'énergie électrique, rubrique convertisseurs électriques et application, articles D3060 à D3178.

Références :

[1] T. Meynard et G. Gateau, « Convertisseur Multiniveaux », *Rev. 3EI*, n° 72, p. 4-10, avr. 2013.

[2] J. Rodriguez, Jih-Sheng Lai, et Fang Zheng Peng, « Multilevel inverters: a survey of topologies, controls, and applications », *IEEE Trans. Ind. Electron.*, vol. 49, n° 4, p. 724-738, août 2002, doi: 10.1109/TIE.2002.801052.

[3] A. Leredde, « Study and Design of Multilevel Converters for High Power Applications », PhD, Institut National Polytechnique de Toulouse - INPT, 2011. Consulté le : 27 février 2020. [En ligne]. Disponible sur : https://tel.archives-ouvertes.fr/tel-00668376

[4] C. Rizet, « Amélioration du rendement des alimentations sans interruption », PhD, Université de Grenoble, 2011. Consulté le: 7 janvier 2020. [En ligne]. Disponible sur : https://tel.archives-ouvertes.fr/tel-00651973

[5] J.-L. Schanen, « Electronique de puissance : au coeur de la commutation », Habilitation à Diriger des Recherches, Institut National Polytechnique de Grenoble - INPG, 2000. [En ligne]. Disponible sur: https://theses.hal.science/tel-00689917

[6] Infineon, « Double Pulse Testing: The How, What and Why », 2020.

[7] C. P. Steinmetz, « On the Law of Hysteresis », *Trans. Am. Inst. Electr. Eng.*, vol. IX, n° 1, p. 1-64, janv. 1892, doi: 10.1109/T-AIEE.1892.5570437.

[8] Magnetics, « Magnetics Powder Core Catalog ». [En ligne]. Disponible sur : https://www.mag-inc.com/Media/Magnetics/File-Library/Product%20Literature/Powder%20Core%20Literature/2017-Magnetics-Powder-Core-Catalog.pdf

[9] A. Levasseur, « Nouvelles formules, valables à toutes les fréquences, pour le calcul rapide de l'effet Kelvin », *J Phys Radium*, vol. 1, n° 3, p. 93-98, 1930, doi: 10.1051/jphysrad:019300010309300.

[10] J. W. Kolar et S. D. Round, « Analytical calculation of the RMS current stress on the DC-link capacitor of voltage-PWM converter systems », *IEE Proc. - Electr. Power Appl.*, vol. 153, n° 4, p. 535-543, juill. 2006, doi: 10.1049/ip-epa:20050458.

[11] M. Marzouk, « Development of integrated chargers for plug-in hybrid vehicles », PhD, Université Grenoble Alpes, 2015. Consulté le : 5 mars 2020. [En ligne]. Disponible sur : https://tel.archives-ouvertes.fr/tel-01239349

[12] European Commission, *ACARE Flightpath 2050 - Europe's Vision for Aviation*. in Policy / European Commission. Luxembourg: Publ. Off. of the Europ. Union, 2011.

[13] A. Voldoire, « Outil de développement et d'optimisation dédié aux onduleurs SiC de forte puissance », phdthesis, Université Grenoble Alpes [2020-....], 2020. doi: 10/document.

Apprentissage par renforcement et transfert simulation vers réalité pour la conduite de voitures autonomes

La Revue 3E.I
Culture Sciences de l'Ingénieur

Édité le 23/05/2024

école normale supérieure paris-saclay

Rania BENNANI[1] – Kévin HOARAU[1] – Anthony JUTON[2]

[1] *Elève de 3ème année au département Nikola Tesla de l'École Normale Supérieure Paris-Saclay*
[2] *Professeur agrégé de physique appliquée au département Nikola Tesla de l'École Normale Supérieure Paris-Saclay*

Cette ressource fait partie du N°112 de La Revue 3EI du 2ème trimestre 2024 et s'intègre au « Dossier Intelligence Artificielle » [2] de Culture Sciences de l'Ingénieur.

Cette ressource présente l'apprentissage par renforcement de la conduite sur circuit d'une voiture autonome 1/10ème, en simulation, puis le transfert du réseau de neurones du simulateur dans la voiture réelle, en utilisant Webots, gymnasium et Stable-Baselines3. Elle est issue du travail de Kévin Hoarau lors de sa participation à la Course de Voitures Autonomes de Paris Saclay, CoVAPSy 2023 [6], repris et validé par plusieurs équipes en 2024.

Figure 1 : Voiture en apprentissage sur le simulateur Webots

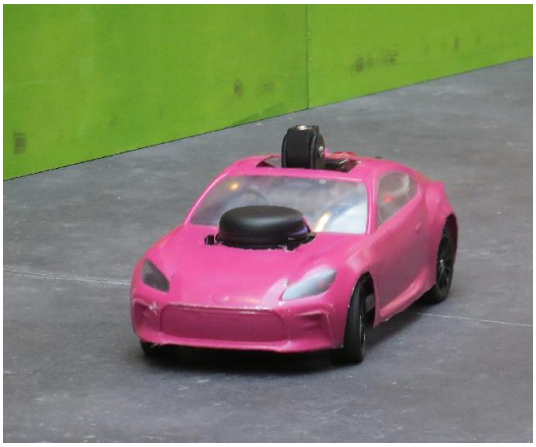


Figure 2 : Voiture réelle, avec le réseau de neurones issu de la simulation

Le simulateur utilisé, Webots, peu gourmand en ressource et open source, permet à chacun d'expérimenter l'apprentissage par renforcement profond sur cet exemple réaliste, même sans disposer de voiture pour le passage à la réalité.

La voiture 1/10ème instrumentée d'un coût modeste (moins de 1000 euros) permet à travers cet exercice d'appréhender le transfert simulation → réalité et les difficultés associées pour une mise en œuvre concrète et matérielle de l'intelligence artificielle. La voiture et le simulateur sont présentés en détails dans les ressources « *Course Voitures Autonomes Paris Saclay (CoVAPSy) : Travaux pratiques autour des voitures autonomes* » [7], « *CoVaPSy : Premiers programmes python sur la voiture réelle* » [8] et « *CoVaPSy : Mise en œuvre du Simulateur Webots* » [9] du numéro 111 de la revue 3EI.

La ressource « *Apprentissage par renforcement de la conduite d'un véhicule sur AirSim* » de Ludovic de Mattteis et Saša Radosalvjevic [3] a servi de point de départ à ce travail. Webots [11] a

été préféré à AirSim pour sa légèreté et sa facilité de mise en œuvre et l'expérience acquise précédemment a permis d'aller jusqu'au transfert de la simulation à la réalité.

1 - Introduction

L'apprentissage par renforcement (*Reinforcement Learning* en anglais) est une catégorie de Machine Learning. L'article « *Introduction à l'apprentissage par renforcement* » [1] *du* « *Dossier Intelligence Artificielle* » [2] présente cette méthode en détail. Cette ressource vient en complément, en proposant un exemple avancé de mise en œuvre de l'apprentissage par renforcement pour la conduite de voitures autonomes réelles sur un circuit.

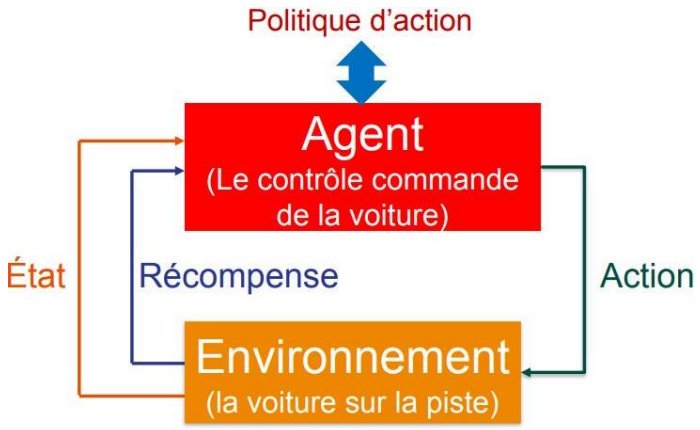


Figure 3 : Schéma explicatif de l'apprentissage par renforcement

En quelques mots, on se place dans un ensemble {**agent**, **environnement**} où une **action** choisie et réalisée par l'agent, en fonction de l'**état** de l'environnement, peut entraîner une modification de l'environnement.

Le processus d'apprentissage vise à doter l'agent d'une politique d'action lui permettant de faire les meilleurs choix. Une **récompense** est alors attribuée à l'agent dont la valeur dépend de si l'action est positive ou négative pour l'agent. Lors de chaque étape de l'apprentissage, l'agent reçoit une observation de l'environnement dans lequel il évolue. Suivant cette observation, l'agent prend une décision d'action. La décision est prise dans un ensemble d'actions appelé espace des actions. Cet espace peut dépendre de l'état.

Un exemple simple est celui d'un jeu d'échec dans lequel l'observation correspond à la position de chacune des pièces de l'échiquier et l'espace des actions est l'ensemble des déplacements possibles des pièces (un fou ne peut pas être déplacé au lancement de partie par exemple). Naturellement, on souhaite que l'agent réalise la meilleure action possible suivant l'observation reçue. L'agent, pour atteindre ce but, applique une politique d'action (notée par la suite π) qu'il utilise pour sa prise de décision. A chaque récompense obtenue, cette politique est mise à jour. Au fil des épisodes d'apprentissage, on espère ainsi atteindre une politique optimale menant à la victoire, quel que soit l'adversaire.

Dans cet article, la voiture doit, à partir de l'observation de l'environnement par son capteur Lidar, agir sur la propulsion et la direction pour parcourir le plus vite possible la piste. L'environnement est constitué de la voiture, de la piste et des voitures adverses et l'agent est le programme de pilotage de la voiture.

Des bibliothèques existent pour l'apprentissage par renforcement profond. Ici est utilisée la bibliothèque Stable-Baselines3 de PyTorch [13], référence dans le domaine de l'IA. Après plusieurs essais, c'est l'algorithme PPO (Proximal Policy Optimization) qui a donné les meilleurs résultats et

est donc retenu pour cet article, associé à Gymnasium [12], ensemble d'outils développés par OpenAI pour l'apprentissage par renforcement, repris depuis par Farama foundation.

2 - Présentation des 4 étapes menant à une conduite autonome

Le travail se présente en quatre étapes, les deux premières s'intéressant à la mise en place de la voiture réelle et de son modèle simulé, la troisième à l'apprentissage automatique sur le simulateur et la dernière au transfert du réseau du simulateur vers la voiture réelle.

2.1 – Étape 1 : fonctions de base sur la voiture réelle

La première étape consiste à équiper la voiture des capteurs et actionneurs nécessaire à la conduite autonome et à développer les fonctions Lidar, direction, propulsion pour la voiture réelle, avec un algorithme de conduite basique. Cette étape est expliquée dans la ressource « *CoVaPSy : Premiers programmes python sur la voiture réelle* » [8].

Les fonctions sont fournies sur le dépôt git [5].

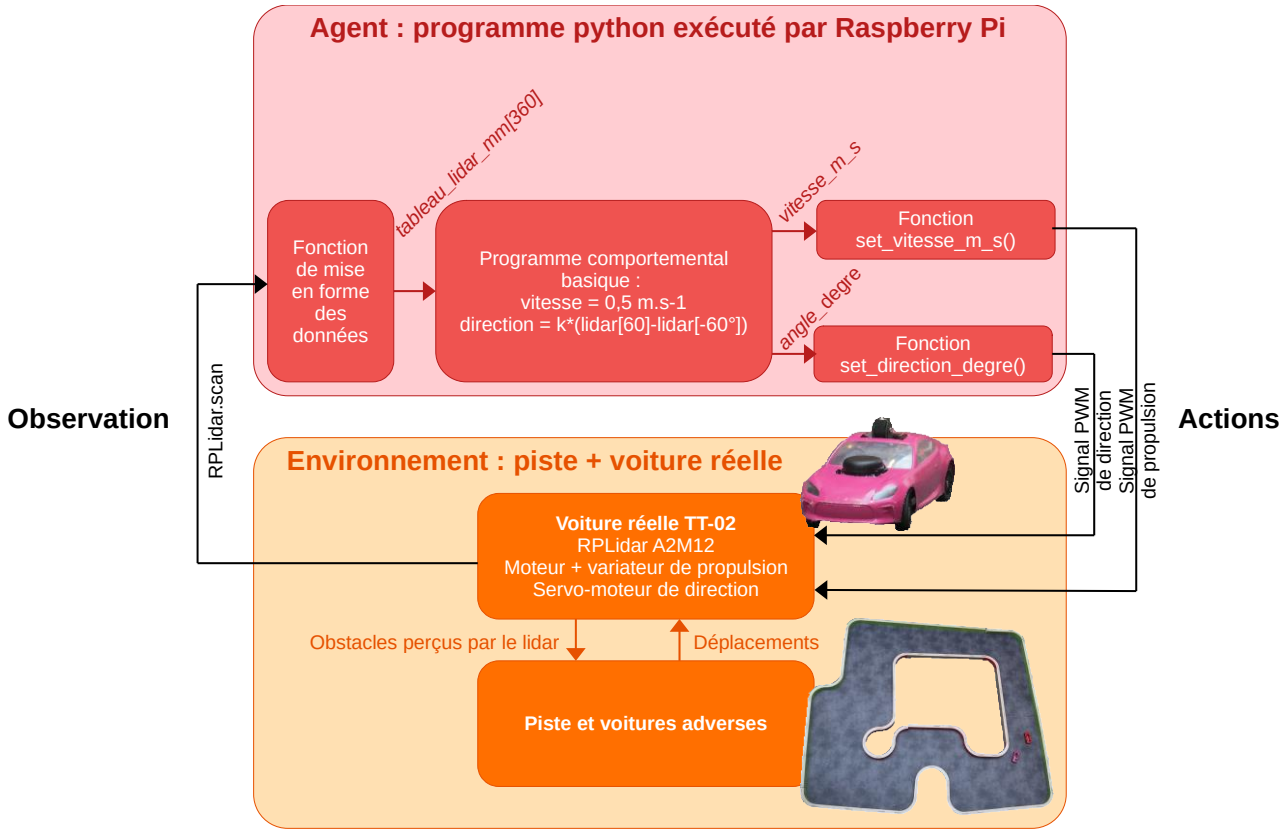


Figure 4 : Fonctions de base de la voiture réelle

2.2 – Étape 2 : fonctions de base sur le simulateur

La deuxième étape consiste à concevoir un modèle simulé de la voiture le plus proche de la voiture réelle et à développer les fonctions pour cette voiture simulée, se comportant comme les précédentes de sorte de fonctionner avec le même algorithme de conduite.

En plus de la présentation du simulateur Webots, cette étape est expliquée dans la ressource « *CoVaPSy : Mise en œuvre du Simulateur Webots* » [9].

Les fonctions et le modèle de voiture TT-02 simulée sont fournies avec le projet de base du simulateur, sur le dépôt git [5].

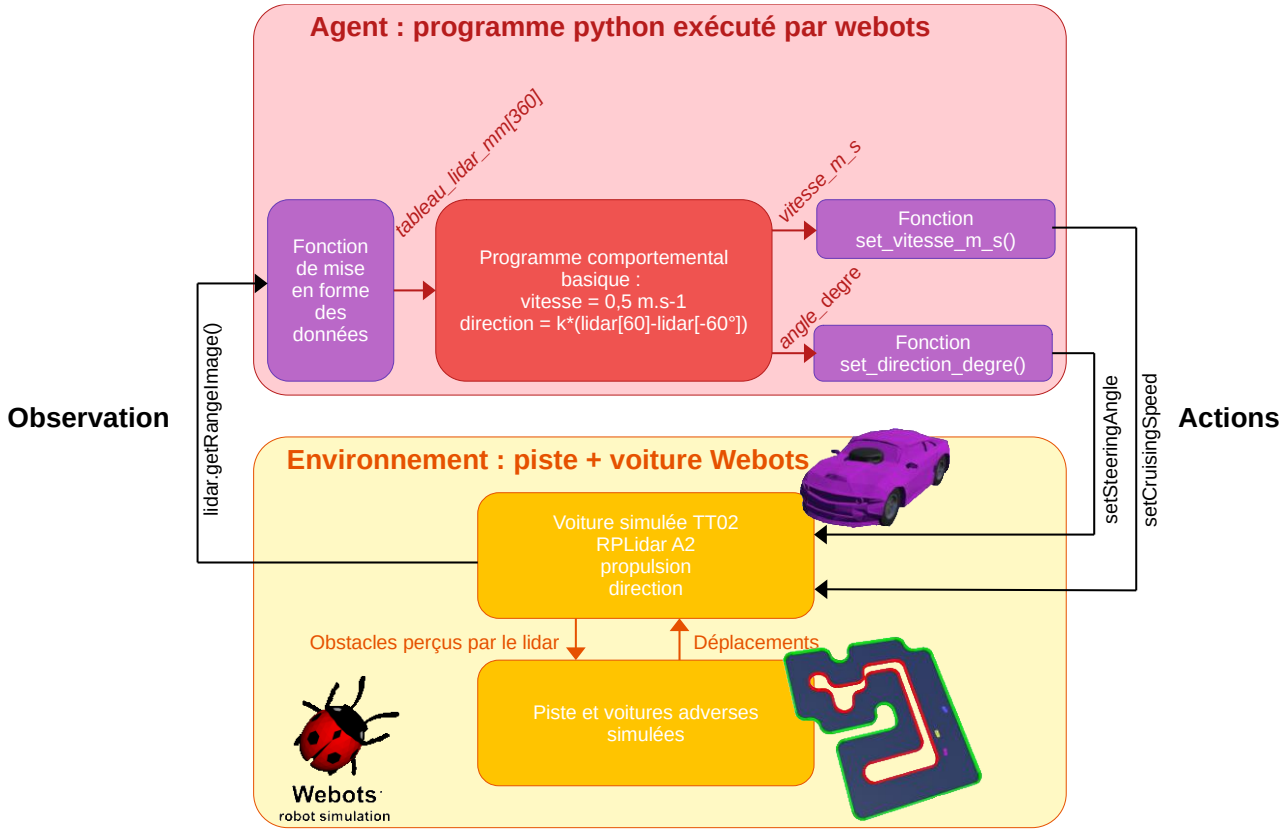


Figure 5 : Fonctions de base du simulateur

2.3 – Étape 3 : Apprentissage automatique de la conduite sur simulateur

La troisième étape remplace l'algorithme basique de conduite par un réseau de neurones et met en place l'apprentissage par renforcement de ce réseau de neurones pour aboutir à une conduite performante. C'est l'objet de la partie 3 de cette ressource.

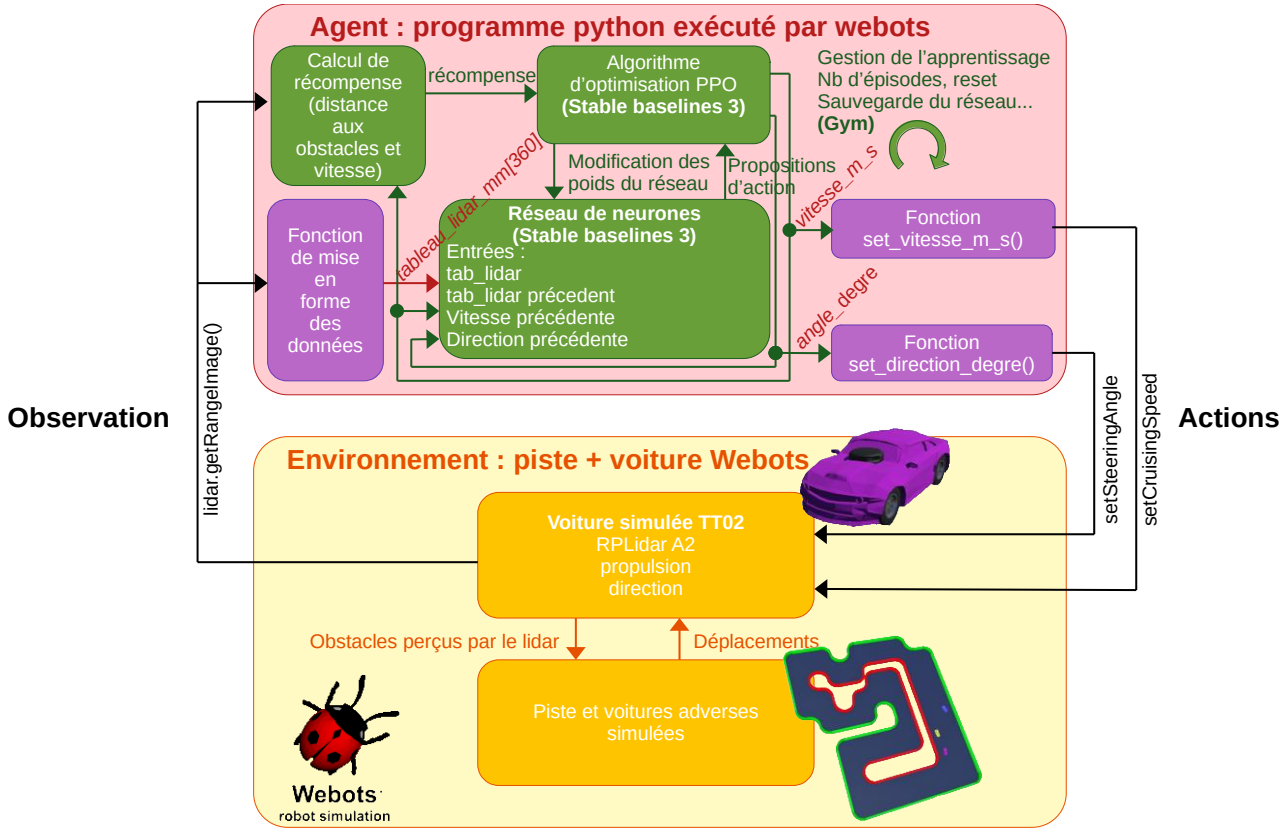


Figure 6 : Fonctions utilisées pour l'apprentissage par renforcement sur le simulateur

2.4 - Étape 4 : transfert du réseau de neurones dans la voiture réelle

Enfin, la dernière étape consiste à transférer le réseau de la voiture simulée dans la voiture réelle pour permettre à la voiture de concourir lors de la course de voitures autonomes. C'est l'objet de la partie 4 de cet article.

