INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Université de Toulouse

INSTITUT NATIONAL DES SCIENCES APPLIQUEES

TOULOUSE

DEPARTEMENT DE GENIE ELECTRIQUE & INFORMATIQUE

PROJET DE FIN D'ETUDES

Spécialité : 5AE

Filière : ESPE

Stage ingénieur développement en électronique analogique

Auteur :
Pannetier Marvyn

Entreprise :
Thales

Référent INSA :
Escriba Christophe

Responsable du stage :
Tremblier Mégane

Année 2021-2022

RESUME

Ce rapport a pour but de présenter et décrire le déroulement de mon stage de fin d'études. Celui-ci est divisé en plusieurs sections afin de mieux appréhender les différentes étapes du stage. Dans un premier temps, une description du sujet est faite afin de cerner ces enjeux. Ensuite, une partie est dédiée au planning initial et son évolution au cours du stage en fonction des obstacles rencontrés ou non. La partie suivante, qui est la plus imposante concerne la réalisation du projet, celle-ci est aussi divisée en plusieurs sous-parties décrivant la théorie, la simulation et les tests sur le PCB. Enfin une partie sur les perspectives pour la fin du stage suivie d'une conclusion viendront clôturer ce rapport. Après avoir lu ce rapport, le sujet et ses enjeux, les compétences que j'ai développées et mises à disposition, mes réalisations, mes conclusions sur l'aspect technique comme humain, seront clairs et connus du lecteur. Bonne lecture !

REMERCIEMENTS

Je tiens tout d'abord à remercier l'Institut National des Sciences Appliquées (INSA) pour m'avoir fait confiance. L'écoute et l'accompagnement dont j'ai bénéficié m'ont permis de trouver rapidement un stage, dans le but d'affiner mon projet professionnel et d'obtenir le diplôme auquel j'aspire depuis plusieurs années.

Je remercie également Thales, qui a cru en mon potentiel et m'a accueilli au sein de ses équipes.

À ce titre, je souhaiterais remercier tout particulièrement François ROBERT, Mégane TREMBLIER ainsi que Elina FIAWOO qui m'ont épaulé, conseillé et m'ont transmis leur expertise dans le domaine de l'électronique analogique.

Ce stage m'a permis d'affiner certaines pistes afin de bâtir mon projet d'orientation professionnel et signe l'aboutissement de mes études d'ingénieur.

Table des matières

GLOSSAIRE

ACS : Alternating current signal

xVDT: Linear/Rotary Variable Differential Transformer

SPWM : Sinusoidal Pulse Width Modulation

FCC : Flight Control Computer

COTS : Component Off-The-Shelf

ESR : Equivalent Series Resistance

CEM : Compatibilité électromagnétique

DMA : Direct memory access

THD : Total harmonic distortion

ELAC : Elevator Aileron Computer

CHAPITRE 1. INTRODUCTION

L'électronique analogique est un domaine vaste et très présent dans notre vie quotidienne. Le monde de l'aéronautique, étant le secteur d'activité principal de Thales, n'échappe pas à l'électronique analogique. En effet, même si la tendance est de plus en plus à l'utilisation du numérique, l'analogique reste néanmoins très présent et indispensable au bon fonctionnement des systèmes développés chez Thales.

Ce stage se déroule au sein de l'entité DMS de Thales, une entité orientée militaire, et a pour sujet le développement d'un circuit d'électronique analogique. Pour être plus précis, le but de ce stage est de reprendre un circuit existant et utilisé actuellement, afin de concevoir, tester et valider un nouveau circuit analogique réalisant la même fonction tout en optimisant le coût, la surface, la consommation ou encore le rendement. L'aéronautique est un domaine où les circuits sont développés pour être fonctionnels plusieurs dizaines d'années, il est donc courant de trouver des cartes électroniques développées avant les années 2000 par exemple. Cependant, les technologies ont grandement évoluées ces dernières années, ce qui rend ces cartes beaucoup moins performantes sur les points cités précédemment, comparé à ce qui est développé aujourd'hui. De plus, écologiquement parlant il est important de réduire les consommations en optimisant tous ces paramètres sur les produits électroniques développés, ce qui donne à ce stage un enjeu plus fort je trouve.