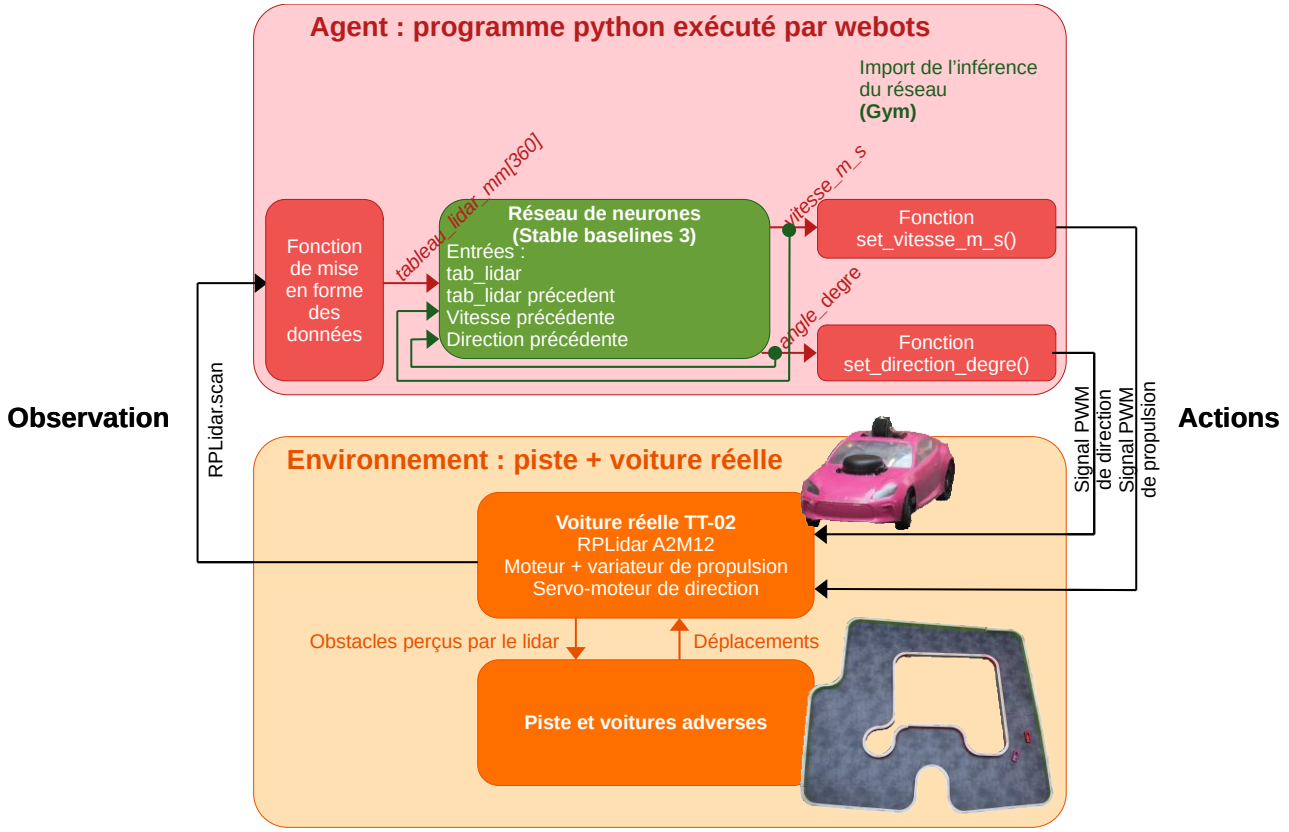


Figure 7 : Utilisation de l'inférence du réseau de neurones issu de l'apprentissage pour la conduite de la voiture réelle

3 - Apprentissage par renforcement sur simulateur

L'utilisation d'un simulateur est indispensable pour réaliser les milliers d'essais d'un apprentissage par renforcement. Pour que le transfert du simulateur à la réalité soit possible, une première exigence est d'avoir un modèle simulé proche de la voiture réelle Tamiya TT-02. La première étape du travail a donc consisté à développer ce modèle et à ajuster ses paramètres à partir des voitures des bibliothèques Webots et de la documentation de la voiture réelle. L'ensemble est décrit dans la ressource « *CoVaPSy : Mise en œuvre du Simulateur Webots* » [9] et le projet de base est disponible en téléchargement sur le dépôt github [5].

Cet article utilise le projet *Simulateur_CoVAPSy_Webots2023b_RL.zip* (en annexe de ce présent article [15]) issu de celui de l'article cité ci-dessus [9], avec en plus le superviseur et des voitures *sparring partners* [5]. Le simulateur utilisé est Webots, de Cyberbotics, dans sa version R2023b. C'est un logiciel open source permettant de créer un « monde » (*world* dans Webots) dans le but de simuler des machines robotiques. Le programme pilotant la voiture est écrit en Python. Il est également possible de programmer en Java, C++ ou Matlab.

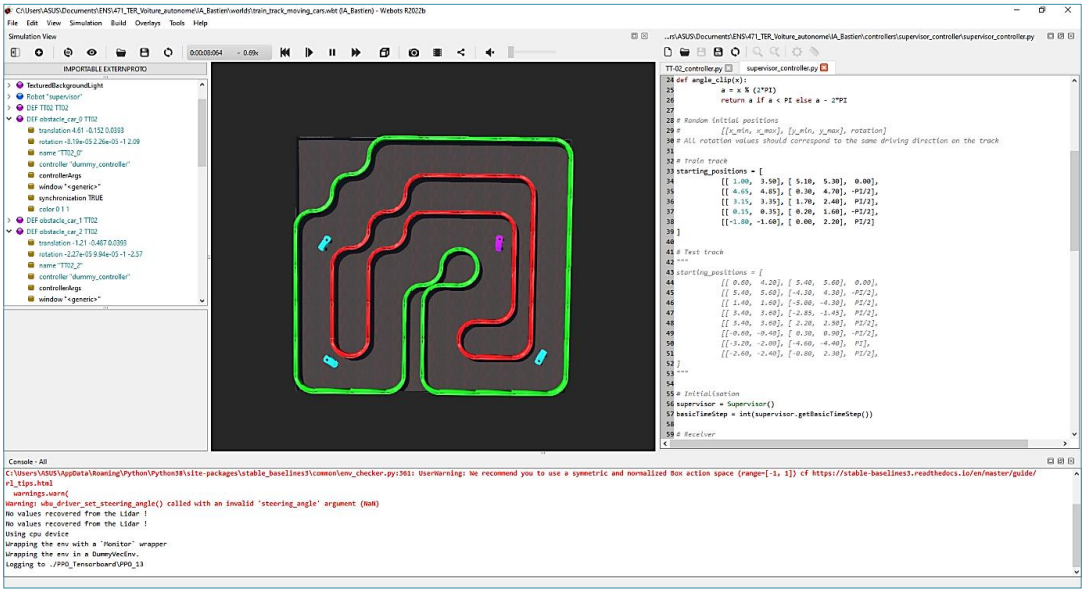


Figure 8 : Interface du simulateur Webots

Le code étant assez complexe, il est recommandé, au moins pour l'agent (le *controller* de la voiture) d'utiliser un logiciel de développement python plus complet que l'éditeur de Webots, comme expliqué dans l'article « CoVaPSy : Mise en œuvre du Simulateur Webots » [9].

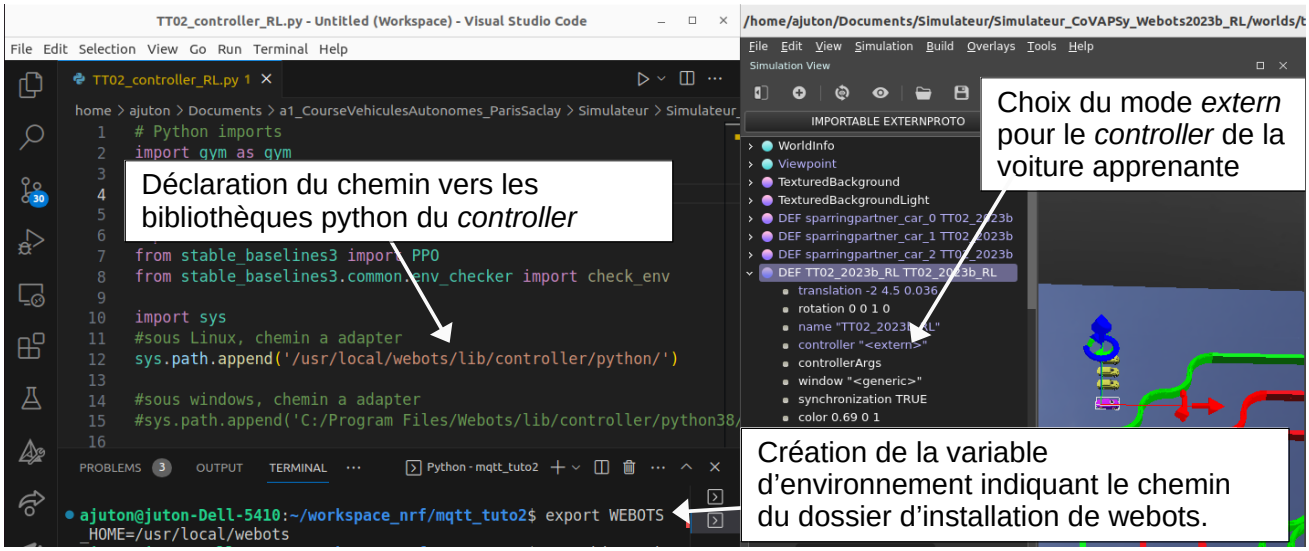


Figure 9 : Configuration de Webots et du programme python pour utiliser un logiciel de développement extérieur à Webots

En plus du modèle de la voiture, pour pouvoir entraîner efficacement celle-ci sur le simulateur, il est nécessaire de disposer de plusieurs éléments qui seront détaillés par la suite :

- Les pistes, à la fois pour l'apprentissage et l'évaluation ;
- Un superviseur capable de replacer les voitures pour les réinitialisations ;
- La bibliothèque Stable-Baselines3 pour la génération du réseau de neurones et son apprentissage avec l'algorithme PPO ;
- La bibliothèque Gymnasium pour gérer le processus d'apprentissage, le lien avec l'environnement Webots, et l'utilisation de l'inférence.

3.1 - Pistes utilisées

Pour un apprentissage par renforcement, il est nécessaire de disposer d'un maximum de situations différentes, représentatives des situations dans lesquelles se trouvera la voiture réelle. Dans le cas

présent, il faut une piste avec de longues lignes droites, des virages dans chaque sens, des virages à 180°, etc. On propose donc la piste suivante comme étant la piste d'entraînement :

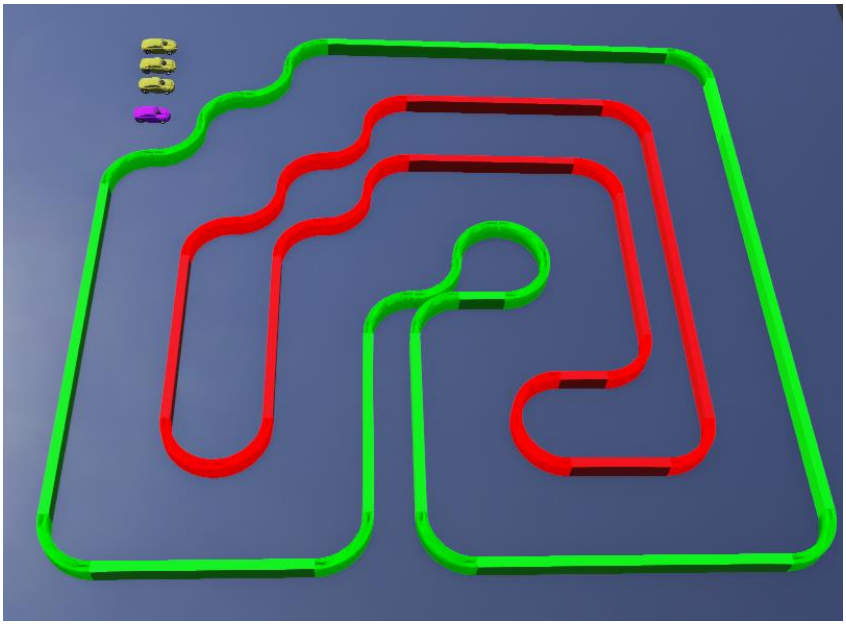


Figure 10 : Piste d'entraînement pour l'apprentissage par renforcement

Sur cette piste, on ajoute trois voitures jaunes ayant pour modèle la TT-02 et ayant un *controller* (programme de conduite) très simple (la vitesse est constante et l'angle de la direction est proportionnel à la distance mesurée à 60° moins celle mesurée à -60°). En démarrant de manière aléatoire sur la piste, on obtient ainsi une piste représentative de la course.

Dans un processus d'apprentissage, il est nécessaire de valider ce qui a été appris sur la piste d'entraînement sur une autre piste, pour vérifier notamment qu'il n'y a pas eu de sur-apprentissage (la voiture apprend à aller très vite sur la piste d'essai mais est incapable de rouler sur une autre piste). Pour cela, est créée une piste de validation avec cinq voitures en plus.

Remarque : la piste de test fournie mériterait d'être un peu élargie car elle présente des zones plus étroites que sur la piste d'entraînement et sur les pistes réelles.

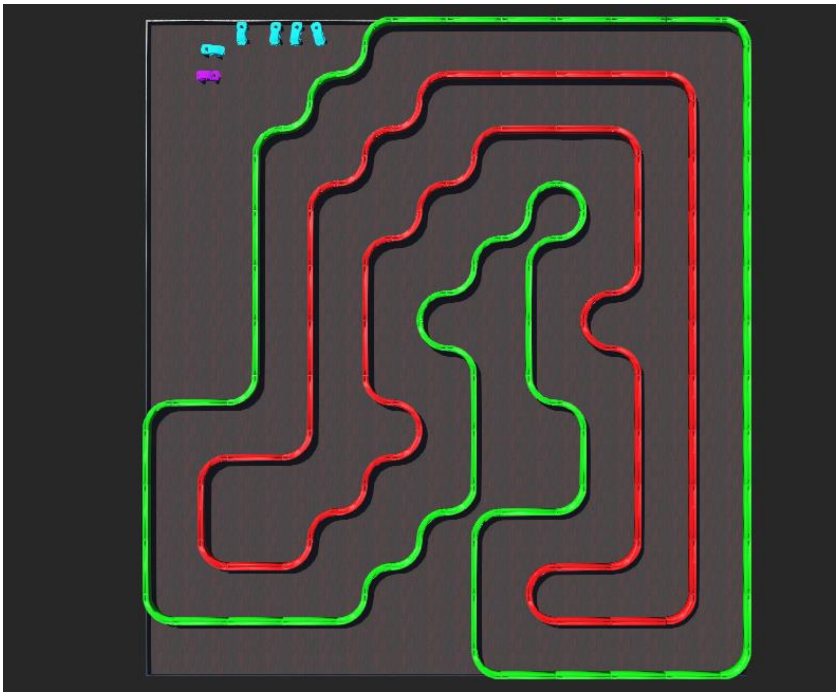


Figure 11 : Piste de validation pour l'apprentissage par renforcement

3.2 - Le superviseur

Sur ces pistes, pour pouvoir lancer l'apprentissage, il est nécessaire de gérer l'ensemble des voitures lors d'une phase de réinitialisation. Pour cela est ajouté un robot « **superviseur** » qui permet de repositionner toutes les voitures.

Le superviseur reçoit, via un récepteur, un message de l'agent de la voiture en apprentissage lors d'une collision pour remettre l'ensemble des voitures en place de manière aléatoire pour un nouvel épisode. Il possède pour cela un tableau d'encadrements de positions dans lesquelles est choisie aléatoirement une valeur pour affecter « intelligemment » une position aléatoire à chaque véhicule. Cela évite que deux voitures ne se touchent au départ, que des voitures ne suivent la piste en sens inverse ou qu'une voiture ne soit face au mur. Le choix de valeurs de positions et de sens de rotation aléatoires évite le sur-apprentissage (la voiture apprend un algorithme efficace uniquement pour un départ dans la position initiale de l'apprentissage).

Une fois la mise en place effectuée, le superviseur, via un émetteur, envoie un message à l'agent indiquant que les voitures sont prêtes pour redémarrer. Les voitures *sparring partners* sont repositionnées par le superviseur mais ne communiquent pas. La voiture apprenant la conduite est un nœud TT02_2023b_RL différent du TT02_2023b par la présence d'un émetteur et d'un récepteur.

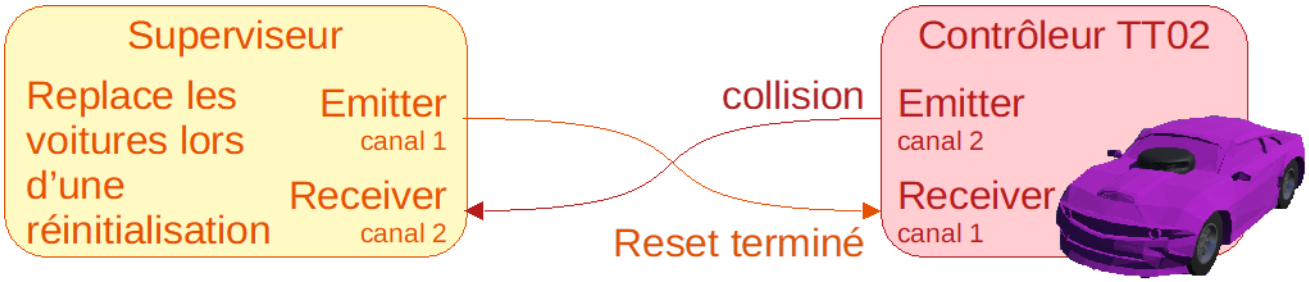


Figure 12 : Messages échangés par les émetteurs/récepteurs du superviseur et de la voiture en apprentissage

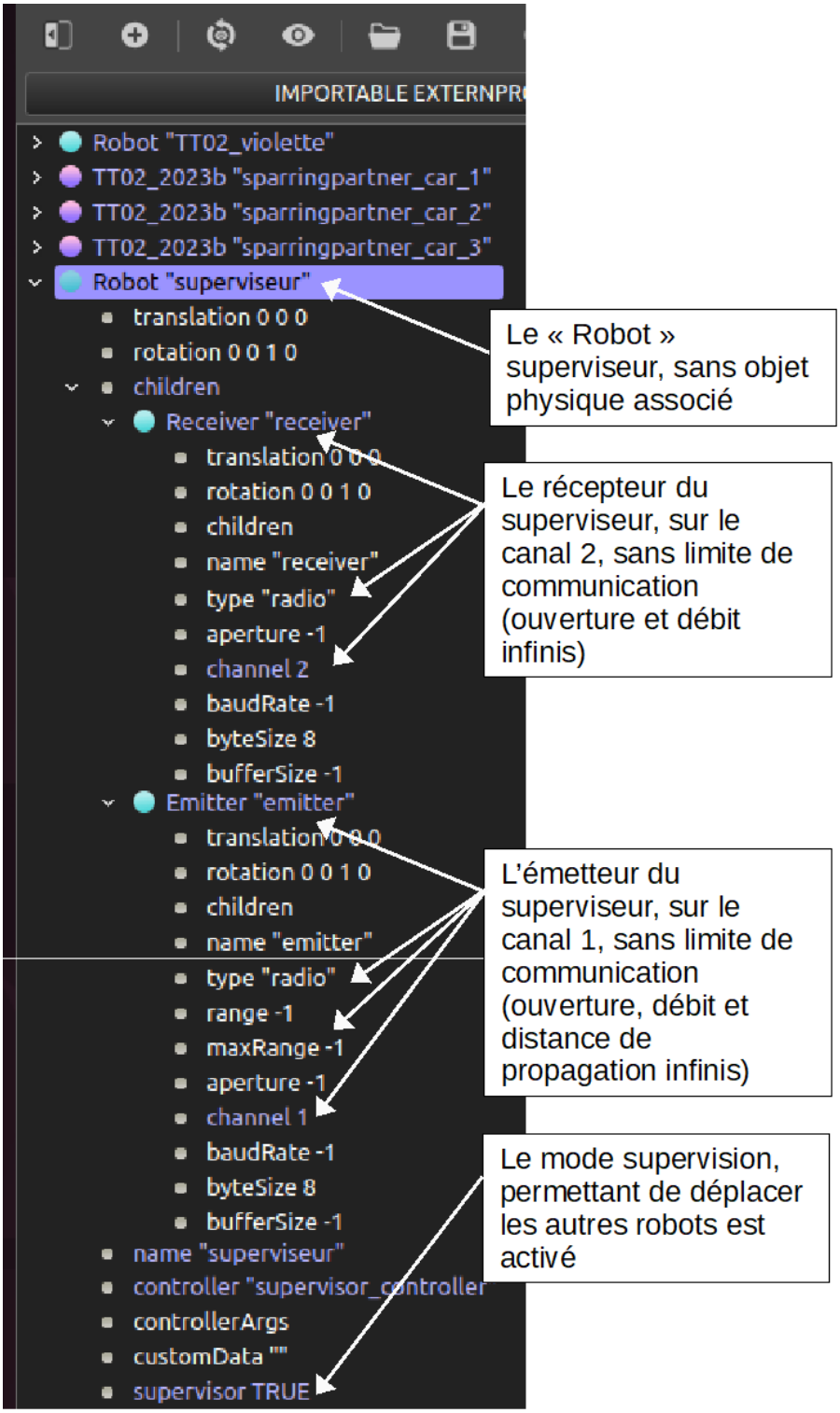


Figure 13 : Le superviseur dans l'arborescence des éléments du projet

On détaille ci-dessous le code du superviseur :

```python
# Python imports
import random

# Webots imports
from controller import Supervisor

# Global constants
PI = 3.141592653589793
RECEIVER_SAMPLING_PERIOD = 64 # In milliseconds

# /!\ Set the number of sparring partner cars
NB_SPARRINGPARTNER_CARS = 3

# Clipping function
def value_clip(x, low, up):
        return low if x < low else up if x > up else x

# Angle normalization function (Webots angle values range between -pi and pi)
def angle_clip(x):
        a = x % (2*PI)
        return a if a < PI else a - 2*PI

# Positions intiales aléatoires
# (plusieurs positions initiales auxquelles s'ajoute un peu d'aléatoire) [[x_min, x_max], [y_min, y_max], rotation]
# Les angles correspondents au même sens de rotation pour les voitures

# Train track
starting_positions = [
        [[ 0.00,  4.00], [ 5.10,  5.30],  0.00],
        [[ 4.95,  5.15], [ 0.30,  4.00], -PI/2],
        [[ 2.70,  4.50], [-1.25, -0.75],  PI  ],
        [[ 3.20,  3.45], [ 2.00,  2.40],  PI/2],
        [[ 1.90,  2.70], [ 3.10,  3.30],  PI  ],
        [[ 0.00,  0.25], [-0.50,  1.50], -PI/2],
        [[-2.20, -1.90], [-0.50,  2.30],  PI/2]
]

"""
Test track
starting_positions = [
        [[ 0.60,  4.20], [ 5.40,  5.60],  0.00],
        [[ 5.40,  5.60], [-4.30,  4.30], -PI/2],
        [[ 1.40,  1.60], [-5.00, -4.30],  PI/2],
        [[ 3.40,  3.60], [-2.85, -1.45],  PI/2],
        [[ 3.40,  3.60], [ 2.20,  2.50],  PI/2],
        [[-0.60, -0.40], [ 0.30,  0.90], -PI/2],
        [[-3.20, -2.00], [-4.60, -4.40],  PI],
        [[-2.60, -2.40], [-0.80,  2.30],  PI/2],
]
"""

# Initialisation
supervisor = Supervisor()
basicTimeStep = int(supervisor.getBasicTimeStep())

# Receiver et emitter
receiver = supervisor.getDevice("receiver")
receiver.enable(RECEIVER_SAMPLING_PERIOD)
emitter = supervisor.getDevice("emitter")
packet_number = 0

# Recuperation des liens vers les noeuds voitures
tt_02 = supervisor.getFromDef("TT02_2023b_RL")
tt_02_translation = tt_02.getField("translation")
tt_02_rotation = tt_02.getField("rotation")
sparringpartner_car_nodes = [supervisor.getFromDef(f"sparringpartner_car_{i}") for i in range(NB_SPARRINGPARTNER_CARS)]
sparringpartner_car_translation_fields = [sparringpartner_car_nodes[i].getField("translation") for i in range(NB_SPARRINGPARTNER_CARS)]
sparringpartner_car_rotation_fields = [sparringpartner_car_nodes[i].getField("rotation") for i in range(NB_SPARRINGPARTNER_CARS)]

erreur_position = 0
```

Positions de départ. Le superviseur choisit pour la voiture apprenante et pour les voitures sparring partner une position dans les plages de valeur proposées avec un angle aléatoire autour de l'angle proposé.

Idem pour le circuit de validation

Configuration des émetteur et récepteur pour la communication avec la voiture apprenante

Récupération des liens vers la voiture apprenante et vers les voitures sparring partner pour contrôler ensuite leur position pour les débuts d'épisodes

Compteur des erreurs de réinitialisation

Figure 14 : Première partie du code python du superviseur : initialisations

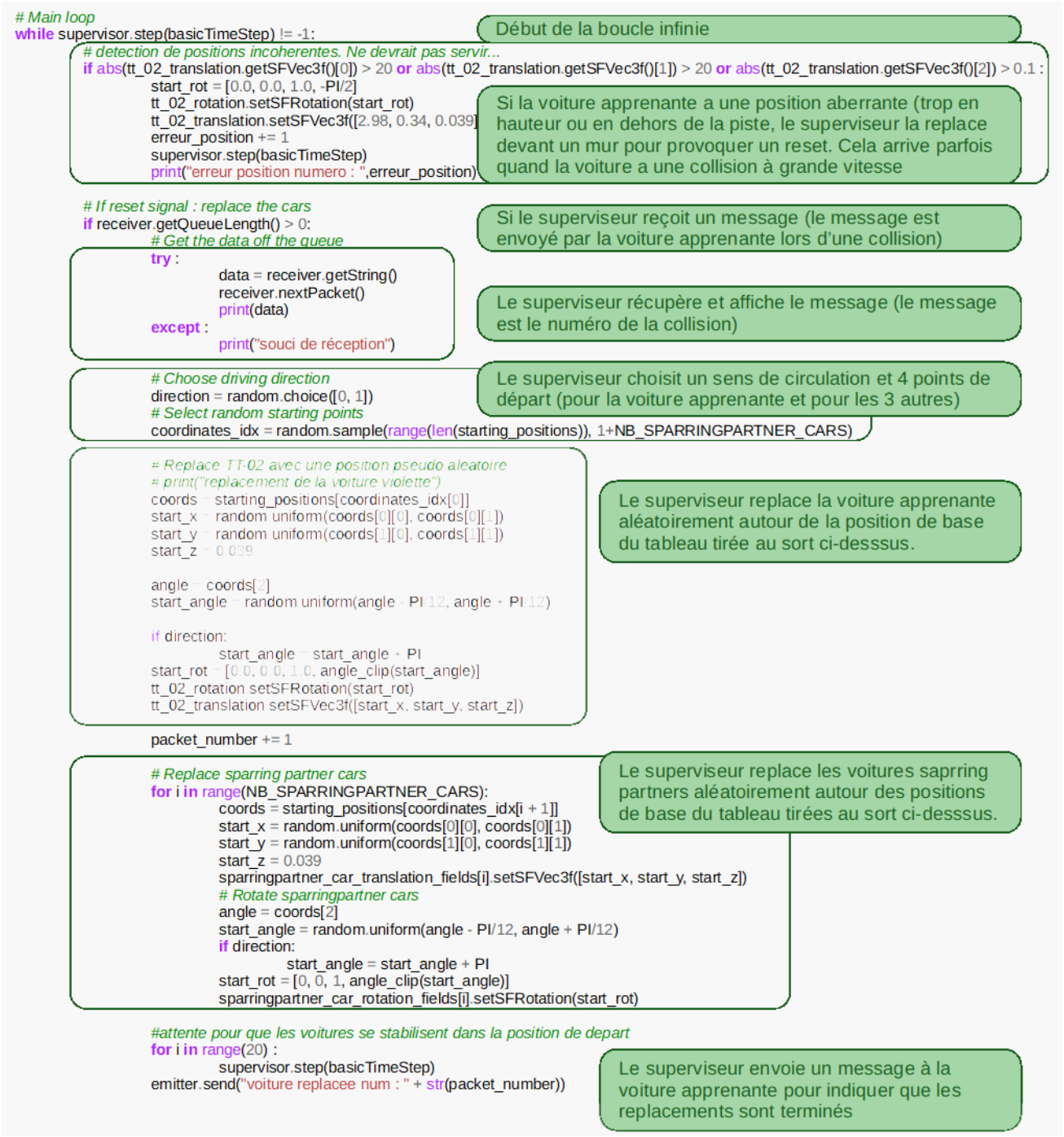


Figure 15 : Seconde partie du code python du superviseur : la boucle principale

3.3 - L'agent (le contrôleur de la voiture) et l'interface Gymnasium

Gymnasium (nommée Gym à sa création) est une bibliothèque Python développée par OpenAI dont l'évolution est, depuis peu, suivie par Farama Foundation, qui permet de gérer l'apprentissage par renforcement de manière normalisée, faisant le lien entre l'environnement (Webots ici mais Gymnasium en propose plusieurs) et l'algorithme d'apprentissage de la politique de l'agent (le contrôleur de conduite de la voiture ici).

Pour plus de détails sur Gymnasium et son utilisation, se référer à la ressource « *Introduction aux bibliothèques Gym et Stable-Baselines pour l'apprentissage par renforcement* » [4] du « *Dossier Intelligence Artificielle* » [2].

Gymnasium (et numpy, demandé par Gymnasium) s'installe à l'aide de la commande suivante :

```
pip3 install numpy
pip3 install gymnasium
```

Dans le cas de la course de voitures autonomes, un premier travail est de faire le lien entre Gymnasium et le « monde » Webots décrit ci-dessus. Gymnasium exige qu'un environnement contienne les fonctions suivantes :

- *get_observation()* : fonction renvoyant les observations de l'environnement.
- *get_reward()* : fonction donnant la récompense selon l'action effectuée par l'agent.
- *reset()* : fonction exécutant la démarche pour repartir au début d'un épisode.
- *step()* : fonction faisant évoluer l'environnement d'un pas.

Toutes ces fonctions sont rassemblées dans une classe nommée ici « WebotsGymEnvironment ». Dans cette classe ont été rajoutées les trois fonctions propres à la voiture, décrites dans l'article sur le simulateur [9] :

- *get_Lidar_mm()* : fonction qui renvoie un tableau des valeurs acquises par le Lidar dans le même format que la voiture réelle avec des valeurs cohérentes en mm.
- *set_vitesse_m_s()* : fonction qui prend en argument une vitesse en $m.s^{-1}$ pour contrôler la propulsion de la voiture. Cette fonction existe aussi sur la voiture réelle.
- *set_direction_degre()* : Fonction qui prend en argument un angle en degré pour contrôler la direction de la voiture. Cette fonction existe aussi sur la voiture réelle.

Tout commence par la construction de l'environnement avec **la fonction *__init__()***

```python
class WebotsGymEnvironment(Driver, gym.Env):
        def __init__(self):
                super().__init__()

                #valeur initiale des actions
                self.consigne_angle = 0.0          # en degres
                self.consigne_vitesse = 0.1        # en m/s

                #compteur servant à la supervision de l'apprentissage
                self.numero_crash = 0              # compteur de collisions
                self.nb_pb_lidar = 0               # compteur de problèmes de communication avec le lidar
                self.nb_pb_acqui_lidar=0           # compteur de problèmes d'acquisition lidar
                self.nb_demarrage_lidar=0          # compteur de démarrages de lidar
                self.reset_counter = 0    # compteur de pas d'apprentissage pour arrêter un épisode après n pas
                self.packet_number = 0             # compteur de messages envoyés

                # Emitter / Receiver
                self.emitter = super().getDevice("emitter")
                self.receiver = super().getDevice("receiver")
                self.receiver.enable(RECEIVER_SAMPLING_PERIOD)

                # Lidar initialisation
                self.lidar = super().getDevice("RpLidarA2")
                self.lidar.enable(int(super().getBasicTimeStep()))
                self.lidar.enablePointCloud()

                # Action space
                self.action_space = gym.spaces.Box(low=np.array([-1, -1]), high=np.array([1, 1]), dtype=np.float32)

                # Observation space (description de l'espace d'observation : lidar, lidar_1, vitesse_1 et direction_1)
                self.observation_space = gym.spaces.Dict({
                "current_lidar":gym.spaces.Box(np.zeros(201), np.ones(201), dtype=np.float64),
                "previous_lidar":gym.spaces.Box(np.zeros(201), np.ones(201), dtype=np.float64),
                "previous_speed":gym.spaces.Box(np.array([0]), np.array([1]), dtype=np.float64),
                "previous_angle":gym.spaces.Box(np.array([-1]), np.array([1]), dtype=np.float64),
                })
```

Figure 16 : Code python de la fonction __init__()

Il est nécessaire de décrire les espaces d'action et d'observation dans la classe de l'environnement Gymnasium. Ici, pour l'espace d'action :

- Un scalaire compris entre -1 et 1 pour l'incrément en vitesse
- Un scalaire compris entre -1 et 1 pour l'incrément en angle

Les valeurs sont normalisées car Stable-Baselines3 préfère des valeurs normalisées. Ce n'est que par la suite que l'on multiplie par 0.5 m/s pour la vitesse et 9° pour l'angle.

L'espace d'observation est le suivant :

- Un tableau de 201 scalaires compris entre 0 et 1 pour les données actuelles du Lidar
- Un tableau de 201 scalaires compris entre 0 et 1 pour les données précédentes du Lidar
- Un scalaire compris entre 0 et 1 pour la vitesse actuelle de la voiture (la consigne)
- Un scalaire compris entre -1 et 1 pour la direction actuelle de la voiture (la consigne)

Les valeurs sont normalisées en les divisant par leurs valeurs maximales possibles respectives.

La fonction *get_observation()*

```
# Get Lidar observation
def get_observation(self,init=False):
        tableau_lidar_mm = self.get_lidar_mm()
        i = 0
        while(tableau_lidar_mm[0] == 0) and (i<50) :
                #on essaie d'avoir un tableau de valeur correct !
                self.nb_pb_acqui_lidar+=1
                print("souci d'aquisition lidar"+str(self.nb_pb_acqui_lidar))
                i = i+1
                tableau_lidar_mm = self.get_lidar_mm() #lidar en mm

        if init:
                current_lidar=tableau_lidar_mm.astype("float64")/12000
                previous_lidar=tableau_lidar_mm.astype("float64")/12000
                previous_speed=[0]
                previous_angle=[0]
        else:
                #grandeurs normalisées pour observation
                previous_lidar=self.observation["current_lidar"]
                current_lidar=tableau_lidar_mm.astype("float64")/12000
                previous_speed=[self.consigne_vitesse/VITESSE_MAX_M_S]
                # si on a un capteur de vitesse sur la voiture relle :
                # previous_speed=[(super().getTargetCruisingSpeed()/3.6)/VITESS
                previous_angle=[self.consigne_angle/MAXANGLE_DEGRE]
                # si on a un capteur de vitesse sur la voiture relle :
                # previous_speed=[(super().getSteeringAngle()*180/PI)/MAXANGLE

        observation = {
                        "current_lidar": current_lidar,
                        "previous_lidar":previous_lidar,
                        "previous_speed":previous_speed,
                        "previous_angle":previous_angle,
                }
        # print(observation["current_lidar"])
        self.observation=observation
        return observation
```

Récupération du tableau des données lidar, en mm

Tant que le tableau est incorrect (on ne peut avoir 0 cm devant le lidar, on recommence l'acquisition

Au début d'un nouvel épisode, on met la valeur actuelle de tableau lidar dans la valeur du précédent.

On remplit les 4 champs de l'observation ; valeur actuelle du lidar, valeur précédente et valeurs des commandes de vitesse et de direction. Les valeurs sont normalisées de sorte d'être entre -1 et +1

On conserve observation dans self.observation (car la valeur actuelle du lidar sera la valeur précédente au prochain pas)

Figure 17 : Code python de la fonction get_observation()

La fonction renvoie les valeurs actuelles normalisées du Lidar à « l'instant présent », les valeurs à « l'instant précédent », (récupérées de l'observation précédente), ainsi que les valeurs normalisées des commandes de direction et de vitesse. Si la fonction est appelée depuis la fonction *reset()* (init = True), on donne le même tableau pour les deux parties de l'espace d'observation concernant le Lidar puisque la voiture a été repositionnée.

Ce sont ces données qui seront les entrées du réseau de neurones.

La fonction *get_reward()*

```
# Reward function
def get_reward(self, obs):
        reward = 0
        done = False
        mini = 1
        #recherche de la distance la plus faible mesuree par le lidar entre -40et +40°
        for i in range(-40,40) :
                if (obs["current_lidar"][i] < mini and obs["current_lidar"][i]!=0) :
                        mini = obs["current_lidar"][i]
        # print(mini)
        #si le lidar touche un mur ou si la voiture va trop vite (chute en dehors du sol de la piste)
        if mini < 0.014 or super().getCurrentSpeed() > 30.0 : #0.014 <-> 160 m
                # Crash
                self.numero_crash += 1
                reward = -300
                done = True

        else:
                #Récompense pour une grande distance aux obstacles et une grande vitesse
                reward = 12 * (mini-0.014) + 3 * super().getTargetCruisingSpeed()
                #print("reward : "+str(reward))

        #Reset si la voiture a fait beaucoup de pas
        self.reset_counter += 1
        if self.reset_counter % RESET_STEP == 0:
                print("Reset")
                done = True

        return reward, done
```

On vérifie si il y a collision en regardant la distance la plus faible à l'avant de la voiture (entre -40° et +40°)

Si il y a collision (160 mm correspondant à la longueur du capot devant le lidar) ou si la vitesse de la voiture est aberrante, done = True et la récompense est très mauvaise (-300)

Si il n'y a pas collision, la récompense valorise une grande distance aux obstacles (une valeur de mini importante) et une grande vitesse.

Si la voiture a fait plus de 16384 pas, on demande à démarrer un nouvel épisode (done = True)

Figure 18 : Code python de la fonction get_reward()

La fonction *get_reward()* attribue la récompense associée à l'état dans lequel se trouve la voiture et indique si l'épisode est terminé (variable done). On distingue deux états possibles pour la voiture. Le premier état est celui d'une collision. Dans le cas de la collision, on donne un malus de -300 « points ». Le deuxième état regroupe toutes les situations autres que celle de collision. On donne comme récompense ici une valeur dépendant de la vitesse actuelle de la voiture ainsi que la distance minimale donnée par le tableau de Lidar à l'avant de la voiture. Les fonctions de récompenses seront détaillées plus tard. On indique aussi ici si l'on a terminé l'épisode via la variable done.

Le calcul de la récompense est très important dans l'apprentissage. Il faut donner des récompenses à la fois pour valider de petites avancées (être loin des obstacles) et pour atteindre l'objectif (aller vite). Trop d'importance donnée à la vitesse amène la voiture à aller très vite en ligne droite pour s'écraser au premier virage. Trop peu d'importance donnée à la vitesse amène la voiture à rouler moins vite, au milieu de la piste.

La fonction *reset()*

```
# Reset the simulation
def reset(self):
        # Reset speed and steering angle et attente de l'arrêt de la voiture
        self.consigne_vitesse = 0.0
        self.consigne_angle = 0.0
        self.set_vitesse_m_s(self.consigne_vitesse )
        self.set_direction_degre(self.consigne_angle)
        for i in range(20) :
                super().step()
        self.reset_counter = 0
        print("crash num : " +str(self.numero_crash))

        if(self.numero_crash != 0):
                #attente de l'arrêt de la voiture
                while abs(super().getTargetCruisingSpeed()) >= 0.001 :
                        print("voiture pas encore arrêtée")
                        super().step()

                # Return an observation
                self.packet_number += 1
                # Envoi du signal de reset au Superviseur pour replacer les voitures
                self.emitter.send("voiture crash numero " + str(self.packet_number))
                super().step()
                #attente de la remise en place des voitures
                while(self.receiver.getQueueLength() == 0) :
                        self.set_vitesse_m_s(self.consigne_vitesse )
                        super().step()

                data = self.receiver.getString()
                self.receiver.nextPacket()
                print(data)

                self.consigne_vitesse = 0
                self.consigne_angle = 0
                self.set_vitesse_m_s(self.consigne_vitesse )
                self.set_direction_degre(self.consigne_angle)
                #on fait quelques pas à l'arrêt pour stabiliser la voiture si besoin
                while abs(super().getTargetCruisingSpeed()) >= 0.001 :
                        print("voiture pas arrêtée")
                        self.set_vitesse_m_s(self.consigne_vitesse )
                        super().step()

        return self.get_observation(True)
```

Arrêt de la voiture (avec quelques pas de simulation pour qu'elle s'arrête réellement), remise à zéro des compteurs

Attente de l'arrêt de la voiture (a priori inutile désormais)

Envoi du message au superviseur indiquant qu'il faut lancer un nouvel épisode (et donc repositionner les voitures)

Attente du message du superviseur indiquant que les voitures sont repositionnées (on en profite pour arrêter de nouveau la voiture, ce qui devrait être inutile normalement…)

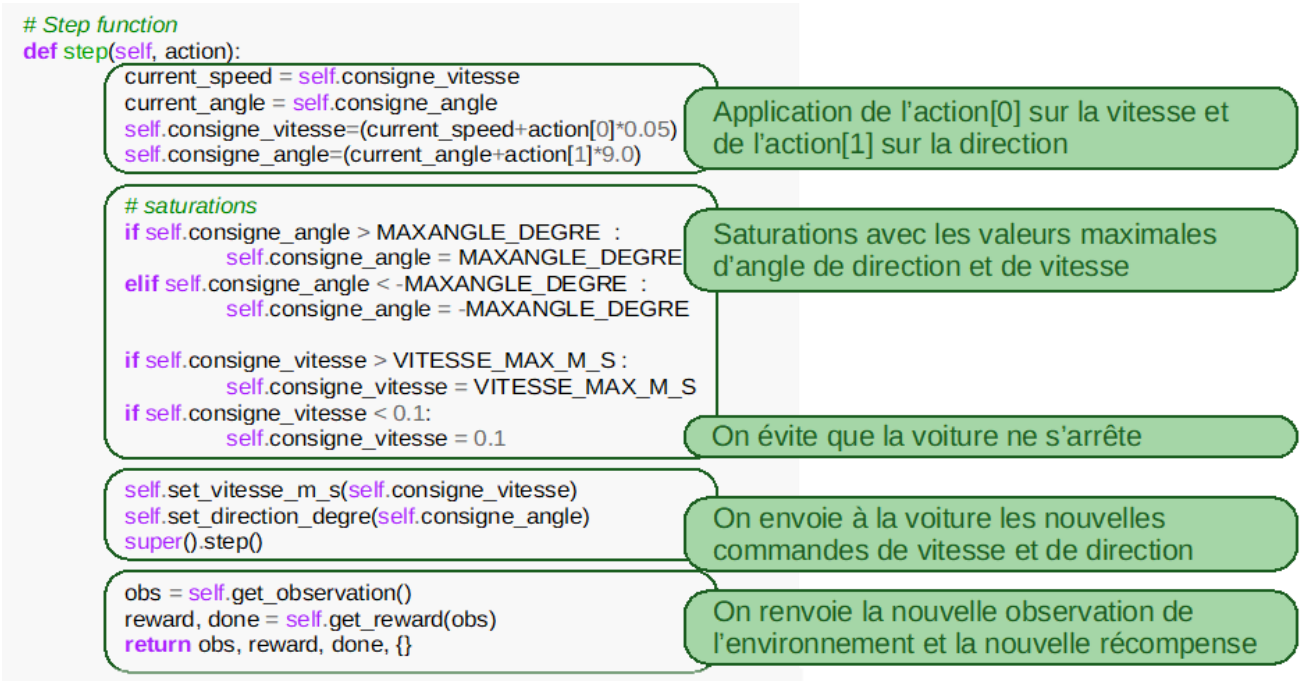
Affichage du message reçu

Arrêt de la voiture, parfois mise en mouvement par le déplacement par le superviseur

Figure 19 : Code python de la fonction reset()

La fonction *reset()* est lancée dans le cas d'une collision de la voiture ou si le nombre d'actions autorisées par épisode est atteint. La fonction *reset()* démarre un nouvel épisode. Pour cela, elle envoie un message au superviseur et attend qu'il ait fini le repositionnement des voitures. En plus de cela, plusieurs instructions visent à arrêter les voitures. En effet, le déplacement par le superviseur d'une voiture non arrêtée engendre des mouvements incohérents, d'où les multiples instructions d'arrêt de la voiture et d'attente qu'elle soit stabilisée.

La fonction *step()*




```
# Step function
def step(self, action):
        current_speed = self.consigne_vitesse
        current_angle = self.consigne_angle
        self.consigne_vitesse=(current_speed+action[0]*0.05)
        self.consigne_angle=(current_angle+action[1]*9.0)

        # saturations
        if self.consigne_angle > MAXANGLE_DEGRE  :
                self.consigne_angle = MAXANGLE_DEGRE
        elif self.consigne_angle < -MAXANGLE_DEGRE  :
                self.consigne_angle = -MAXANGLE_DEGRE

        if self.consigne_vitesse > VITESSE_MAX_M_S :
                self.consigne_vitesse = VITESSE_MAX_M_S
        if self.consigne_vitesse < 0.1:
                self.consigne_vitesse = 0.1

        self.set_vitesse_m_s(self.consigne_vitesse)
        self.set_direction_degre(self.consigne_angle)
        super().step()

        obs = self.get_observation()
        reward, done = self.get_reward(obs)
        return obs, reward, done, {}
```

Figure 20 : Code python de la fonction step()

La fonction *step()* correspond à un pas dans le processus d'apprentissage. Dans cette fonction on fait avancer la voiture avec les actions issues de la politique d'action de l'agent (compromis entre hasard/exploration et exploitation du réseau de neurones pendant l'apprentissage). Ensuite, on récupère une observation depuis le Lidar et on calcule la récompense avant de retourner toutes les informations obtenues.