Au début de mon stage, deux sujets m'ont été proposés. Ils vont donc tous deux être présentés dans la suite de ce rapport, et mon choix sera justifié ensuite. Le planning sera ensuite exposé et commenté, avant de s'attaquer à la partie réalisation du projet qui expliquera chaque étape de la théorie jusqu'au test de la carte électronique. Enfin, quelques paragraphes viendront conclure ce rapport.

CHAPITRE 2. CADRE ET OBJECTIFS DU STAGE

Les sujets qui vont être présentés ci-dessous ne font pas parties d'un projet particulier mené par THALES, mais s'apparentent plus à de la recherche technologique. En effet, la mission qui m'est confiée consiste à concevoir de nouveaux circuits électroniques pour remplacer les circuits utilisés actuellement et qui ont été conçus il y a plusieurs années. Le but étant d'utiliser des technologies plus récentes afin de diminuer la surface, le coût, l'efficacité…etc.

1.1. Description des sujets

1.1.1. Sujet 1: Générateur de signaux alternatifs à base d'amplificateur de classe D

La fonction appelée ACS est présente dans de nombreuses commandes de vols (FCC), car elle permet de stimuler les capteurs xVDT. Ces capteurs, représentés sur la figure 1, permettent de mesurer un angle ou un mouvement en excitant une bobine primaire et en regardant les signaux des bobines secondaires.

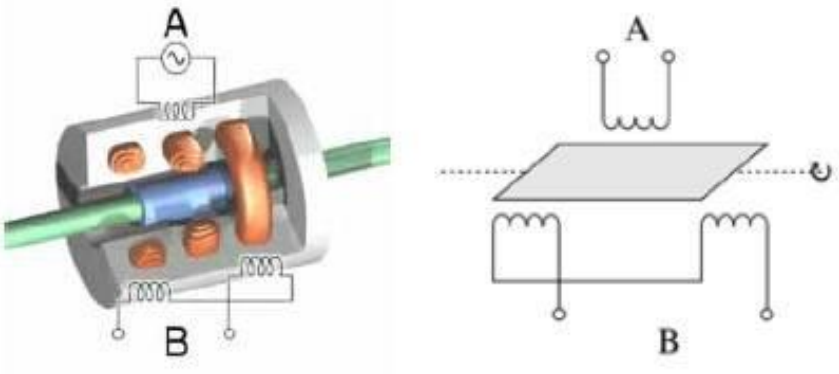


Figure 1: Exemple de capteur xVDT ici un type LVDT

Le but de ce sujet est de développer une nouvelle fonction analogique ACS, afin d'améliorer la fonction actuelle. En effet, la solution actuelle utilise un amplificateur opérationnel linéaire et a par conséquent une consommation et un rendement non optimaux. Le rendement avoisinant les 10% dans les meilleurs cas. L'idée est donc de développer un nouveau circuit basé sur un amplificateur de classe D à la place d'un amplificateur opérationnel linéaire, qui pourrait théoriquement avoir une meilleure efficacité. Cependant, certains points pourraient être problématiques, par exemple, les ondulations sur le signal de sortie dues aux commutations qui peuvent engendrer des problèmes de CEM (EMI) et rendre ce circuit inutilisable car il ne respecte pas les gabarits d'émissions conduites imposés par les normes.

L'amplificateur de classe D sera une partie importante de ce projet. La figure 2, donne une vue globale de son fonctionnement, mais une étude plus spécifique sera faite dans le rapport.