Une fois créée l'interface avec l'environnement au format Gymnasium, il est possible d'utiliser une bibliothèque d'apprentissage par renforcement.

3.4 - Bibliothèque Stable-Baselines3

La bibliothèque Stable-Baselines3 propose plusieurs algorithmes d'apprentissage par renforcement, plus ou moins adaptés pour des problèmes différents. Après plusieurs essais, c'est l'algorithme PPO (Proximal Policy Optimization) qui a donné les meilleurs résultats. Il se caractérise par une évolution des paramètres évitant les discontinuités dans les résultats.

Pour installer Stable-Baselines3, on utilise la commande :

```
pip3 install stable-baselines3
```

La bibliothèque Stable-Baselines3 permet de plus de créer le réseau de neurones (*model* =) avant d'utiliser l'algorithme d'apprentissage pour en optimiser les poids (*model.learn()*). Elle propose un large choix de paramètres pour l'apprentissage.

Pour plus de détails sur la bibliothèque Stable-Baselines3 et son utilisation, se référer à la ressource « *Introduction aux bibliothèques Gym et Stable-Baselines pour l'apprentissage par renforcement* » [4] du « *Dossier Intelligence Artificielle* » [2] et au site web officiel de Stable-Baselines3 [13].

Voici un descriptif des quelques paramètres utilisés ainsi que les valeurs choisies dans le programme de référence :

- ***policy*** : Type de politique utilisée. Ici, pour de multiples entrées ('MultiInputPolicy')
- ***env*** : Environnement d'apprentissage Gym

- ***learning_rate*** : Facteur d'apprentissage, il correspond à l'importance donné à un nouveau pas par rapport à ce qui a été acquis avant. Plus il est faible, plus l'apprentissage est lent et stable (5.10−4)
- ***verbose*** : indique si on affiche les résultats (récompense acquises, nombre de pas avant crash…) en cours d'apprentissage
- **device** = 'cpu' : choisit une exécution uniquement sur le cpu ou avec la carte vidéo ('cuda')
- **tensorboard_log** : Emplacement pour enregistrer les données de Tensorboard

Les autres paramètres (décrits sur le site web de Stable-Baselines3) n'ont pas été explorés et laissés à leur valeur par défaut.

```
def main():
        t0 = time.time()                        # variable t0 pour mesurer la durée de l'apprentissage
        env = WebotsGymEnvironment()            # création de l'environnement
        print("environnement créé")
        check_env(env)                          # vérification de l'environnement
        print("vérification de l'environnement")

        # Model definition                      # paramètres de l'environnement pour l'apprentissage
        model = PPO(policy="MultiInputPolicy",
                env=env,
                learning_rate=5e-4,
                verbose=1,
                device='cpu',
                tensorboard_log='./PPO_Tensorboard',)

        ##########################################################################
        # A lancer pour rejouer un réseau déjà entraîné, à commenter pour entraîner un réseau
        ##########################################################################
        # Load learning data
        # print("demonstration")
        # model = PPO.load("PPO_results_20240516")
        ##########################################################################

        ##########################################################################
        # A lancer pour entraîner le réseau, à commenter pour rejouer un réseau déjà entraîné
        ##########################################################################
        print("début de l'apprentissage")
        model.learn(total_timesteps=1000000)            # lancement de l'apprentissage, pour 1 M de pas
        t1 = time.time()                                # variable t1 pour mesurer la durée de l'apprentissage
        print("fin de l'apprentissage après " + str(t1-t0) + "secondes")
        # Save learning data
        model.save("PPO_results_20240517")              # sauvegarde des poids du réseau entraîné
        ##########################################################################

        ###########################################
        # Démonstration du fonctionnement sur 10000 pas
        obs,_ = env.reset()             # RAZ de l'environnement et récupération de l'observation
        print("Demo of the results.")
        c = 0                           # RAZ du cumul de la récompense
        for _ in range(10000):
        # Play the demo
                action, _ = model.predict(obs, deterministic=True) # choix de l'action à partir du modèle (le réseau)
                obs, reward, done, _, _ = env.step(action)      # exécution de l'action et récup. de l'observation
                c += reward                                     # et de la récompense et d'un éventel crash
                if done:                                        # RAZ si crash
                        obs,_ = env.reset()
                print("Exiting.")                               # Affichage de la récompense cumulée
                print(c)                                        # au bout des 10000 pas

if __name__ == '__main__' :
        main()
```

Figure 21 : Création du réseau, lancement de l'apprentissage (model.learn()), sauvegarde du réseau et démonstration du résultat

La bibliothèque permet de lancer l'apprentissage grâce à la fonction *learn()*. Cette fonction prend en paramètre le nombre d'épisodes que va comporter la phase d'apprentissage. Une fois l'apprentissage terminé, il est possible de sauvegarder le réseau de neurones ce qui est particulièrement intéressant pour notre projet dans le but de charger ce réseau dans la voiture réelle. La fonction *load()* permet notamment de charger un réseau de neurones déjà entraîné.

Il est également possible de repartir d'un réseau entraîné en combinant *load* et *learn* :

```
model=PPO.load("nom_du_reseau_entraine")
print("modèle chargé")
model.set_env(env)
print("set_env effectué")
# Training
t0 = time.time()
print("début de l'apprentissage")
model.learn(total_timesteps=12e7,reset_num_timesteps=False)
t1 = time.time()
print("fin de l'apprentissage en t = " + str(t1-t0) + " s")
# Save learning data
model.save("nom_du_reseau_entraine_v2")
```

Stable-Baselines, si l'argument *verbose* est à 1 lors de la création de l'environnement, donne de manière périodique un retour sur l'apprentissage avec notamment la moyenne du nombre de pas des derniers épisodes (ep_len_mean) et la récompense totale moyenne (ep_rew_mean).

```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 345         |
|    ep_rew_mean          | 3.06e+03    |
| time/                   |             |
|    fps                  | 103         |
|    iterations           | 137         |
|    time_elapsed         | 2706        |
|    total_timesteps      | 280576      |
| train/                  |             |
|    approx_kl            | 0.0048211683 |
|    clip_fraction        | 0.0355      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.8        |
|    explained_variance   | 0.815       |
|    learning_rate        | 0.0005      |
|    loss                 | 6.74e+03    |
|    n_updates            | 1360        |
|    policy_gradient_loss | -0.00103    |
|    std                  | 0.595       |
|    value_loss           | 1.75e+04    |
-----------------------------------------
```

Figure 22: Messages Stable Baselines en cours d'apprentissage

3.5 - Tensorboard

Stable-Baselines3 est compatible avec tensorboard, outil très populaire de monitoring de l'évolution des grandeurs d'apprentissage. Il fait partie de la bibliothèque open source tensorflow développée par Google.

Le dossier de sauvegarde de Tensorboard est donné en paramètre du modèle lors de la création de l'environnement (tensorboard_log =).

Lancé alors au début de l'apprentissage, Tensorboard fait l'acquisition d'un certain nombre de valeurs représentatives de la performance du système (dont la récompense moyenne totale d'un épisode ep_rew_mean) et propose un tableau de représentations graphiques des grandeurs acquises, permettant ainsi de comparer les performances de différents set d'hyperparamètres.

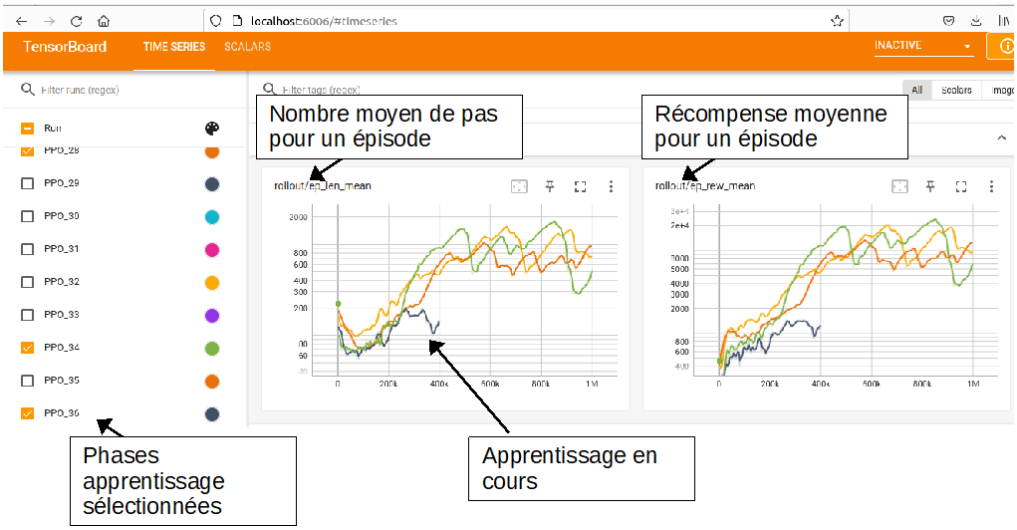


Figure 23 : Tableau d'affichage d'évolution des grandeurs d'apprentissage tensorboard

L'outil est décrit en détail sur sa page web [14]. Il s'installe comme un module python.

```
Pip3 install tensorboard
```

Une fois installé, et après avoir lancé au moins une fois un apprentissage avec tensorboard en paramètre, le tableau tensorboard est créé par la commande tensorboard suivi du dossier où sont stockées les acquisitions (dossier qui est par défaut à l'emplacement du fichier python de l'apprentissage) :

```
tensorboard –logdir PPO_Tensorboard
```

ou

```
python3 -m tensorboard.main –logdir PPO_Tensorboard/
```

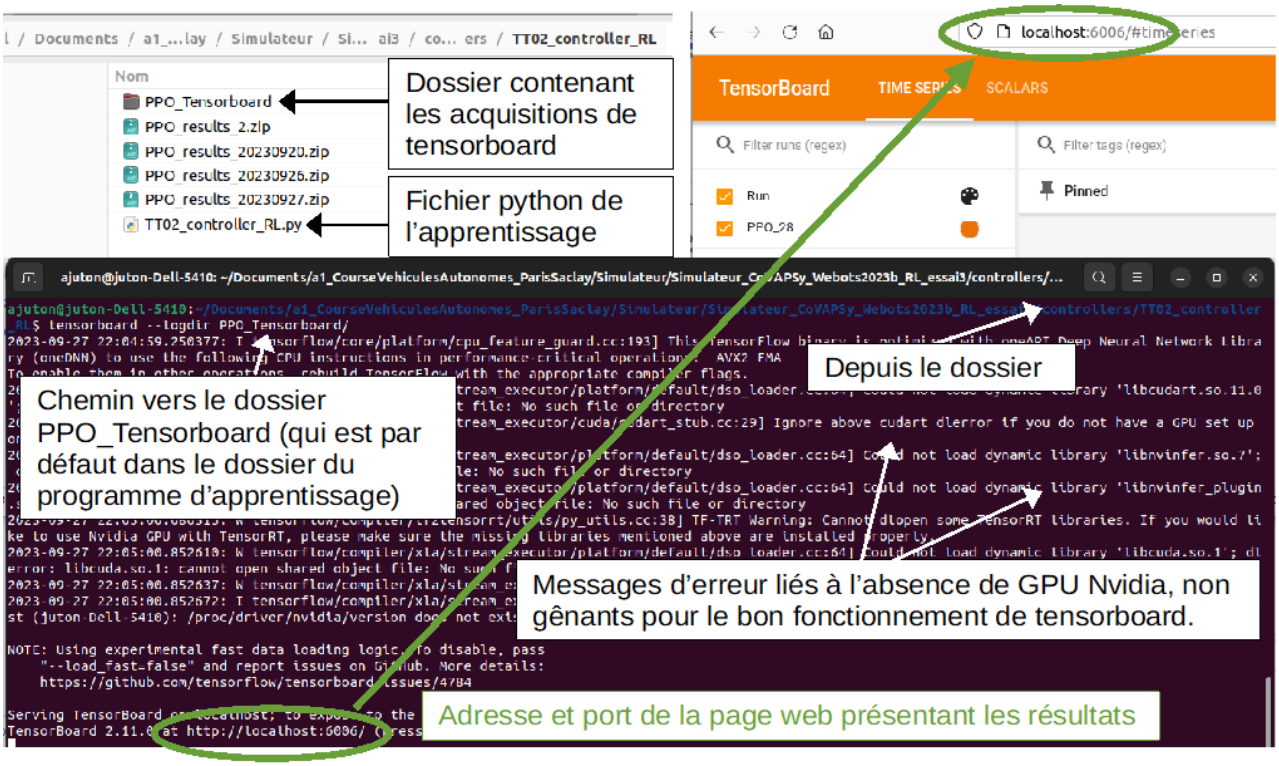


Figure 24 : Exécution de tensorboard

3.6 - Fonctions de récompense

La fonction de récompense est un des hyperparamètres les plus importants dans un processus d'apprentissage par renforcement. Elle est aussi un des plus problématiques. En effet, c'est la récompense qui influe sur comment l'apprentissage s'effectue. Par exemple, si l'on ne sanctionne pas suffisamment le crash de la voiture alors celle-ci pourrait avoir tendance à foncer dans le mur. Autre cas : si on ne favorise pas la vitesse alors la voiture aura tendance à rouler lentement voire être complètement à l'arrêt. Il est donc nécessaire de trouver une bonne fonction de récompense pour avoir le meilleur comportement possible. Dans cet exemple, il a été choisi de donner un malus de -300 lors d'un crash et dans toutes les autres situations la fonction suivante :

$$reward = 12 \times distanceAuMurLePlusProche + 3 \times vitesse$$

```
reward = 12 * (mini-0.014) + 3 * super().getTargetCruisingSpeed()
```

avec *mini*, la distance la plus courte du tableau de Lidar et 0,014, la distance entre le bord de la voiture et le centre du Lidar.

Cette fonction récompense amène la voiture à rouler au centre de la piste. C'est fonctionnel mais pas optimal.

4 - Exemples d'apprentissage

Les principes du simulateur et des bibliothèques nécessaires pour l'apprentissage par renforcement étant expliqués, cette partie revient en détail sur l'entraînement du réseau de neurones.

Dans le cas de la voiture autonome, il est tout d'abord nécessaire de définir un espace d'observation cohérent avec la réalité car toutes les grandeurs ne sont pas mesurables sur la voiture. Ainsi, L'espace d'observation est basé sur le Lidar qui permet de donner la distance des différents obstacles. Le Lidar de simulation est très proche du Lidar réel, ce qui rendra le transfert du réseau de la simulation vers la voiture réelle plus robuste. Ce n'est pas le cas avec une caméra aux images en simulation moins proches des images réelles (luminosité, couleurs, alentours de la piste, …).

Ont été ajoutées à l'espace d'observation les valeurs du Lidar au tour précédent (ce qui permet d'avoir une « dérivée » des valeurs du Lidar), les consignes de vitesse et de direction précédentes. Un asservissement de vitesse est prévu sur la voiture réelle pour avoir une vitesse égale à la consigne quel que soit l'état de la batterie (ce n'est pas le cas sans cet asservissement). Cet asservissement étant lié à la mesure de la vitesse réelle, il permettrait de mettre en entrée du réseau de neurones la vitesse réelle de la voiture et non la consigne, ce qui semble plus intéressant.

Plusieurs algorithmes d'entraînement de réseaux de neurones sont présents dans la Library Python Stable-Baselines3 :

- "Proximal Policy Optimization" (PPO) : Algorithme de type "policy-based"
- "Deep Q-Network" (DQN) : Algorithme de type "valued-based"
- "Soft Actor-Critic" (SAC) : Algorithme de type "actor-critic"

Au regard, de nombreux tests, l'algorithme qui a été retenu pour la suite est l'algorithme PPO, qui converge, entre autres, le plus rapidement.

Remarque : les collisions sont simplifiées par Webots. A chaque collision, il affiche un message Warning. Il est possible de ne pas afficher ces messages en cliquant droit sur la console et en sélectionnant dans Level tous les messages, sauf les warnings : *Error, Info* et *All Controller*

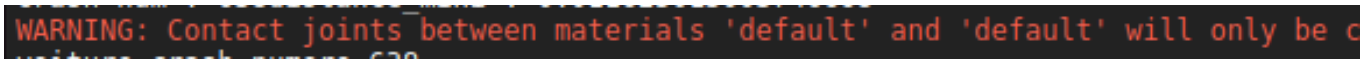


Figure 25: Message de warning obtenu lors d'une collision

4.1 - Principe de l'algorithme PPO :

L'algorithme PPO [10] a pour but d'être facile à implémenter, d'être facile à paramétrer et d'avoir une bonne efficacité au niveau des échantillons. Cet algorithme est un algorithme de type « policy-based » avec lequel on peut utiliser un ensemble d'action discret ou continu.

L'algorithme PPO utilise le principe de « *Trust Region Policy Optimization* ». Ce principe s'assure que la politique π_{old} ne soit pas trop éloignée de la politique actualisée π.

Les références sur le fonctionnement détaillé de PPO sont données dans l'article cité ci-dessus [10] et plus brièvement dans la documentation de Stable-Baselines [13].

4.2 - Modèle de réseau de neurones

Le réseau de neurones utilisé est celui proposé par défaut par Stable-Baselines / PPO, composé de deux sous-couches (*Hidden layers* en anglais). Les neurones d'entrée (*Input layer*) sont au nombre de 404 (201 entrées pour le Lidar actuel, 201 pour le Lidar précédent, 2 pour les consignes de direction et vitesse). Les neurones de sortie sont au nombre de deux correspondant à l'incrément de commande de vitesse et l'incrément de commande de direction.

Dans le but d'améliorer le comportement de la voiture, d'autres données ont été rajoutées dans l'espace d'observation : d'une part les données du Lidar à « l'instant précédent » ont été ajoutées afin de donner une continuité dans l'observation des obstacles. De l'autre, la vitesse et la direction de la voiture avant une nouvelle commande du réseau de neurones ont été ajoutées.

Le schéma ci-dessous illustre le réseau de neurones utilisé. D'autres structures ont été testées (plus de neurones par couche, plus de couches), sans apporter de gain significatif.

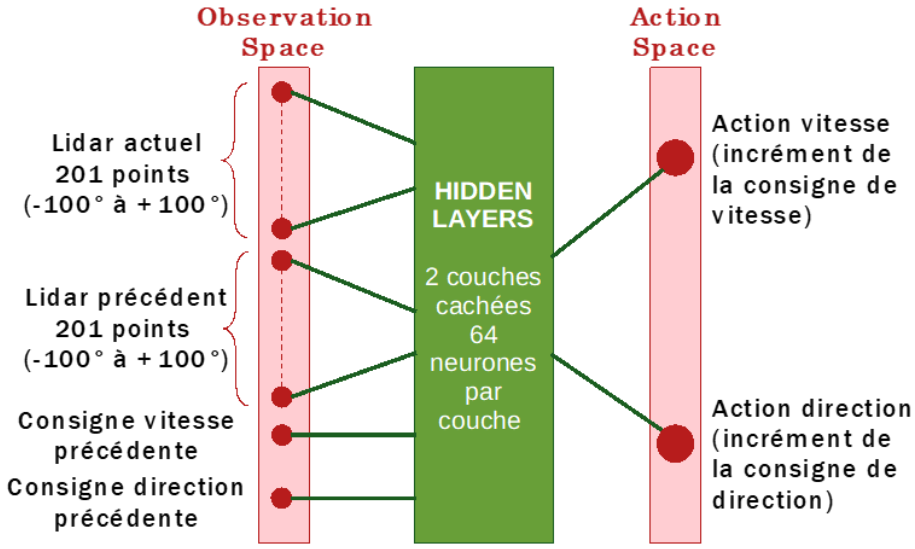


Figure 26 : Modèle du réseau de neurones utilisé

4.3 - Résultats obtenus

L'entraînement dure de 2 à 6 heures selon la puissance de l'ordinateur. Le tableau ci-dessous présente d'une part les résultats obtenus avec le modèle de réseau de neurones sur la piste d'entraînement, de l'autre les résultats obtenus en contre-la-montre sur la piste de validation :

Réseau de neurones sur la piste d'entraînement		
Réseau N°	**Particularité**	**Observation sur la piste d'entraînement**
1	Configuration de base	- Evite les obstacles - Percute parfois les autres voitures - Dépasse les voitures même dans les lignes droites
2	Lors d'un crash : $reward = -300 - n°_{crash}$	- Capable de faire plusieurs tours sans se crasher - Peut éviter les autres voitures et les dépasser - Ralentit beaucoup dans les virages
3	500 000 épisodes pour l'apprentissage	- Pas de différence notable avec les autres réseaux de neurones
4	*gae_lambda=0.90*	- Mouvement un peu hasardeux - Ne tourne pas certaines fois
5	*ent_coef=0.02*	- Fait plusieurs tour sans crash - Roule un peu moins vite que les autres réseaux des neurones
6	*ent_coef=0.02* *angle entre -2° et 2°*	- A réussi à faire un tour complet - Lente dans les virages - Assez réactive dans les lignes droites

Réseaux de neurones sur la piste de validation			
Réseau N°	**Particularité**	**Temps de passage (en minutes)**	**Observation sur la piste de validation (sans obstacles)**
1	Configuration de base	1 :03 :74	- Ne s'est pas crashée – Ralentit dans les virages
2	Lors d'un crash : $reward = -300 - n°_{crash}$	1 :31 :26	- S'est crashée une fois - A réussi mieux dans un sens - Ralentit dans les virages
3	500 000 épisodes pour l'apprentissage	2 :05 :93	- S'est crashée une fois - A réussi mieux dans un sens - Ralentit dans les virages
4	*gae_lambda=0.90*	Aucun	- S'est crashée plusieurs fois - N'arrive pas à gérer les virages après les lignes droites - Voiture la plus rapide mais incapable de faire un tour de piste
5	*ent_coef=0.02*	2 :25 :27	- Ralentit beaucoup dans les lignes droites - Réagit bien dans les virages
6	*ent_coef=0.02* *angle entre -2° et 2°*	2 :17 :49	- N'avance pas très vite - Oscille moins dans les lignes droites

Au vu de ces résultats, on peut en conclure que le réseau de neurone fourni est celui qui donne les résultats les plus satisfaisants. En revanche, lors du passage à la réalité, c'est le réseau n°5 qui a tout de même été choisi puisqu'il a un meilleur comportement sur le simulateur. Dans la réalité, il est préférable de choisir un réseau de neurones le plus stable possible (où la voiture ne se crashe pas) au détriment d'une vitesse moins importante.

5 - Passage à la réalité

Il s'agit ici de la dernière étape du projet, le transfert du réseau de neurones entraîné du simulateur à la voiture réelle.

5.1 - Implémentation dans la voiture réelle

Comme indiqué dans la partie 2, ont été développées deux fonctions similaires à celles présentes dans le simulateur : *set_vitesse_m_s()* et *set_direction_degre()*. Elles sont expliquées dans la ressource [8].