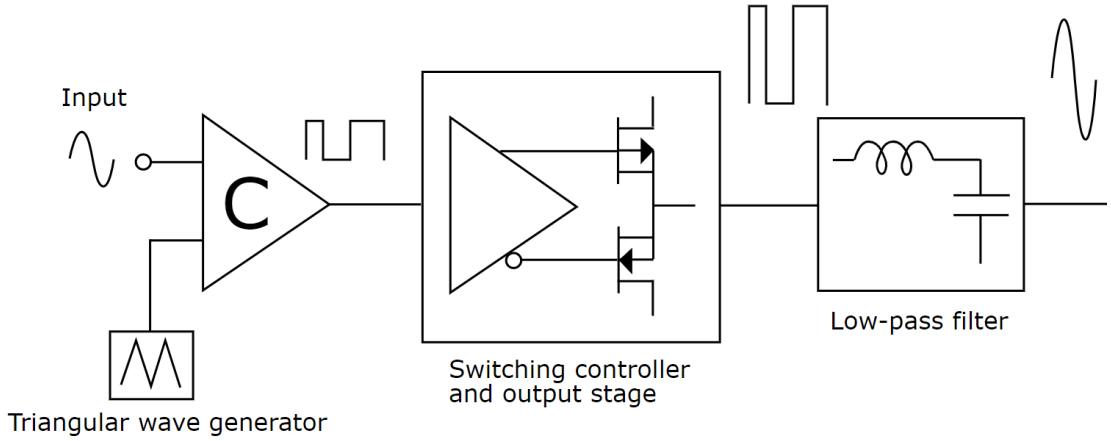


Figure 2: Schéma simplifié d'un amplificateur de classe D

À partir de cette technologie, le but serait de concevoir un circuit générant une sinusoïde ayant une tension RMS imposée par le cahier des charges. De plus les capteurs de type xVDT sont excités par des tensions différentielles, c'est-à-dire que deux sinusoïdes de phases opposées sont appliquées aux bornes des capteurs. Il faut donc concevoir un circuit avec une sortie différentielle. Enfin une partie de ce sujet est axé sur l'asservissement de la tension différentielle aux bornes du capteur, il faut donc concevoir un correcteur PI ou PID afin d'asservir cette tension correctement.

1.1.2. Sujet 2: Solution de substitution à des DAC à l'aide de PWM

Le but de ce sujet est de pouvoir remplacer les DAC utilisés dans les conceptions actuelles par un circuit utilisant un PWM, suivi d'un filtre passe-bas. Cette solution si elle respecte les exigences pourrait s'avérer beaucoup moins lourde en terme de surface et de consommation. En jouant sur le rapport cyclique du PWM ainsi que sur la fréquence de sa porteuse on peut générer des tensions analogiques après filtrage plus ou moins précisent. On arrive donc à retrouver le comportement d'un DAC en générant une tension continue à la valeur souhaitée. La figure ci-dessous donne une idée claire du fonctionnement de ce système. On remarque un lien entre le rapport cyclique et la tension après filtrage. En commandant le rapport cyclique on peut donc générer la tension souhaitée en sortie, tout comme un DAC le fait.

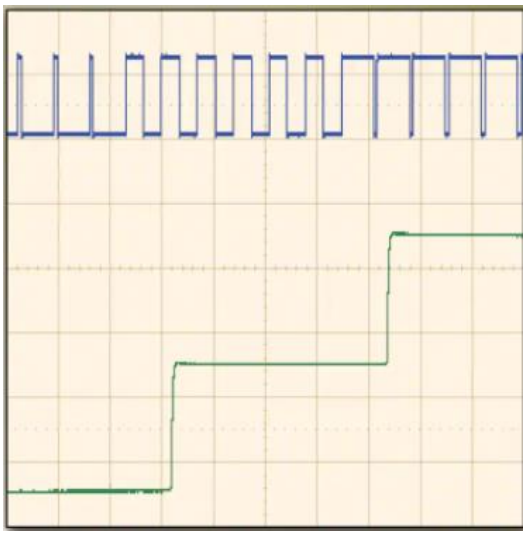


Figure 3 : Lien entre un PWM et le signal après le filtre

Cette solution a déjà fait ses preuves sur plusieurs autres systèmes embarqués et la littérature à son sujet est assez riche et claire. Le but de ce sujet est donc de concevoir un prototype de carte et d'évaluer précisément les gains ou les pertes en consommation, performances, coût et surface utilisée. L'objectif est bien sûr d'optimiser ces points.