Dans le code, on remarque qu'on s'assure de ne pas laisser de valeur à 0 dans la partie « utile » du Lidar (les valeurs des indices 0 à 99 et celles de 260 à 359). Pour cela, si une valeur 0 (valeur que ne peut donner le Lidar) apparaît entre deux valeurs non nulles, on la considère comme étant la moyenne de ces deux dernières. On se retrouve ainsi dans les mêmes dispositions que la simulation au niveau du Lidar.

Le réseau de neurones, il a pu être chargé grâce la fonction *load()* de Stable-Baselines3. Il est possible d'exploiter ce réseau grâce à la fonction *predict()*, déjà utilisée dans la partie démo en fin d'apprentissage sur le simulateur. Elle prend en argument l'observation et renvoie les actions. Les observations sont stockées sous forme de dictionnaire.

Un objet a été créé pour le châssis, avec les fonctions de déplacement de la voiture :

```python
from rplidar import RPLidar
import numpy as np
import time
from rpi_hardware_pwm import HardwarePWM
import threading
from stable_baselines3 import PPO

################################################
class Chassis():

    def __init__(self):
        #Parametres a ajuster sur chaque voiture
        self.pwm_stop_prop = 7.42
        #...

        #Configuration des PWM
        self.pwm_prop = HardwarePWM(pwm_channel=0, hz=50)
        self.pwm_dir = HardwarePWM(pwm_channel=1, hz=50)

        self.vitesse_consigne = 0
        self.direction_consigne =0

    def demarrage_voiture(self):
        print("demarrage voiture")
        #Démarrage des PWM
        self.pwm_prop.start(self.pwm_stop_prop)
        self.pwm_dir.start(self.angle_pwm_centre)

    def get_direction(self,obs=False):
        if obs:
            return self.direction_consigne/self.angle_degre_max
        else:
            return self.direction_consigne

    def get_vitesse(self,obs=False):
        if obs:
            return self.vitesse_consigne/self.vitesse_max_m_s_hard
        else:
            return self.vitesse_consigne

    def set_direction_degre(self,angle_degre,adresse=1) :
        #voir article Premiers programmes python sur la voiture réelle [8]

    def set_vitesse_m_s(self,vitesse_m_s):
        #voir article Premiers programmes python sur la voiture réelle [8]

    def recule(self):
        #voir article Premiers programmes python sur la voiture réelle [8]

    def arret_voiture(self):
        self.pwm_prop.stop()
        self.pwm_dir.stop()
        print("PWM arrêtées")
```

Import du module stable_baselines3 et du module threading

Classe chassis : ensemble de fonctions liées au déplacement de la voiture. Voir [8] pour les détails

Figure 27 : Import des modules et objet Chassis()

Pour récupérer en continu les données du Lidar, pour éviter que le buffer ne déborde, on utilise une tâche *thread_scan_Lidar* chargée de récupérer les données Lidar et une tâche *thread_conduite_autonome* chargée d'exploiter les données reçues pour conduire la voiture avec le réseau de neurones.

Pour utiliser le Lidar, un objet Lidar_TT02() a été créé permettant de l'initialiser, le démarrer ainsi qu'acquérir les valeurs. Au démarrage du programme, on lance un thread *thread_scan_Lidar* qui permet de faire l'acquisition des données du Lidar en continue. On active et désactive un drapeau dans le code pour indiquer lorsque l'on veut récupérer ces données acquises. Le traitement de ces données comme décrit plus haut se fait en dehors de ce *thread*.

```
class Lidar_TT02():

    def __init__(self):
        #connexion et démarrage du lidar
        self.lidar = RPLidar("/dev/ttyUSB0",baudrate=256000)
        self.tableau_lidar_mm=np.zeros(360)
        self.acqui_lidar=np.zeros(360)
        self.drapeau_nouveau_scan = False
        self.scan_avant_en_cours = False
        self.Run_lidar = False

    def demarrage_lidar(self):
        self.lidar.connect()
        print (self.lidar.get_info())
        self.lidar.start_motor()
        time.sleep(2)

    def lidar_scan(self):
        while(self.Run_lidar == True):
            try:
                for _,_,angle_lidar,distance in self.lidar.iter_measures(scan_type='express'):
                    angle = min(359,max(0,359-int(angle_lidar)))

                    if(angle >= 260) or (angle <= 100):
                        self.acqui_lidar[angle] = distance
                    if(angle<260) and (angle>110) and (self.scan_avant_en_cours == True):
                        self.drapeau_nouveau_scan = True
                        self.scan_avant_en_cours = False
                    if(angle >= 260) or (angle <= 100):
                        self.scan_avant_en_cours = True
                    if(self.Run_lidar == False):
                        break;
            except:
                print("souci acquisition Lidar")

    def get_values(self):
            for i in range (-100,101):
                self.tableau_lidar_mm[i] = self.acqui_lidar[i]
            self.acqui_lidar = np.zeros(360)

            for i in range(-98,99):
                if self.tableau_lidar_mm[i] == 0:
                    if self.tableau_lidar_mm[i-1] != 0 and self.tableau_lidar_mm[i+1] != 0:
                        self.tableau_lidar_mm[i] = (self.tableau_lidar_mm[i]+self.tableau_lidar_mm[i])/2
            self.drapeau_nouveau_scan =  False
            return self.tableau_lidar_mm

    def get_drapeau(self):
        return self.drapeau_nouveau_scan

    def get_run(self):
        return self.Run_lidar

    def set_run(self, valeur):
        self.Run_lidar = valeur

    def arret_lidar(self):
        self.lidar.stop_motor()
        self.lidar.stop()
        time.sleep(1)
        self.lidar.disconnect()
```

Classe lidar : ensemble de fonctions liées à l'acquisition des valeurs mesurées par le lidar. Voir [8] pour les détails

Initialisation de la connexion et des variables. Démarrage du lidar, vérification de la connexion par get_info()

Fonction lancée ensuite dans un thread, en charge des acquisitions du lidar. A chaque passage de l'angle 100, un drapeau est levé pour indiquer que l'acquisition est terminée.

Fonction lancée par la tâche conduite pour récupérer les valeurs du lidar, en enlevant les éventuels points « erreur » de valeur 0.

Figure 28 : Objet Lidar_TT02()

En parallèle de ce thread (ou tâche), on lance un second thread *thread_conduite_autonome* attaché à la fonction *conduite()* qui est le programme de pilotage de la voiture. Dans la fonction *conduite()*,

on commence par récupérer les données du Lidar puis on procède au traitement de ces données. On récupère les valeurs traitées tous les 20 degrés pour faciliter l'analyse des obstacles. Dans le cas où l'on considère qu'il y a collision (valeur minimale inférieure à un certain seuil), on recule dans la direction opposée à l'obstacle détectée (indice de cette valeur minimale). Si ce n'est pas le cas, on laisse la main au réseau de neurones pour le pilotage.

```
def conduite():
        global lidar
        global voiture
        print("début conduite")
        tableau_lidar_mm = np.zeros(360)
        previous_lidar = np.zeros(360)
        while lidar.get_run() == True:
            if lidar.get_drapeau() == False:
                time.sleep(0.01)
            else:
                tableau_lidar_mm = lidar.get_values()

                min_secteur = [0]*10
                for index_secteur in range(0,10) :
                    angle_secteur = -90 + index_secteur*20
                    min_secteur[index_secteur] = 12000
                    for angle_lidar in range(angle_secteur-10,angle_secteur+10) :
                        if tableau_lidar_mm[angle_lidar] < min_secteur[index_secteur] and
tableau_lidar_mm[angle_lidar] != 0 :
                            min_secteur[index_secteur] = tableau_lidar_mm[angle_lidar]
                    if min_secteur[index_secteur] == 12000 : #aucune valeur correcte
                        min_secteur[index_secteur] = 0

                if (min_secteur[4]<=160 and min_secteur[4] !=0)\
                    or (min_secteur[3]<=160 and min_secteur[3] !=0)\
                        or (min_secteur[2]<=160 and min_secteur[2] !=0) :
                    angle_degre = -18
                    voiture.set_direction_degre(angle_degre)
                    print("mur à droite")
                    voiture.recule()

                elif (min_secteur[5]<=160 and min_secteur[5] !=0)\
                    or (min_secteur[6]<=160 and min_secteur[6] !=0)\
                        or (min_secteur[7]<=160 and min_secteur[7] !=0) :
                    angle_degre = +18
                    voiture.set_direction_degre(angle_degre)
                    print("mur à gauche")
                    voiture.recule()

                else :
                    current_lidar=tableau_lidar_mm.astype("float64")/12000
                    previous_speed=np.array([float(voiture.get_vitesse(obs=True))])
                    previous_angle=np.array([float(voiture.get_direction(obs=True))])
                    if previous_angle[0] > 1 :
                        previous_angle[0] = 1
                    elif previous_angle[0] < -1 :
                        previous_angle[0] = -1

                    obs={"current_lidar":current_lidar,
                         "previous_lidar":previous_lidar,
                         "previous_speed":previous_speed,
                         "previous_angle":previous_angle,
                         }

                    action,_= modele.predict(obs,deterministic=True)
                    angle=float(voiture.get_direction(obs=True))+action[1]*18.0

                    voiture.set_direction_degre(angle)
                    nouvelle_vitesse = float(voiture.get_vitesse())+action[0]
                    if (nouvelle_vitesse <0.1) :
                        vitesse_m_s = 0.1
                    else :
                        vitesse_m_s = nouvelle_vitesse
                    voiture.set_vitesse_m_s(vitesse_m_s)
```

Fonction conduite, en charge d'appliquer la politique de conduite. Elle sera attachée à une tâche conduite.

Tant que le drapeau indiquant l'arrivée de nouvelles données n'est pas levé, la tâche est en sommeil.

Quand le drapeau est levé, on récupère les données et la fonction get_value() abaisse le drapeau.

On cherche par secteur de 20°, la valeur minimale, pour voir si la voiture est bloquée dans un mur.

Si la voiture mesure moins de 160 mm dans le secteur 3 ou 4, elle est bloquée dans un mur à droite => elle recule en braquant vers la droite.

Si la voiture mesure moins de 160 mm dans le secteur 5 ou 6, elle est bloquée dans un mur à gauche => elle recule en braquant à gauche.

Si la voiture n'est pas bloquée dans un mur :
→ on calcule l'observation et
→ on choisit l'action à l'aide du réseau de neurones entraîné sur le simulateur
→ on applique les actions

Figure 29 : La fonction conduite() qui sera attachée à la tâche conduite_autonome()

On remarque que pour éviter une situation bloquée dans certains cas, la voiture ne peut rester immobile.

Le programme principal se contente alors de créer des instances des objets, de charger le réseau de neurones et de créer et lancer les tâches *thread_scan_Lidar()* et *thread_conduite_autonome()* présentées ci-dessus :

```
lidar= Lidar_TT02()
voiture = Chassis()

modele= PPO.load("chemin_vers_le_reseau_entraine.zip")
```

Dans le programme principal, on crée une instance lidar et une instance voiture. On charge le réseau entraîné sur le simulateur

```
while True :
    x = input("Appuyer sur 'c' pour commencer: ")
    while True:
        if x=="c":
            #connexion et démarrage du lidar
            lidar.demarrage_lidar()
            lidar.set_run(True)
            print("lidar démarré")

            voiture.demarrage_voiture()
            print("voiture démarée")

            # Création du thread lidar
            thread_scan_lidar = threading.Thread(target=lidar.lidar_scan)
            thread_scan_lidar.start()
            time.sleep(1)
```

Démarrage du lidar et de la voiture, création et lancement de la tâche thread_scan_lidar attachée à la fonction lidar_scan()

```
            x = input("Appuyer sur une touche 'g' pour commencer la course: ")
            if x=='g':
                # Création du thread conduite
                thread_conduite_autonome = threading.Thread(target = conduite)
                thread_conduite_autonome.start()
```

Création et lancement de la tâche thread_conduite_autonome attachée à la fonction conduite()

```
            while True :
                try :
                    x=input("Appuyer sur 'a' pour arrêter la voiture\n")
                    time.sleep(1)
                    if x=='a' :
                        lidar.set_run(False)
                        voiture.set_vitesse_m_s(0)
                        break
                except KeyboardInterrupt: #récupération du CTRL+C
                    print("arrêt du programme")
                    lidar.set_run(False)
                    voiture.set_vitesse_m_s(0)
                    break

            #Destruction des threads
            thread_scan_lidar.join()
            thread_conduite_autonome.join()
            #arrêt et déconnexion du lidar
            lidar.arret_lidar()
            voiture.arret_voiture()
```

Attente d'un signal pour arrêter la voiture, détruire les tâches et arrêter le lidar.

Figure 30 : Programme principal de la conduite autonome sur voiture réelle

5.2 - Résultats obtenus

Les résultats présentés ici sont ceux de la voiture réelle avec le réseau de neurones avec *ent_coef = 0.02*. En effet, il s'agit de celui qui a montré le meilleur comportement en réel.

Les résultats sont ceux réalisés le jour de la course de voitures autonomes 2023. Une première phase concerne une piste sans obstacle où le temps comptabilisé est le temps que prend la voiture pour réaliser deux tours.

Figure 31 : 1ère piste de qualification

Les résultats obtenus sont les suivants :

	1er passage	2ème passage
Temps de passage (2 tours de piste)	31 sec	25 sec

On a pu observer que sans aucun obstacle, la voiture réussit très bien à se diriger et ne prend aucun mur pendant son tour. L'apprentissage est donc concluant sur ce point puisque la conduite est très satisfaisante et conforme à ce qui est attendu. En revanche pour atteindre ce résultat, une réduction du coefficient la consigne de vitesse s'imposait. En effet, sans cette correction, la voiture allait trop vite et se prenait quelques fois les murs dans les tests effectués.

Le problème de cette correction est que sans asservissement de vitesse, celle-ci dépend de l'état de charge de la batterie. In fine, la voiture réagit bien mais est assez lente en comparaison avec les autres voitures. Aussi, on peut observer que dans les lignes droites, la voiture a tendance à vaciller contrairement à la simulation.

La deuxième phase se fait avec une nouvelle piste et des obstacles fixes, comme présenté ci-dessous.

Figure 32 : 2ème piste de qualification

Les résultats sont les suivants :

	1er passage	2ème passage
Temps de passage (2 tours de piste)	32 sec	34 sec

En présence d'obstacles fixes, la voiture se comporte un peu moins bien, l'entraînement ayant eu lieu avec quelques voitures sparring partner que la voiture agent pouvait pousser un peu avant qu'une collision ne soit détectée. Sur la piste réelle, la voiture a tendance à ne pas suffisamment tourner pour éviter l'obstacle. En revanche, c'est une bonne situation pour tester la marche arrière couplée au réseau de neurones. La voiture a pu finir ses deux tours malgré des collisions avec les obstacles. On peut conclure que la phase d'entraînement sur simulateur n'a pas suffisamment d'obstacle fixe.

Enfin, la dernière partie de l'évènement était la course en elle-même. Là encore, il y a eu deux courses. Ici pas de chrono à présenter mais quelques observations sont possibles. Lors de la première course, la voiture n'a pas pu finir la course à la suite d'un carambolage et une conduite à contre-sens. Durant la deuxième course cependant, la voiture a pu terminer la course en 2e position. Elle termine 3ème de la compétition. La vidéo « *Apprentissage par renforcement pour la conduite de voiture autonome* » [17], met en valeur le transfert de la simulation vers la réalité.

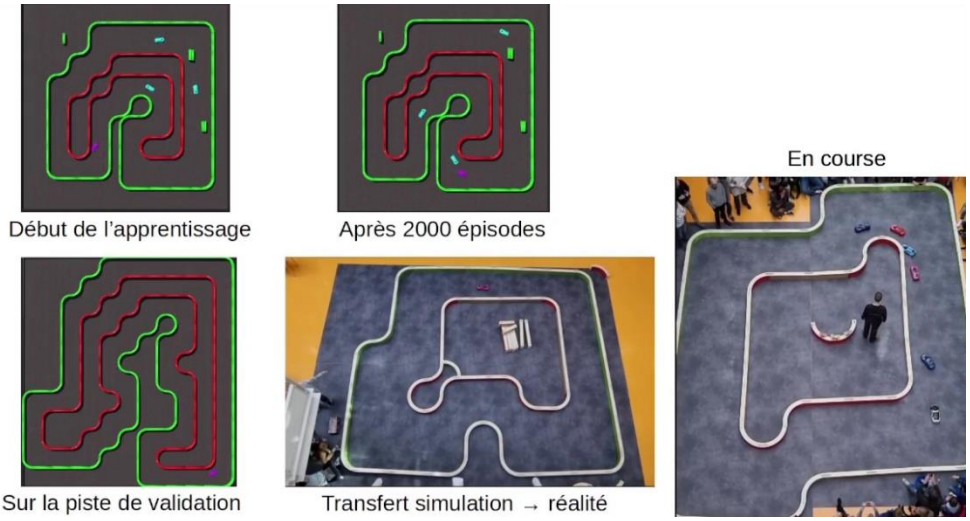


Figure 33 : Extrait de la vidéo « Apprentissage par renforcement pour la conduite de voiture autonome » [17]

On retrouve la voiture rose conduite par réseau de neurones lors de la course 2023 que l'on peut visionner sur le lien suivant : « *Course de voitures autonomes 2023* ».

5.3 - Pistes de travail

La course 2023 a mis en évidence quelques défauts, devenus pistes de travail :

- Un asservissement de vitesse via un microcontrôleur permet d'avoir une vitesse indépendante de la charge de la voiture et conforme à la consigne, améliorant la reproductibilité. On peut de plus ajouter la vitesse réelle en entrée du réseau de neurones, sur le simulateur comme sur la voiture réelle.
- Un servomoteur numérique (Dynamixel AX-12 ou HerkuleX DRS101) permet d'améliorer la dynamique de la direction. Il serait également possible de lire la position réelle du moteur, pour l'utiliser en entrée du réseau de neurones.
- La prise en compte de la dynamique (voire même aussi de la non-linéarité) de la direction réelle dans le simulateur. La direction dans le simulateur est plus dynamique que sur la voiture réelle, ce qui semble être à la source des ondulations dans les lignes droites.
- Un entraînement avec une piste (ou plusieurs pistes) plus réaliste : plus de variations sur la largeur de piste, plus d'obstacles.

- L'amélioration de la simulation par un ré-entraînement du réseau, sans chercher à rester éloigné des murs mais en ne regardant que les temps de passage.
- L'ajout d'une caméra permet d'éviter de repartir en contre-sens suite à un accident.
- La mise en œuvre de méthodes d'amélioration du passage de la simulation à la réalité [29].

En 2024, plusieurs voitures ont repris et fiabilisé ce travail, avec asservissement de vitesse pour les uns et servo-moteur numérique pour les autres. 2 voitures de l'ENS Paris Saclay ont remplacé le Lidar par une caméra pour travailler uniquement avec de la vision. Ce sont les voitures UFR Sciences, ENS Paris Saclay et Institut Villebon Georges Charpak (IVGC) de la course 2024 que l'on peut visionner sur le lien suivant : « *Course de Voitures Autonomes Paris-Saclay 2024* ».

6 - Conclusion

Nombreux sont les exemples d'apprentissage par renforcement en simulation. Beaucoup moins nombreux sont ceux qui vont jusqu'à mettre en œuvre l'inférence du réseau de neurones entraîné dans le monde réel.

Cette ressource présente, sur du matériel accessible (logiciels open-source et voiture entre 800 et 1500 euros) une application pour la mise en œuvre de l'apprentissage par renforcement en simulation et le transfert simulation à réalité.

C'est une base de travail solide, avec de nombreux apports possibles par les enseignants et étudiants qui souhaiteront travailler sur ce sujet et participer à son amélioration.

Références :

[1]: Introduction à l'apprentissage par renforcement, A. Juton, V. Noël, R. Lali, juillet 2022, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/introduction-a-lapprentissage-par-renforcement

[2]: Dossier Intelligence Artificielle, 2022, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/dossier-intelligence-artificielle

[3]: Apprentissage par renforcement de la conduite d'un véhicule sur AirSim, L. de Mattteis, S. Radosalvjevic, juillet 2022, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/apprentissage-par-renforcement-dela-conduite-dun-vehicule-sur-airsim

[4]: Introduction aux bibliothèques Gym et Stable-Baselines pour l'apprentissage par renforcement, G. Chérot, A. Godinot, juillet 2022, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/introduction-aux-bibliotheques-gym-et-stablebaselines-pour-lapprentissage-par-renforcement

[5]: Dépôt GitHub public sur les voitures autonomes https://github.com/ajuton-ens/CourseVoituresAutonomesSaclay

[6]: Site web de la course de voitures autonomes de Paris Saclay : https://ajuton-ens.github.io/CourseVoituresAutonomesSaclay/

[7]: Course Voitures Autonomes Paris Saclay (CoVAPSy) : Travaux pratiques autour des voitures autonomes, T. Boulanger, E. Délègue, K. Hoarau, A. Juton, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/covapsy-tp-autour-des-voitures-autonomes

[8]: CoVaPSy : Premiers programmes python sur la voiture réelle, T. Boulanger, E. Délègue, K. Hoarau, A. Juton, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/covapsy-premiers-programmes-python-sur-voiture-reelle

[9]: CoVaPSy : Mise en œuvre du Simulateur Webots, T. Boulanger, E. Délègue, K. Hoarau, A. Juton, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/covapsy-mise-en-oeuvre-du-simulateur-Webots

[10]: John Schlman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov. "Proximal Policy Optimization Algorithms". In : (2017), https://arxiv.org/pdf/1707.06347.pdf

[11]: Site officiel de Webots : https://cyberbotics.com/

[12]: site officiel de gymnasium : https://gymnasium.farama.org/

[13]: site officiel de Stable-Baselines3: https://stable-baselines3.readthedocs.io

[14]: site officiel de tensorboard : https://www.tensorflow.org/tensorboard?hl=fr

[15]: Annexes Apprentissage par renforcement et transfert simulation vers réalité pour la conduite de voitures autonomes, R. Bennani, K. Hoarau, A. Juton, https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/apprentissage-renforcement-transfert-simulation-vers-realite-pourla-conduite-voitures-autonomes

[16]: Wei Zhu, Xian Guo, Dai Owaki, Kyo Kutsuzawa, Mitsuhiro Hayashibe. "A Survey of Sim-to-Real Transfer Techniques Applied to Reinforcement Learning for Bioinspired Robots". In: IEEE Transactions on Neural Networks and Learning Systems (sept. 2021), https://ieeexplore.ieee.org/document/9552429

[17]: Apprentissage par renforcement pour la conduite de voiture autonome, R. Bennani, K. Hoarau, A. Juton, mai 2024, https://eduscol.education.fr/sti/si-ens-paris-sacaly/ressources_pedagogiques/apprentissage-par-renforcement-pour-la-conduite-de-voiture-autonome

Ce document est accompagné :

- **de la vidéo "Apprentissage par renforcement pour la conduite de voiture autonome"** dont le lien est https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/apprentissage-par-renforcement-pour-la-conduite-de-voiture-autonome
- **des annexes zip et py** dont le lien est https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/apprentissage-renforcement-transfert-simulation-vers-realite-pourla-conduite-voitures-autonomes

Gestion de la Charge de Batteries Lithium (BMS)

La Revue 3E.I
Culture Sciences de l'Ingénieur
école normale supérieure paris-saclay

Arnaud SIVERT[1] - Bruno VACOSSIN[1]
Sébastien CARRIERE[1] - Franck BETIN[1]

Édité le 30/04/2024

[1] *I.U.T de l'Aisne Département Génie Electrique SOISSONS FRANCE*
Laboratoire pour l'Innovation Technologique (L.T.I)

Cette ressource fait partie du N°112 de La Revue 3EI du 2ème trimestre 2024.

Bien que largement répandus, les chargeurs de batterie lithium restent difficiles à caractériser et leur qualité sujette à questions. En effet, les caractéristiques des cellules des accumulateurs présentent des disparités relativement importantes fonction de leurs vieillissements et des chimies employées et changent en fonction de l'état de charge et de la température. A cause de ces disparités, l'équilibrage est nécessaire et doit se faire en toute sécurité et à faible coût. Il est alors nécessaire de maitriser cet équilibrage quelle que soit la résistance interne des éléments, l'intensité du courant de charge, la tension d'alimentation du chargeur, la température de fonctionnement. Dans cet article, nous proposons une étude basée sur des simulations qui permet de se faire une idée claire des performances des chargeurs et des améliorations que l'on peut en attendre.

1 - Introduction

La durée de charge des batteries dépend des caractéristiques des éléments de l'accumulateur et de la gestion de ces éléments par le BMS (Battery Management System). Le BMS permet de stopper la décharge ou la charge lorsqu'un élément est respectivement en dessous de la tension critique ou au-dessus de sa tension de seuil. De même, si le courant dépasse une certaine valeur limite destructrice pour l'accumulateur, le BMS interrompt la décharge de l'accumulateur. Par mesure de sécurité, un fusible surdimensionné est utilisé au cas où l'interrupteur statique (plusieurs transistors en parallèles) du BMS se retrouve en court-circuit après un défaut.

La grandeur physique qui permet de déterminer l'arrêt de la charge ou de la décharge est en général la tension de la batterie. En effet, la mesure des résistances internes de chacun des éléments étant rarement gérée par les BMS, la tension électrochimique OCV (Open Circuit Voltage=E_B) ne peut être utilisée.

Les caractéristiques électriques de chaque élément de l'accumulateur (tension électrochimique, résistance et capacité internes) évoluent en fonction du nombre de cycles de charge et de décharge, donc de leur état de santé (SoH : State of Health) [1] mais aussi en fonction de la température et de l'état de charge (SoC : State of Charge). Ainsi, le courant de charge maximal dépend de l'évolution de la tension électrochimique de l'accumulateur, des résistances internes de chaque élément R_B, de la résistance du câble R_C.

Le courant de charge limite en mode CC s'exprime ainsi de la manière suivante [2] :

$$I_{limit} = \frac{U_{alimentation} \cdot PWM - \sum E_B(SoC, °C)}{\sum R_B(SoC, °C) + R_C} \qquad (1)$$

La régulation de courant de charge CC est nécessaire pour limiter le courant du chargeur. Pour préserver la durée de vie des éléments, le courant de décharge est en général préconisé par les fabricants à 3C pour la technologie Lipofer, NCM et à 6C pour LTO, Na-ion.

De plus, tous les BMS assurent la sécurité thermique de l'accumulateur par l'intermédiaire de plusieurs capteurs de température. En général, le paramétrage de la température max est fixé à 50°C. Ainsi le courant de consigne de la charge est limité par la nécessité de la régulation de température (CT). Une communication entre le BMS et le chargeur est alors nécessaire pour adapter la consigne de courant. Sans cette liaison, le BMS interrompt arbitrairement le courant de la batterie pour une valeur de température de 50°C (charge en tout ou rien).

La tension d'alimentation fournie par le chargeur correspond classiquement à la somme des tensions de seuil de chaque élément série. Lorsque le rapport cyclique de la PWM est égal à 1, le courant diminue naturellement en fonction de la variation de la somme de la tension E_B et correspond à la charge à tension constante (CV). Cependant, avec une tension d'alimentation plus importante, le courant de consigne pourra être plus élevé et diminuer ainsi la durée de charge [3, 5].

Le passage du mode CC au mode CV permet de définir l'état de charge SoC_{CV} par l'équation (2) qui dépend du courant de charge en CC, de la résistance interne du chargeur et de manière indépendante de la résistance interne de la batterie.

$$SoC_{CV} \approx \frac{U_{a\,lim} - \sum E_B(SoC, °C) - I_{limit} \cdot R_C}{U_{a\,lim} - \sum E_B(SoC, °C)} \quad (2)$$

Tous les BMS passifs fonctionnent de la même manière. Dès qu'il y a un élément de l'accumulateur qui atteint la tension de seuil, il y a une déviation du courant de charge via une résistance et vers une LED qui permet d'indiquer visuellement quels sont les éléments qui ont atteint cette tension de seuil et de signaler le moment où l'équilibrage est atteint pour toutes les cellules (tous les éléments sont chargés à leur tension de seuil). Si la tension de l'élément devient trop grande (tension de seuil plus une hystérésis) alors le courant de charge doit être coupé.

En fonction de la chimie et du nombre de cycles, des différences entre chaque élément série peuvent atteindre de plus de 10% sur le SoH (State of Health) et plus 400% sur la résistance interne. Ceci engendre pendant l'équilibrage, une commutation tout ou rien de l'alimentation du chargeur de fréquence relativement élevée dépendant du courant de charge, du courant de déviation, de la résistance interne de chaque élément et de l'hystérésis entre la tension de seuil et la tension maximale choisie.

Dans cet article, nous nous intéressons à la charge et l'équilibrage de petits accumulateurs dont l'énergie est inférieure à 2000 W.h et qui correspondent à des véhicules à faible consommation d'énergie ou à des équipements électroportatifs. Les simulations proposées ici permettent d'analyser le comportement du programme du microcontrôleur d'un BMS numérique, tâche difficile à réaliser en pratique.

2 - Chargeur à courant constant et BMS tout ou rien

A ce jour, certains BMS communiquent avec le chargeur (Bluetooth, bus CAN, I2C ...) pour améliorer la sécurité et optimiser la charge [4]. De plus, pour des applications aux véhicules électriques cette fonctionnalité commence à être utilisée dans le cadre d'un délestage résidentiel (chargeur Renault 5 via la technologie V2G vehicle-to-grid).

Cette communication permet d'accéder aux données de la batterie que sont :

- la tension de chaque élément série permettant de vérifier l'état de l'équilibrage,
- la température,
- la résistance interne ESR « equivalent series resistance »,
- l'autodécharge de chaque élément,
- l'estimation du SoH,
- et de la capacité restante.

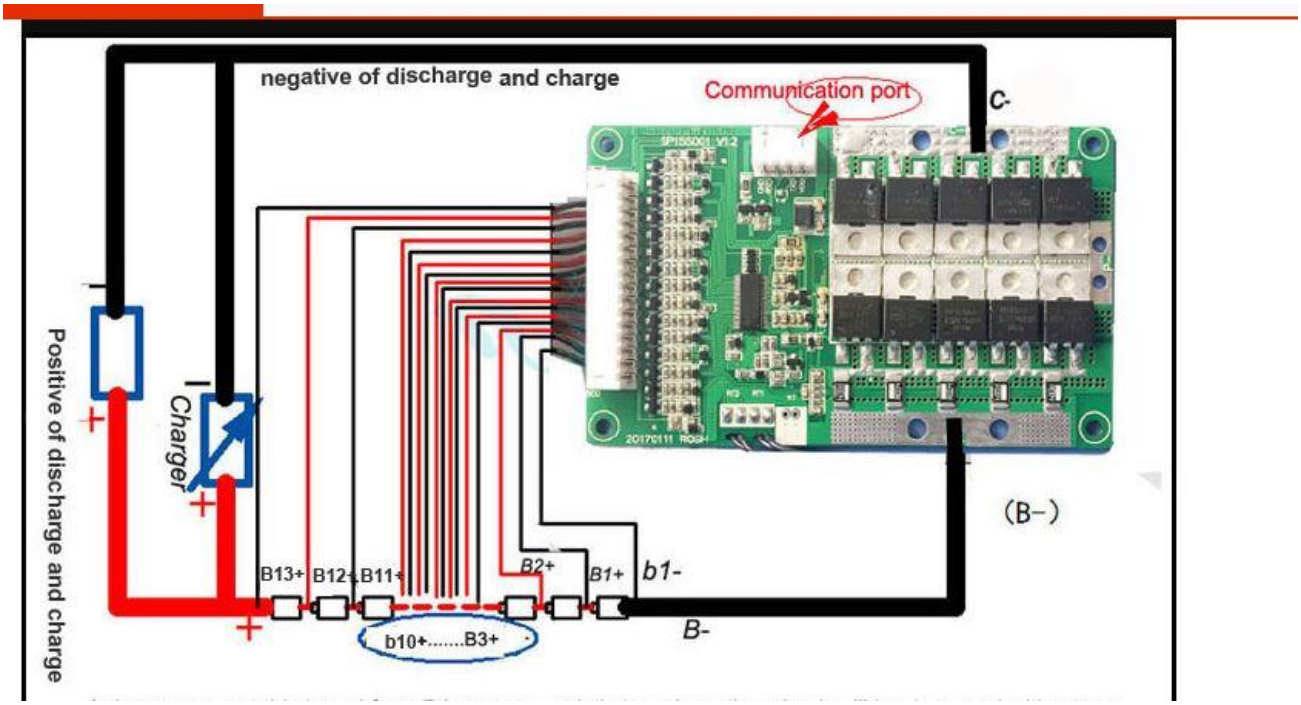


Figure 1 : BMS avec le faisceau de mesure des tensions de chaque élément de l'accumulateur et les 10 transistors MOS TO252 pour une charge ou décharge max de 40A.

Le Bluetooth permet par exemple, de collecter les données et de visualiser les courbes caractéristiques d'une batterie sur son smartphone ou PC.

Sur la figure 2, le schéma fonctionnel classique d'un BMS pour batterie lithium est présenté. Le courant d'équilibrage est limité par les transistors et les résistances nommées CBx.

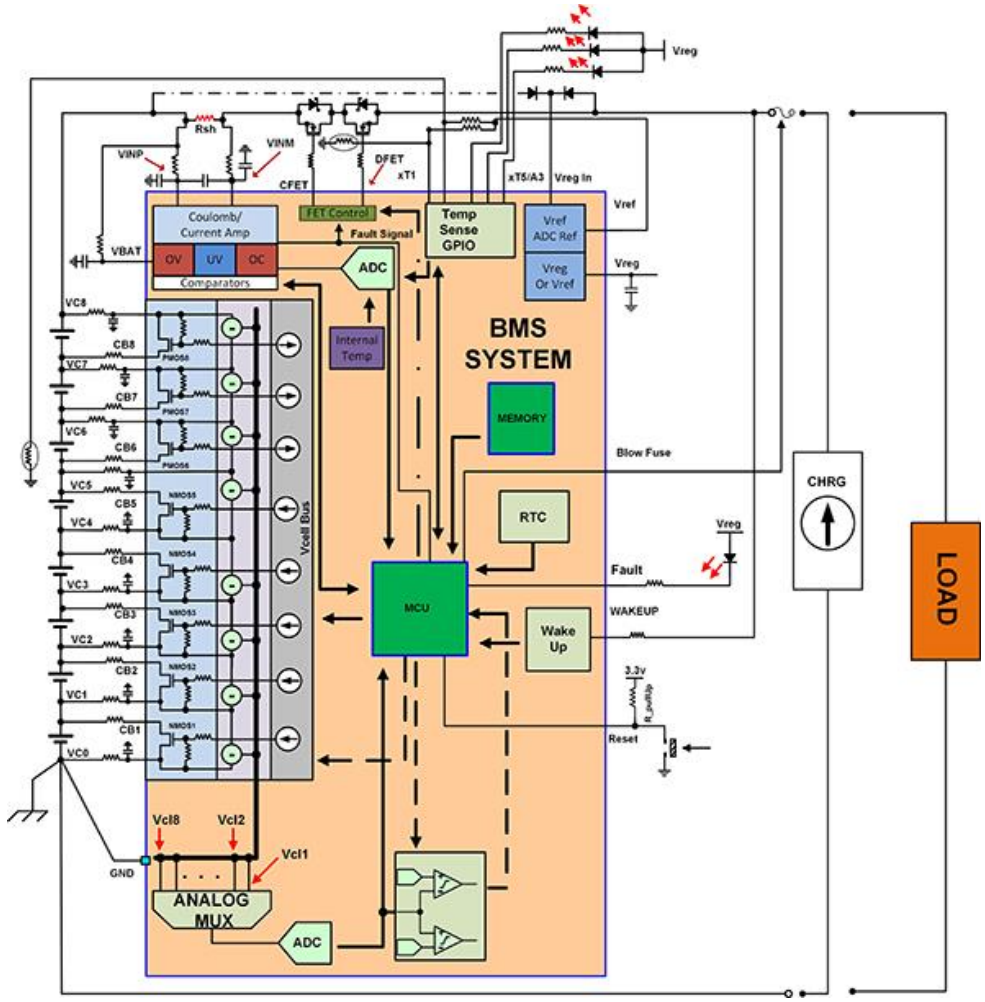


Figure 2 : Intersil BMS ISLl94203 avec charge et décharge sur la même entrée, CAN 12 bits, programmable, 8€ [6].

Pour simplifier les explications sur la problématique du BMS tout ou rien, les simulations présentées dans cet article ne comportent que 2 éléments Lipofer dont la tension de seuil est fixée à 3,7 V. La tension de coupure du BMS a été choisie arbitrairement à 3,72 V.

En pratique, l'hystérésis entre la tension maximale et la tension de seuil correspond à un décalage de 0,05 V. Le courant de déviation (équilibrage) est de 1A entre les 2 tensions précitées.

La tension d'alimentation du chargeur a donc été fixée à une tension de 3,72 V x 2.

Les 2 éléments Lipofer possèdent une capacité de 20 A.h avec un courant de charge régulé inférieur à 10 A. La tension électrochimique pour un SoC nul est de 2,5V. Si la résistance du chargeur est de 0,044 Ω, alors la régulation CC cessera à 81% (16,3A.h) pour basculer en régulation CV.

$$SoC_{CV} = \frac{2\times3.72V-2\times2.5V-10A\times0.044\Omega}{2\times3.72V-2\times2.5V} = 81\% \quad (3)$$

Pour la simulation, l'évolution de la tension électrochimique de la batterie (OCV) est supposée être une fonction linéaire de la capacité énergétique (équation (4)) :

$$E_B = \left[E_{B\,seuil(SoC=100\%)} - E_{B(SoC=0)} \right] \cdot \frac{Capacity}{20A.h} + E_{B(SoC=0)} \quad (4)$$

La tension de chaque élément correspond au modèle simplifié électrochimique d'une batterie :

$$U_B = E_B + R_B . I_B \quad (5)$$

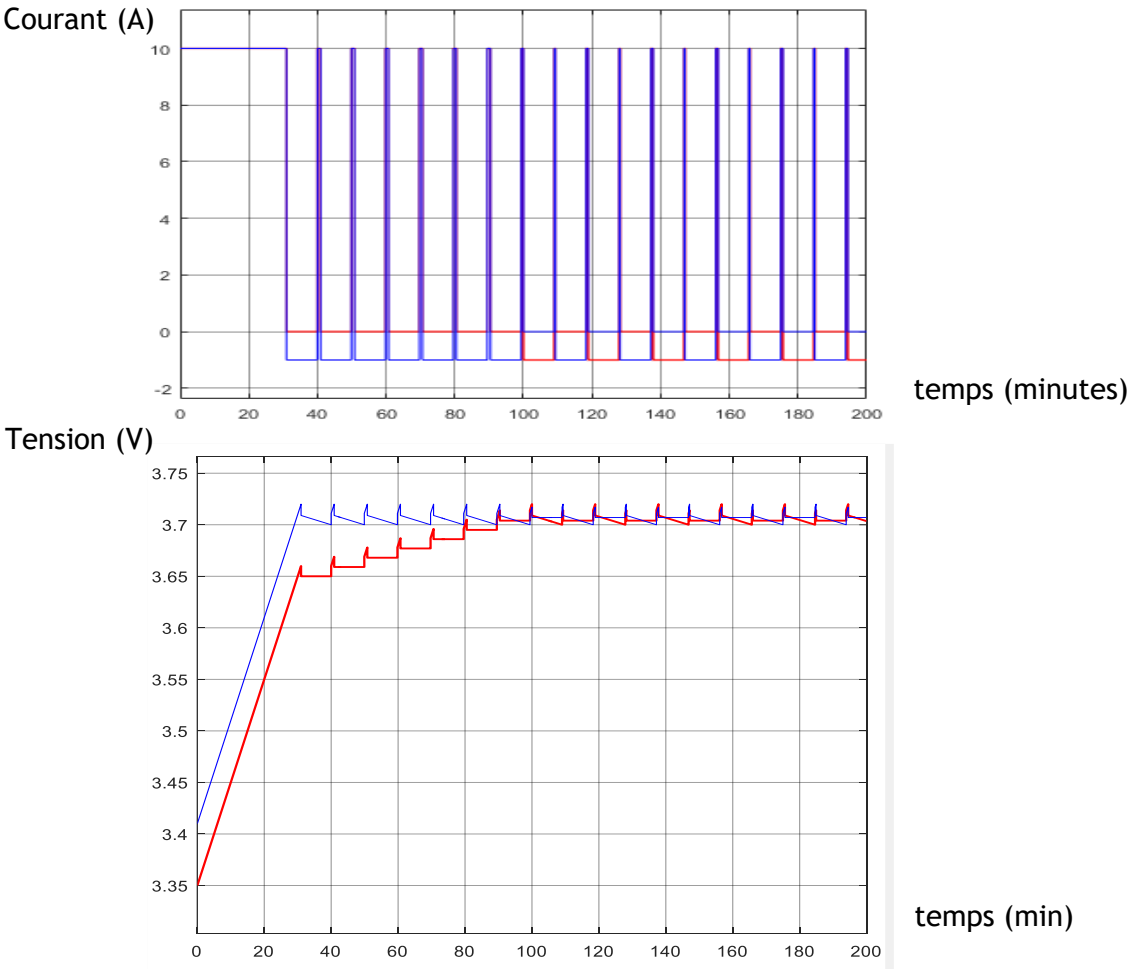
L'algorithme du BMS est relativement simple :
(x correspond à une des cellules séries de l'accumulateur)

- si UBx>3,7V il y a équilibrage, le courant dévié de la cellule est de 1A(= UBx/Resistance)
- si un des éléments est UBx>3,72V, arrêt de charge de l'élément
- si, toutes les cellules UBx≥ 3,7V, arrêt du chargeur

Sur la figure 3, la charge et les commutations du courant se font avec tous les éléments de même résistance interne (1 mΩ), mais un déséquilibre de 1A.h est à noter entre les éléments 1 et 2 (courbe de la capacité). La périodicité des commutations du courant est fonction de la pente de l'équation 4 et donc de la valeur de la consigne du courant ainsi que la valeur du courant de déviation et de l'hystérésis choisie sur la tension.

Sur l'évolution des capacités des deux éléments de la figure 3, le rattrapage du déséquilibrage de 1.A.h est atteint en 1 heure (équation (6)) avec un courant moyen de décharge de 1A.

$$\Delta Q(A.h) = I_{moy} \cdot temps(h) \quad (6)$$



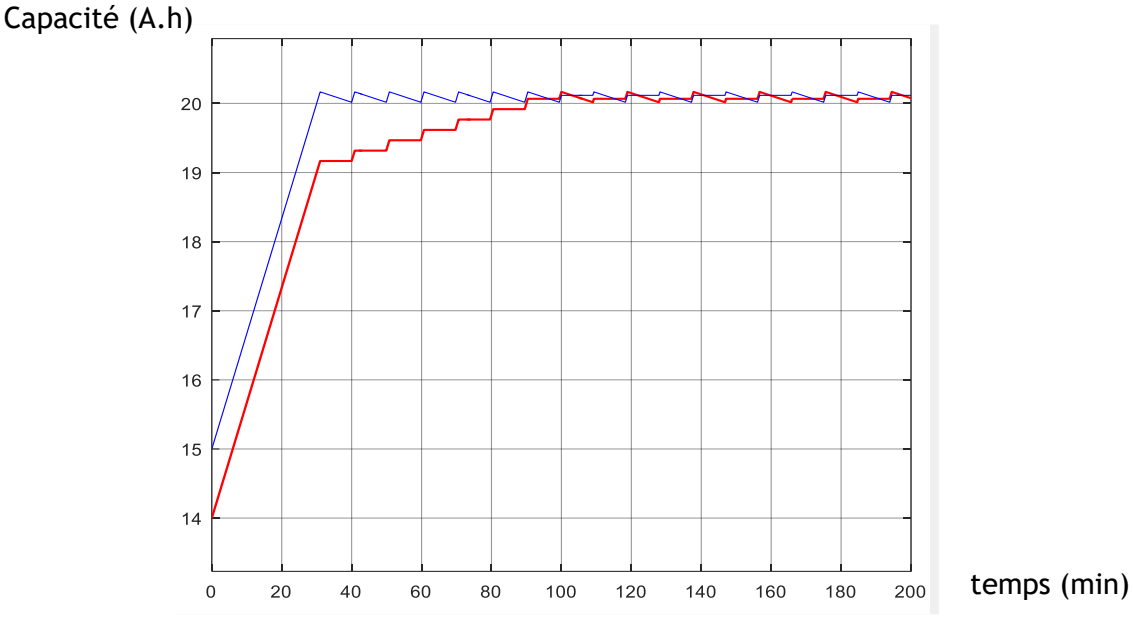


Figure 3 : Intensité, tension et capacité énergétique en fonction du temps de 2 éléments (rouge et bleu) d'une batterie avec un déséquilibrage de 1 A.h.

La perte de puissance due à l'équilibrage est donnée par l'équation (7) :

$$P_{perdue}(W) = (nb_{elements} - 1) \cdot E_{B\,seuil} \cdot I_{dévié} = 3,7 \cdot 1A = 3,7W \quad (7)$$

L'énergie perdue, quant-à-elle, dépend de la somme des équilibrages de chaque cellule :

$$W(W.h) = \sum E_B(V).\Delta Q(A.h) \quad (8)$$

Avec des éléments lipofer, le phénomène de relaxation de la tension entraîne une diminution de la tension à 3,45V en une heure et 3,42V au bout de 24h. Par conséquent, sans mémorisation de l'arrêt de charge par le BMS, la charge est à nouveau active inutilement.