En faisant une recherche rapide sur ce sujet, il semble qu'un des enjeux de cette méthode soit de trouver le juste équilibre entre la qualité du filtrage et le temps de réponse du système selon l'application. En effet, la littérature indique que les filtres passe-bas utilisés sont des filtres d'ordre 1. Augmenter l'ordre de ce filtre rendrait la méthode beaucoup moins intéressante comparée à la simple utilisation d'un DAC. De plus, réduire la fréquence de coupure du filtre augmentera le temps de réponse du système, ce qui peut être un problème, il faut donc trouver le compromis entre temps de réponse et qualité du filtrage tout en essayant de limiter la complexité du filtre au maximum pour que cette solution soit viable.

1.2. Choix du sujet et détails

Après plus de recherches pour comprendre les projets et leurs enjeux, j'ai décidé de choisir le premier sur la fonction ACS avec des amplificateurs de classe D. En effet, les deux sujets étaient très intéressants et pouvaient m'apporter beaucoup de compétences, mais la problématique de l'amplificateur classe D avec de la commutation, est quelque chose que je trouvais vraiment attirant et dans lequel je souhaitais approfondir mes connaissances.

Les deux sujets ont leurs propres particularités. Le second, avec le PWM est quelque chose de plus mature et avec plus de recul, mais les attentes étaient par conséquent plus grandes. Cela signifie qu'à la fin du stage un prototype qui montre des améliorations doit être réalisé, alors qu'avec le premier projet le but est plus de déterminer si cette solution peut vraiment fonctionner et être intéressante. Bien sûr, cela ne signifie pas qu'il ne doit pas avoir un prototype complet qui fonctionne correctement à la fin. Dans chaque projet, le but est d'améliorer les solutions actuelles, mais le deuxième projet est un peu plus avancé et donc la problématique et les défis ne sont pas les mêmes.

Pour conclure ce choix, le premier projet a une dimension de challenge que j'apprécie particulièrement, et je suis sûr que j'apprécierai et apprendrai beaucoup en travaillant dessus.

Avant de passer à la suite, il est important de comprendre plus précisément ce qu'est un amplificateur de classe D. Premièrement, le but d'un amplificateur de classe D est d'amplifier la puissance d'un signal tout en limitant les pertes, cela signifie que la puissance (P= V*I) sera plus grande après l'amplificateur qu'avant. Cela peut être fait en augmentant la valeur du courant et/ou de la tension. Si nous revenons à la Figure 2, nous pouvons identifier plusieurs étapes.

En entrée, il y a deux signaux qui peuvent provenir d'un DAC ou d'un microcontrôleur par exemple. Le premier est le signal que nous voulons amplifier. Les amplificateurs de classe D sont souvent utilisés pour les applications audio pour lesquels le signal d'entrée est donc un signal audio. Pour notre application, ce signal sera une sinusoïde, car nous voulons une sinusoïde en sortie permettant d'exciter un capteur. Le deuxième signal est un signal en dents de scie. En utilisant un comparateur et ces deux signaux, nous pouvons créer un PWM représentant une sinusoïde via ces rapports cycliques (SPWM). Ce n'est pas le seul moyen de générer un SPWM, il en existe d'autres comme les modulations delta ou Delta-Sigma, qui permettent de contrôler la stabilité et l'erreur du SPWM, mais sont un peu plus difficiles à mettre en œuvre. La méthode utilisant un comparateur et un signal en dent-de-scie mettra les potentiels haut et bas du PWM à Vcc et GND, ou Vcc- si l'alimentation est symétrique.