En général, l'autodécharge des éléments est inférieure à 1% par mois mais pour des cellules âgées, l'autodécharge peut atteindre une dizaine de pourcents par mois. La « mémoire » de fin d'équilibrage bloque la charge due à l'autodécharge mais si la tension d'un élément est inférieure à une certaine valeur (par exemple 3,35V), la mémoire de fin de charge doit être remise à zéro.

Les résistances internes sont la cause de la présence de pics de tension de 10 mV (10 A et 1 mΩ) (figure 3) dont l'intensité reste négligeable par rapport à l'hystérésis choisie.

En revanche, si la résistance interne augmente (par exemple à 4 mΩ), on observe que les pics de tension sur l'élément 2 sont relativement importants (figure 4). Cette simulation est obtenue pour un même déséquilibre de 1 A.h entre les 2 éléments et la résistance du connecteur du chargeur est choisie arbitrairement à 6 mΩ (valeur nominale d'un connecteur Anderson de 5mm).

La durée de l'équilibrage est légèrement plus longue que dans le cas précèdent sans être trop pénalisante. En effet, comme le courant diminue lorsque la tension des 2 éléments est proche de la tension de seuil, les pics de tension s'amenuisent et l'équilibrage s'effectue correctement mais plus lentement.

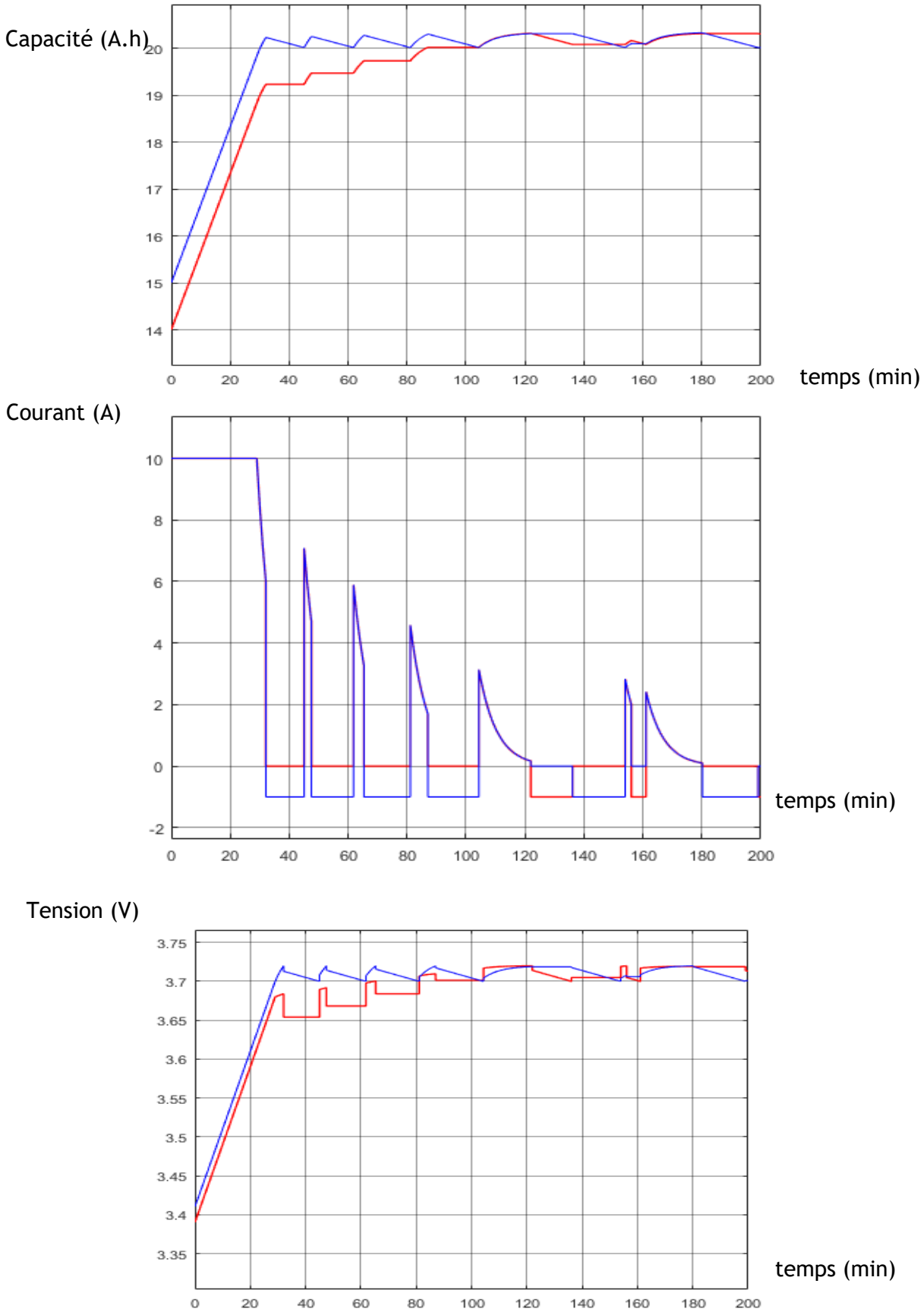


Figure 4 : Intensité, tension et capacité énergétique élément 1 et 2 avec un déséquilibrage de 1A.h, une résistance interne de 4mΩ sur l'élément 2 et une résistance RC (Résistance de Connexion) de 6 mΩ.

La simulation dont les résultats sont présentés sur la figure 5, correspond au même cas de figure que précédemment mais avec une résistance de connexion choisie arbitrairement à 44 mΩ. Dans ce cas, la durée de charge à courant de charge constant, correspond à 81% du SoC (équation 3).

Par conséquent, la durée de charge et l'équilibrage sont bien plus longs.

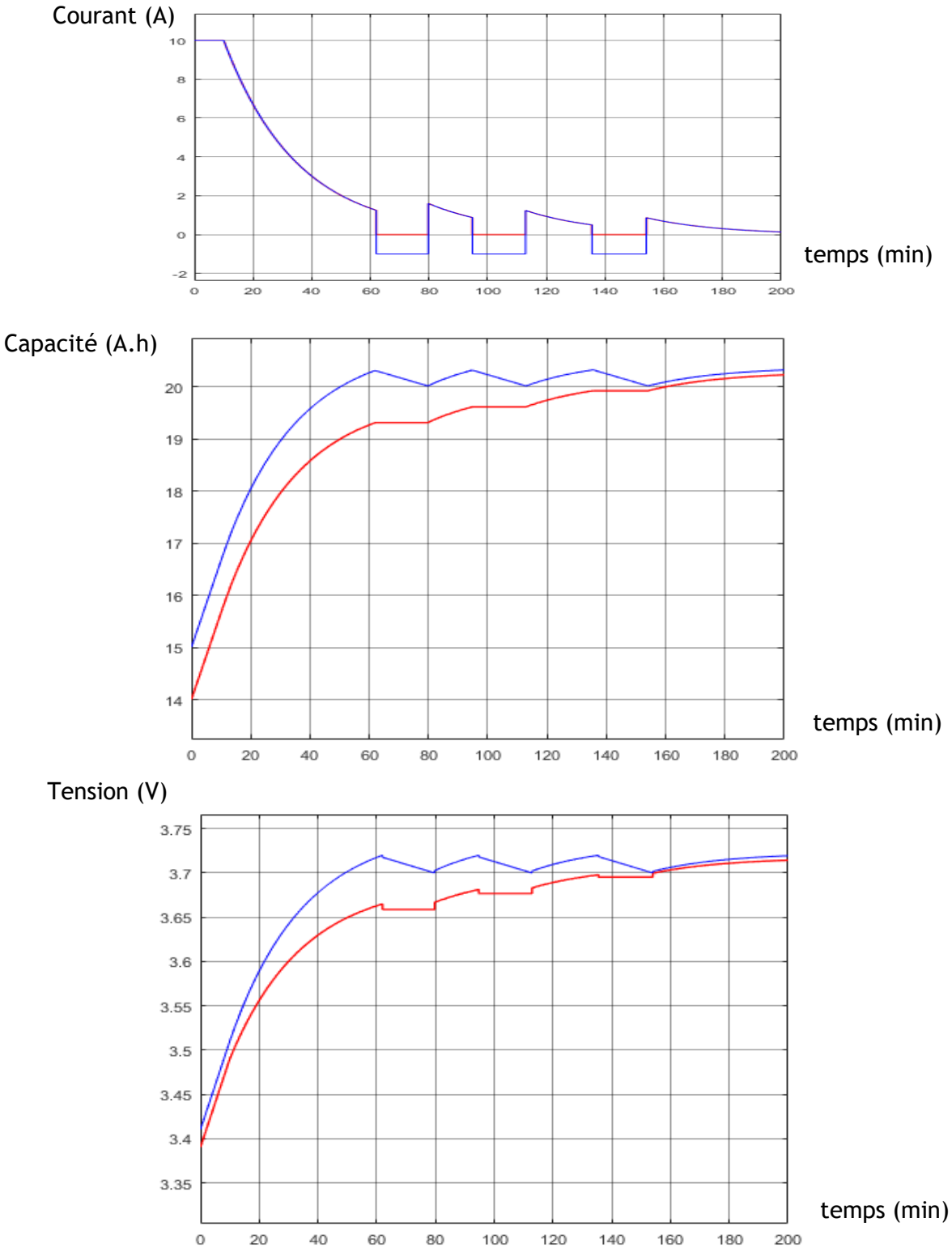


Figure 5 : Intensité, tension et capacité énergétique élément 1 et 2 avec un déséquilibrage de 1A.h, une résistance interne de 4mΩ sur l'élément 2 et une résistance de connexion RC de 44 mΩ.

Pour minimiser l'impact de la résistance de connexion et réduire la durée du mode à tension constante, les chargeurs puissants ont une tension d'alimentation bien supérieure à la somme des tensions de seuils des cellules. Dans ce cas, un réglage précis de la PWM est nécessaire pour obtenir la consigne de courant souhaitée.

Sur la figure 6, la consigne de courant est de 18 A, la période est relativement faible avec l'hystérésis choisie arbitrairement à 0,02 V.

On peut noter que pour des courants de consigne un peu plus élevés, alors les commutations sont bien plus nombreuses et la période d'échantillonnage du courant d'un BMS numérique (microcontrôleur) peut être dépassée alors que les BMS utilisant des circuits analogiques n'ont pas ce problème d'échantillonnage.

Avec cette consigne de courant de 18A, le temps de charge est relativement rapide mais le temps d'équilibrage reste inchangé.

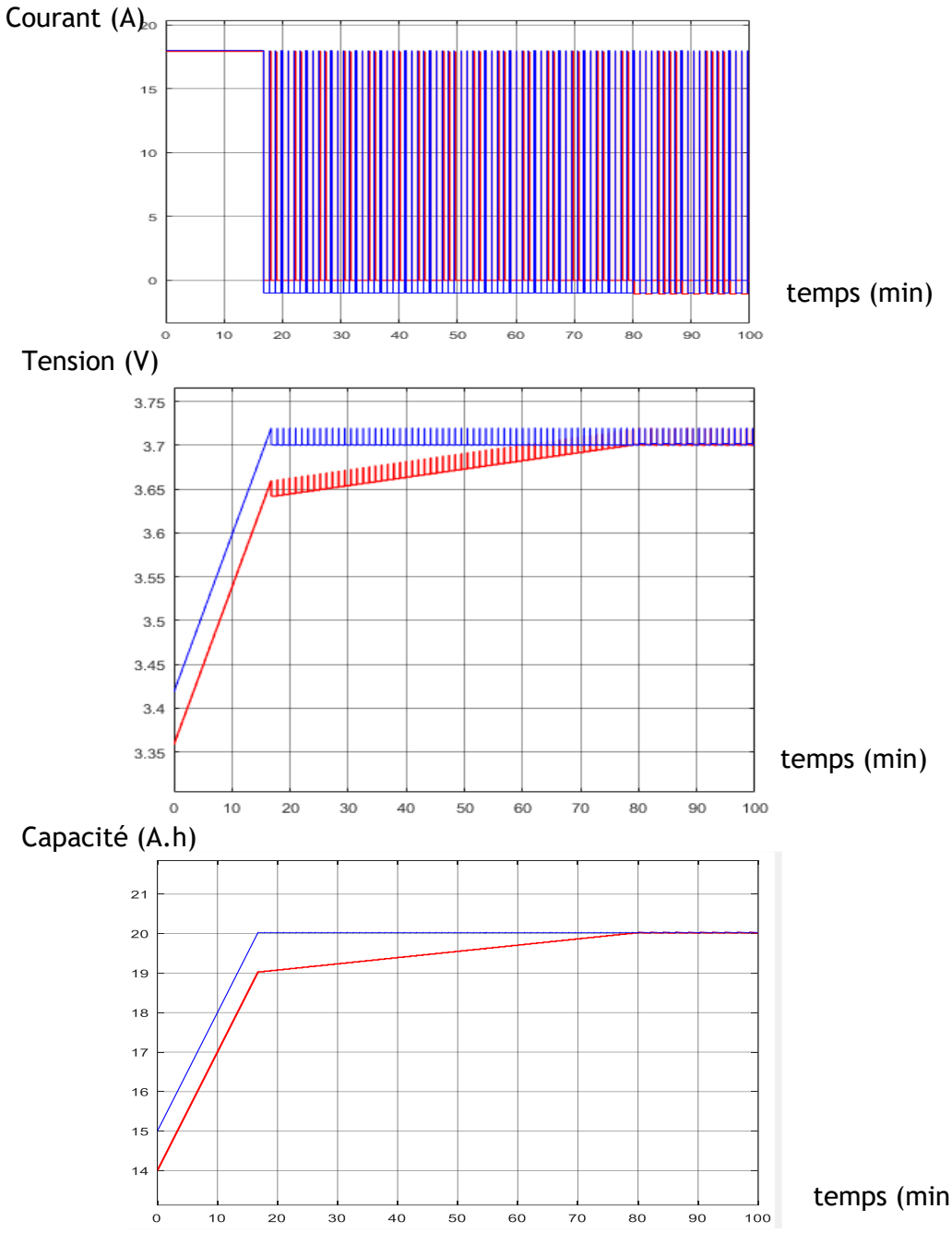


Figure 6 : Intensité, tension et capacité énergétique élément 1 et 2 avec un déséquilibrage de 1 A.h et une résistance des différents éléments de 1 mΩ avec courant de 18A.

Pour minimiser le nombre important de commutations lors de l'équilibrage, on fixe la consigne du courant du chargeur pour qu'elle soit identique à la valeur du courant dévié de l'équilibrage. Il est alors nécessaire que le BMS puisse indiquer l'état de l'équilibrage au chargeur. De plus, pour un

faible courant de charge, la résistance interne de chaque élément a moins d'influence sur la fréquence de détection de la tension max synonyme de l'arrêt de la charge.

Notons qu'à chaque charge, il n'est pas obligatoire de faire un équilibrage de la capacité énergétique car celle-ci est en général bien inférieure à 1%. Mais le déséquilibre de la capacité énergétique de l'élément de plus grande résistance interne s'accroît et l'autonomie du système s'en trouve fortement dégradée car la coupure de la batterie par le BMS est basée sur la capacité énergétique de l'élément le plus faible. Remarquons que bien qu'un équilibrage tous les 30 cycles est en général un bon compromis, les BMS commercialisés ne permettent pas à l'utilisateur de faire ce choix.

A ce jour, les BMS passifs proposent des équilibrages pour des courants très faibles (en général inférieur à 0,05A même pour des batteries de 20A.h). L'équilibrage peut donc être très long si le déséquilibrage est profond [9].

Pour éviter la déperdition thermique due au BMS passif, il est possible de charger l'accumulateur élément par élément. C'est une des solutions d'équilibrage rapide réalisée avec un chargeur isolé par cellule. Dans ce cas, le faisceau de fils de chaque cellule doit posséder une section correcte et la connectique doit être adaptée. Pour charger à 10A, une section de 1,5 mm^2 est acceptable, pour les intensités plus grandes, cela devient plus difficile.

On notera qu'il existe de l'équilibrage actif avec un seul chargeur associé à un multiplexage permettant ainsi une alimentation élément par élément [7]. Les autres solutions d'équilibrage actif comme le transfert de charge par capacité tel que le LTC3305 via des capacités de 1mF n'est pas encore à ce jour très démocratisé chez les fabricants de BMS [8]. En revanche, l'équilibrage actif inductif avec le circuit ETA3000 pour des courants de 0.2A à 1A commence à être fréquemment distribué [12].

La figure 7 donne un exemple de BMS passif avec équilibrage à 0,03A et actif à 1A en option.

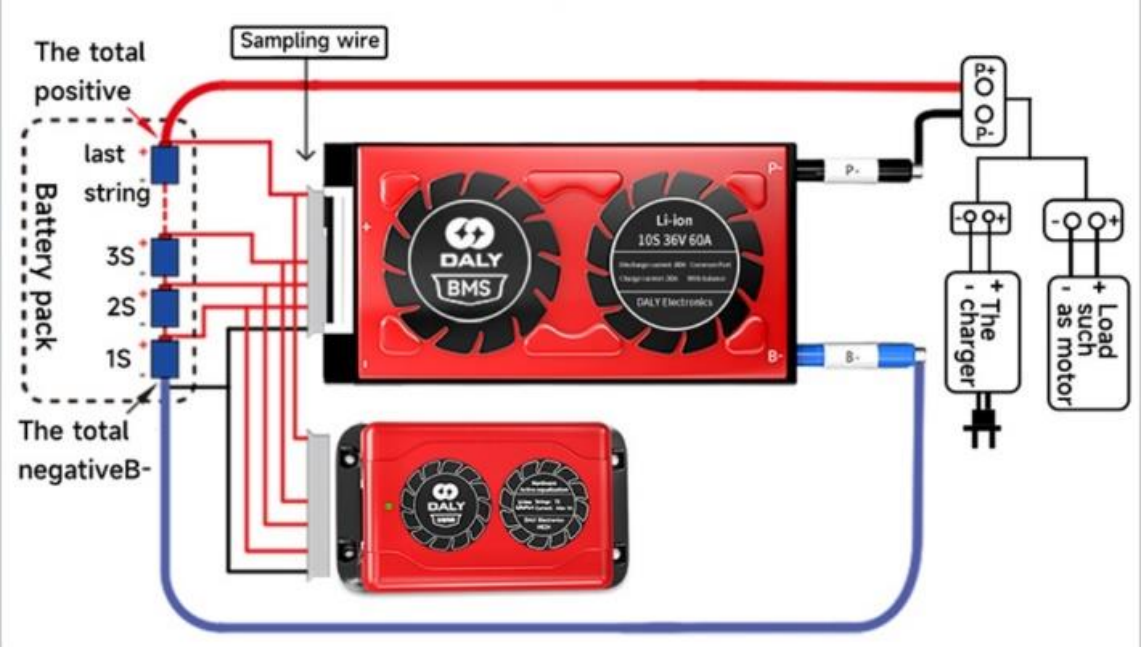


Figure 7 : BMS passif et actif [10]

Pour les BMS numériques, les « firmware updates » sont rarement « open source ». Les modifications du paramétrage des BMS et des méthodes de mesures et d'estimation de modèle sont alors difficiles. Par ailleurs, la production à grande échelle des BMS engendre des prix bas et

explique d'une part que les BMS sont rarement réparés et d'autre part que leurs schémas électriques sont rarement disponibles.

En revanche les BMS « open source » réalisés par des passionnés sont facilement réparables et reprogrammables [11].

3 - Efficacité et norme d'un chargeur

La réglementation UE fixe des limites à la pollution harmonique afin d'améliorer le facteur de puissance (EN61000-3-2 (basée sur la norme CEI 61000-3-2) des chargeurs. Ainsi, toutes les alimentations à découpage dont la puissance de sortie est supérieure à 75 Watt doivent inclure une correction du facteur de puissance.

En pratique, c'est rarement le cas, même pour des puissances de 1500W (72V, 20A) dont le facteur de puissance est d'environ de 0,6.

En revanche, avec un rendement d'environ 90% en charge et des pertes en fonctionnement à vide égales à 3% de la puissance maximale les chargeurs ne nécessitent en général pas de système de mise en veille automatique propre. Toutefois, le nombre de chargeurs par foyer augmentant, l'option de la veille reste intéressante pour réaliser des économies d'énergie.

4 - Durée de charge

La durée de charge dépend de la chimie, du courant de charge et de la déperdition thermique.

Les batteries lithiums Lipofer, Lipo et Li-ion, peuvent se charger à 1C jusqu'à 80% (donc en 0,8 h avec un SoC de départ de 0%) pour une température ambiante de 40°C. Les 20% de charge restantes peuvent durer assez longtemps (environ 1h) selon la nécessité d'équilibrage des cellules.

La durée de charge inférieure à 1C est donnée par les relations (9, 10) suivantes en fonction de la puissance ou du courant.

$$\text{durée}_{\text{charge}} \approx \frac{(\text{SoC}_{\text{final}}(\%) - \text{SoC}_{\text{initiale}}) \cdot \text{Q}_{\text{nominal}}(\text{A.h}) \cdot \text{V}_{\text{batterie}}}{\text{Puissance}_{\text{charge}}(\text{W})} \quad (9)$$

$$\text{durée}_{\text{charge}} \approx \frac{(\text{SoC}_{\text{final}}(\%) - \text{SoC}_{\text{initiale}}) \cdot \text{Q}_{\text{nominal}}(\text{A.h})}{\text{I}_{\text{charge}}(\text{A})} \quad (10)$$

Pour la batterie précédente et pour des températures ambiantes de plus de 15°C, les pertes thermiques avec un courant de 20A provoquent une augmentation de la température de 5°C. En comparaison, à 0°C l'augmentation de la température est de 13°C car la résistance interne de la batterie est plus grande. Une température ambiante de 40°C entraîne l'arrêt de la charge par le BMS.

L'évolution du courant de charge jusqu'au basculement en mode CV est donnée par la relation 11 dans laquelle Q_{cv} représente la capacité énergétique du passage en mode CV et Q_f la capacité finale ou nominale de la batterie.

$$\text{I}_{(\text{Q})} = \frac{\text{I}_{\text{limit}} \cdot (\text{Q} - \text{Q}_{\text{f}})}{(\text{Q}_{\text{cv}} - \text{Q}_{\text{f}})} \quad (11)$$

En mode CV, la capacité énergique de la batterie ne varie plus proportionnellement au temps (comme dans le mode CC) mais de façon exponentielle :

$$Q(t) = Q_f + (Q_{cv} - Q_f)e^{-\frac{t \cdot I_{limit}}{Q_f - Q_{cv}}} \quad (12)$$

Théoriquement, il faudrait une durée infinie pour recharger à 100% mais la charge peut être considérée comme achevée lorsque la capacité atteint 95% de la valeur finale.

Finalement, la durée de charge totale à partir d'une capacité initiale est donnée par l'équation suivante :

$$temps_{charge}(h) = \frac{Q_{cv} - Q_i}{I_{limit}} + \frac{Q_{cv} - Q_f}{I_{limit}} \ln \frac{(0.95-1)Q_f}{Q_{cv} - Q_f} \quad (13)$$

La charge avec la limitation du courant et prise en compte de la valeur du SoC_{CV} peut être simulée à l'aide du modèle Simulink ci-dessous :

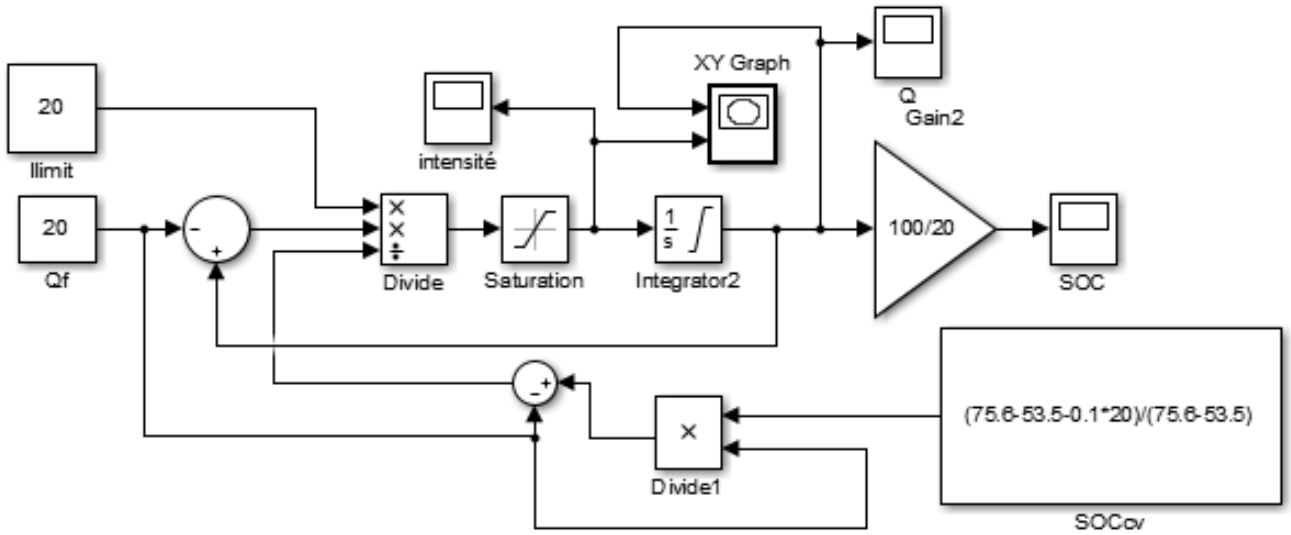


Figure 8 : Schéma de simulation de la charge d'une batterie de 75,6V en fonction du temps avec le mode CV et CC (sous Simulink).

Pour une charge de batterie à 20 A.h avec une résistance équivalente de 0,1 Ω pour laquelle le SoCcv s'établit à 91%, la durée de charge est très faiblement impactée. Les évolutions temporelles du courant et de la capacité énergétique sont représentées sur les 2 figures suivantes pour 2 valeurs de limitations de courant (20A et 40A) à partir d'une capacité énergétique restante de 1A.h.

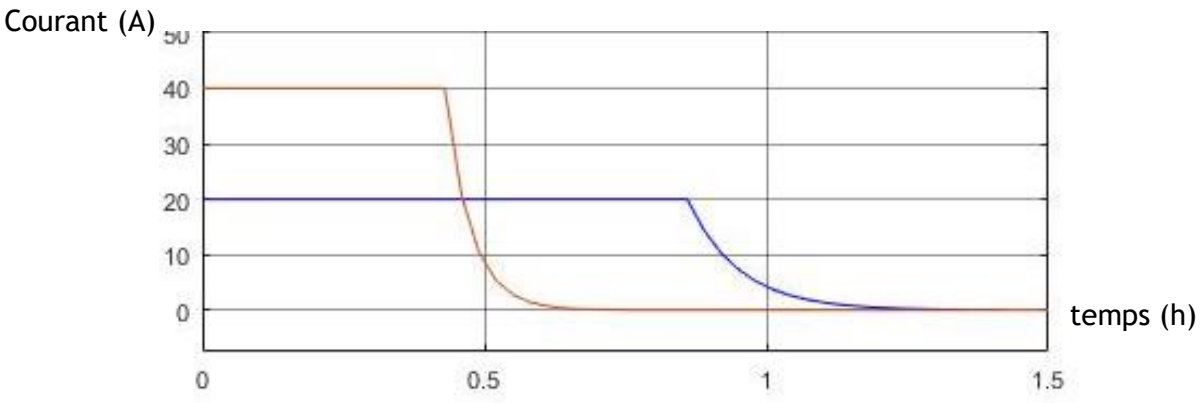


Figure 9 : Évolution du courant de charge en fonction du temps

Pour un courant de charge supérieur à 1C, le « courant limité » va dépendre des caractéristiques thermiques de l'accumulateur (inertie, refroidissement, variation de température). La durée de charge n'est donc plus donnée par les relations (9,10).

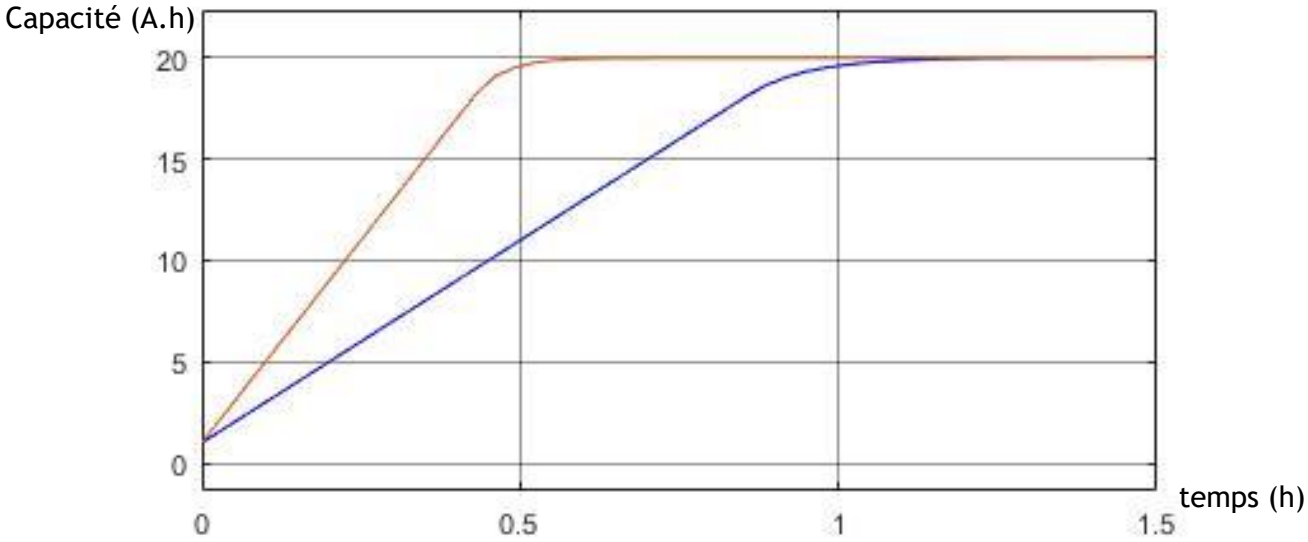


Figure 10 : Capacité énergétique en fonction du temps, résistance interne de 4mΩ.

5 - Paramétrage BMS

La capacité énergétique des batteries diminue fortement si la batterie est stockée avec une charge de 100% et lorsque sa température dépasse 35°C (effet calendaire). En effet, de nombreux essais indiquent qu'il y a une dégradation bien plus importante lorsque la batterie est stockée à 90% de charge par rapport à 60% ou à 30% et surtout pour des températures supérieures à 35°C quelque que soit la chimie. La figure 11 permet de mettre en évidence l'effet calendaire sur l'énergie stockée de la batterie qui diminue avec le temps même si elle n'est pas utilisée [13,14].

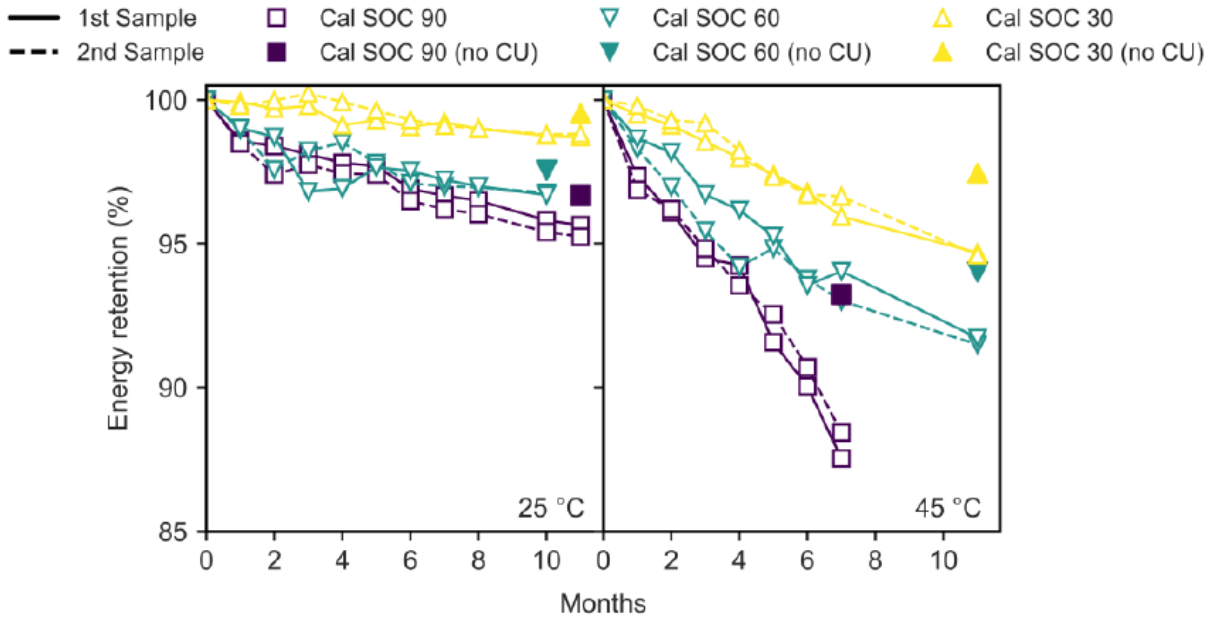


Figure 11 : Énergie de la batterie en fonction du SoC de stockage et de la température [13]

Si la capacité énergétique n'est pas limitante, il est possible de diminuer la tension du chargeur pour obtenir une utilisation de 80% de la capacité maximale avec équilibrage. La tension de seuil des éléments du BMS doivent alors être changée. Seul les BMS numériques permettent éventuellement ce réglage si toutefois une communication chargeur – BMS est réalisée.

A titre d'exemples, les chargeurs de modélisme avec équilibrage interne permettent de recharger à la capacité désirée. De plus, l'équilibrage passif peut se faire avec des courants de 0,5 A grâce à la ventilation forcée et déportée.

6 - Conclusion

Dans cet article, nous avons mis l'accent sur des pistes d'évolution pour les BMS, que ce soit sur les aspects réglages ou communication avec le chargeur.

Par ailleurs la communication par Bluetooth via un smartphone ou un PC permet, tout en évitant un écran sur le chargeur, de réaliser un diagnostic de la batterie (mesure de résistances internes de tous les éléments, prise en compte de la température, ou paramétrage du BMS en fonction d'une utilisation désirée, …).

Nous avons aussi vu que l'utilisateur peut optimiser la durée de charge, lisser, les pics de puissance et régler la capacité énergétique afin d'augmenter la durée de vie de l'accumulateur.

7 - Bibliographie

[1] A.Sivert, F.Betin, B.Vacossin, A.Yazidi , H.Caron «Stratégie de choix entre différentes technologies de batteries Lithium (Durée de vie, fiabilité…)» Symposion génie électrique, Juillet 2018,

[2] A.Sivert, B. Vacossin, F. Betin, « Etude technique comparative de différents véhicules électriques de 50 kg à 1500 kg , Vitesse, consommation, batteries, prix, modes de charge » revue R3EI nov 2021, N°106
http://actes.sge-conf.fr/2018/articles/article_184697.pdf

[3] A.Sivert, F.Betin, B.Vacossin , H.Caron, « Etat de Santé de batterie lithium par Régression Linéaire et par Incremental Capacity » SGE 2021 juillet

[4] «A Novel Electric Bicycle Battery Monitoring System Based on Android Client »
https://www.researchgate.net/publication/318998340_A_Novel_Electric_Bicycle_Battery_Monitoring_System_Based_on_Android_Client

[5]«Optimized charging of power battery» Advanced Energy Storage and Application
http://en.aesa.net.cn/About.aspx?ClassID=125

[6] Intersil ISL94203 datasheet
https://manualzz.com/doc/30396782/isl94203-datasheet
https://pdf1.alldatasheet.fr/datasheet-pdf/view/1052355/RENESAS/ISL94203.html

[7] BQ76PL455A « 16-Cell Li-Ion Battery Active Balance capacitor»
https://www.ti.com/tool/EM1402EVM

[8] Techniques balancing battery
https://circuitdigest.com/article/cell-balancing-techniques-and-how-to-use-them

[9]« Basic principles of automotive modular battery management system design » IOP Conference 2020
https://www.researchgate.net/publication/341765471_Basic_principles_of_automotive_modular_battery_management_system_design

[10] BMS balancing active
https://www.dalybms.com/the-optional-description-of-daly-active-equilibrium-bms-2-product/

[11] BMS open source Arduino
https://forum.arduino.cc/t/bms-etat-de-charge-et-sante-de-batterie-lithium-banc-cyclage-arduino/607106/50?page=3
https://github.com/Eman2000/Arduino-BMS

[12] Test de PCB d'équilibrage actif distribué
http://velorizontal.1fr1.net/t26614p275-eclairage-a-del-pour-velo-led-light-for-bike-partie-2

[13]Thèse, Marc Haber, 2023 «Improving the lifetime prediction methodology of Li-ion batteries for electric vehicles»

[14] Thèse, Eduardo Redondo Iglesias « Étude du vieillissement des batteries lithium-ion dans les applications "véhicule électrique" : combinaison des effets de vieillissement calendaire et de cyclage »

En 2024, ✔ j'adhère à la SEE

La SEE, société savante française fondée en 1883, forte de 2 000 membres, couvre les secteurs de l'électricité, de l'électronique et des technologies de l'information et de la communication.

BULLETIN À COMPLÉTER ET RENVOYER À :

SEE - Service adhésions - 17 rue de l'Amiral Hamelin
75116 Paris - France.
Tél : +33(0)1 56 90 37 17 - adhesion@see.asso.fr

❑ J'adhère à la SEE

- ❑ Standard ..... 130 €
- ❑ Retraité ..... 70 €
- ❑ Enseignant ..... 70 €
- ❑ Jeune actif < 35 ans ..... 70 €
- ❑ Etudiant ..... 15 €
- ❑ Recherche d'emploi ..... 15 €

** Une remise de 10% est accordée aux membres IEEE*
Merci d'indiquer votre n° de membre IEEE : .....

La SEE change son mode d'adhésion et passe à une adhésion d'un an, date à date à partir de la date de paiement. Le 1er numéro servi pour la REE sera le numéro suivant la date de paiement. Plus d'informations sur l'abonnement sur le site web de la SEE.

❑ Je m'abonne à la revue REE de la SEE à un tarif préférentiel !**

REE - La Revue de l'Électricité et de l'Électronique

4 numéros par an (Mars, Mai, Octobre Décembre)

- ❑ Livraison France : **68 € TTC**
- ❑ Livraison UE : **78 € TTC** (76,40 € HT*)
- ❑ Livraison Hors UE : **83 € TTC** (81,70 € HT*)

La Revue 3EI

4 numéros par an

Retrouvez la revue 3EI en accès gratuit sur le site web de la SEE

** Prix HT valide si le pays de facturation est hors UE, ou si la TVA Intracommunautaire est fournie pour un pays de l'UE. **Prix réservés aux adhérents - Abonnement dans la limite des stocks disponibles.*

Mes coordonnées / Adresse de livraison

❑ Mr ❑ Mme Prénom* : ..... Nom* : .....

Adresse* : .....

Code postal* : ..... Ville* : ..... Pays* : .....

Téléphone* : ..... email** : .....

*(*Obligatoire)*

***En adhérant à la SEE, votre mail est le moyen de contact et d'information de l'association & de ses activités (congrès, soirées débats, revues, etc.). Vous acceptez donc de recevoir les diffusions de l'association.*

Adresse de facturation (si différente)

Raison sociale de l'employeur : ..... Service : ..... Activité : ..... *(Facultatif)*

Adresse : .....

Code postal : ..... Ville : ..... Pays : .....

email* : .....

*(*Obligatoire)*

BON DE COMMANDE :

Mon règlement

❑ Virement bancaire : **BNP Paribas, Paris Associations (02837) IBAN - FR76 3000 4002 7400 0103 3624 258**

❑ Chèque bancaire ou postal **à l'ordre de la SEE**

❑ Carte bancaire Visa / Euro / Mastercard

N° |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Date d'expiration |_|_|_|_| Cryptogramme |_|_|_|

Signature obligatoire

+ RAPIDE :

Adhérez dès aujourd'hui via le site internet de la SEE sur : www.see.asso.fr

En tant que membre de la SEE, ✔ vous bénéficiez de nombreux avantages

see since 1883

1 Élargir son réseau professionnel par la participation à des structures de réflexion adaptées

- **6 Clubs techniques**
  - Cybersécurité et Réseaux Intelligents (CRI)
  - Capteurs et Systèmes ElectroMagnétiques (CSEM)
  - Eco-conception en Génie Electrique (EGE)
  - Ingénierie des Systèmes d'Information et de Communication (ISIC)
  - Systèmes Electriques (SE)
  - Stockage et Moyens de Production (SMP)
- **6 Groupes régionaux**

2 Participer et bénéficier de tarifs préférentiels pour les Conférences et Journées d'études SEE

- Conférences nationales et internationales
- Journées d'études thématiques
- Conférences-débats
- Congrès internationaux, en partenariat ou non avec d'autres sociétés scientifiques

3 Consulter et télécharger gratuitement les publications (REE et 3EI) en version numérique et s'abonner aux publications papier à tarif préférentiel

- **La Revue de l'électricité et de l'électronique (REE)** est destinée aux ingénieurs, chercheurs, enseignants, décideurs techniques et économiques intéressés par les secteurs de l'électricité, de l'électronique, de l'information et de la communication. Paraissant cinq fois par an, la revue s'articule autour de dossiers techniques, flash-infos, articles invités, entretiens avec des personnalités du monde de la recherche et de l'industrie.
- **La revue 3EI** est une publication trimestrielle destinée aux professeurs, universitaires et industriels concernés par l'enseignement de l'électricité et de l'électronique industrielle.

4 S'engager dans une association qui reconnaît les talents et crée une émulation parmi ses membres

- Grades senior et émérite SEE
- Remise de prix et trophées pour les professionnels confirmés, jeunes actifs et étudiants (Brillouin-Glavieux, général Ferrié, André Blanc Lapierre…)
- Remise de médailles (Ampère, Blondel…)

Pour rejoindre la SEE, deux modes d'adhésion :

Adhésion individuelle :
via le site www.see.asso.fr
ou le bulletin d'adhésion

Adhésion collective :
partenariat d'entreprise ou d'école via des conventions

NOUS CONTACTER :
SEE - Service adhésions - 17 rue de l'amiral Hamelin - 75116 Paris - France
+33 (0)1 56 90 37 17 - adhesion@see.asso.fr - www.see.asso.fr

OUVERTURE DES CANDIDATURES EN AVRIL ET MAI 2024

GRANDS PRIX SEE 2024

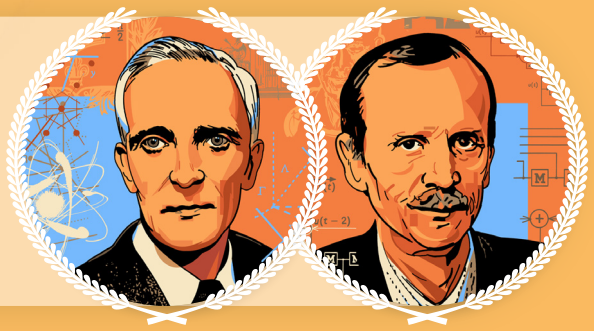
Comme chaque année, la SEE lance les appels à candidatures pour ces Prix et Médailles de grand renom, afin d'identifier les personnalités scientifiques et techniques

GRAND PRIX DE L'ÉLECTRONIQUE GÉNÉRAL FERRIÉ | 67ÈME ÉDITION

Depuis 1963, le Grand Prix de l'Électronique « Général Ferrié » est décerné annuellement.
Il récompense un ingénieur ou scientifique dont les travaux ont contribué d'une manière importante aux progrès des Systèmes d'Information et de Communication, y compris dans leurs aspects énergétiques.

PRIX BRILLOUIN – PRIX GLAVIEUX | 17ÈME ÉDITION

En 2024, le prix Brillouin ainsi que le Prix Glavieux seront tous les deux remis.
Les domaines concernés par le Prix Brillouin sont :
physique des matériaux et des composants, optique, électronique et électrotechnique.
Les domaines concernés par le Prix Glavieux sont :
sciences de l'information et des communications, automatique et commande des systèmes, traitement du signal et des images et les domaines connexes.



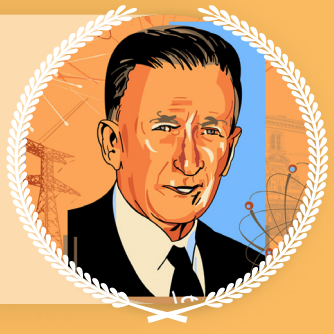
MÉDAILLE BLONDEL | 82ÈME ÉDITION

La médaille Blondel couronne chaque année des scientifiques universitaires ou industriels, français ou étrangers, pour des travaux remarquables contribuant aux progrès de la Science et des Industries Electrique et Electronique, et menés avec les mêmes soucis d'approfondissement et de rigueur que ceux qui caractérisaient les travaux d'André Blondel.

PRIX «JEUNES» ANDRÉ BLANC-LAPIERRE | 22ÈME ÉDITION

Les Prix «Jeunes» André Blanc-Lapierre ont été institués en mémoire de ce grand scientifique français, déclinés en Prix Régionaux puis en Prix National.

Ces distinctions s'adressent aux étudiants au niveau master ingénieur, des établissements français d'enseignement supérieur dans les domaines de l'électricité, de l'électronique, des technologies de l'information et de la communication, de se distinguer en bénéficiant d'une reconnaissance nationale.



POUR TOUT RENSEIGNEMENT :
www.see.asso.fr