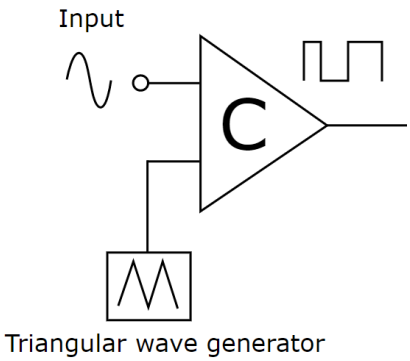


Figure 4: PWM modulation stage

Maintenant qu'un signal SPWM est généré, nous pouvons l'utiliser pour contrôler l'étage de sortie et ses MOSFET. Pour être plus précis cet étage est un demi-pont, les MOSFETS sont allumés et éteints alternativement, pour pousser et tirer le courant vers les potentiels d'alimentation. Le courant du signal sera amplifié dû aux caractéristiques des MOSFET et par conséquent la puissance aussi. Cet étage d'amplification est nécessaire pour satisfaire les contraintes en tension et courant exigés dans la charge, ce qui n'est pas possible directement avec un FPGA ou microcontrôleur ne pouvant pas fournir 10V et 50mA en sortie par exemple.

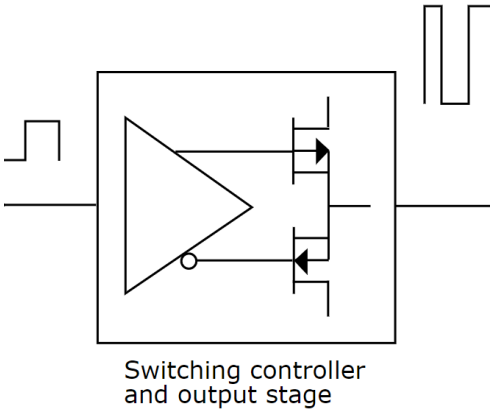


Figure 5: Amplifier stage

Enfin, pour récupérer la sinusoïde, un filtre doit être implémenté. Le filtre peut être un filtre passe-bas pour couper les hautes fréquences et ne garder que la fréquence fondamentale du sinus, mais un filtre passe-bande peut également être utilisé, si nous devons couper la partie DC pour supprimer le décalage en tension. Après cette étape, nous avons la sinusoïde que nous voulions.

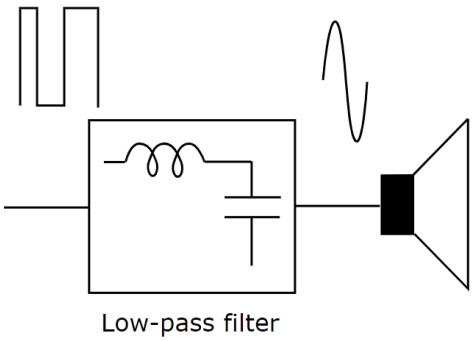


Figure 6: Etage de filtrage

Bien sûr, le fonctionnement du circuit qui va être développé pendant le stage va s'inspirer de ce fonctionnement mais ne va pas lui être identique à 100%. Le choix de l'architecture adaptée au stage sera expliqué dans le chapitre 3, en fonction du cahier des charges et de l'objectif du stage.

1.3. Les objectifs et compétences

L'objectif fixé au début du stage était de concevoir le circuit décrit ci-dessus, d'estimer son rendement théoriquement ainsi que son coût et la surface qu'il utilise, mais aussi de simuler ce circuit à l'aide du logiciel LTspice afin de confirmer la théorie. Enfin, si le temps restant suffit, de faire un PCB avec des composants préalablement sélectionnés afin de faire des tests et mesures sur le circuit réel. En parallèle de ça, de la documentation concernant la théorie, mais aussi les simulations et les tests vont être rédigés afin de garder une trace claire de mon stage et réutilisable dans le futur. En résumé, je vais devoir appliquer mes compétences en électronique analogique et numérique, en automatique et informatique embarquée avec l'asservissement de la tension de sortie évoqué plus tôt dans le rapport. J'ai notamment utilisé des logiciels sur lesquels j'ai été formé lors de mes études à l'INSA, LTspice pour les simulations électroniques, KICAD EDA pour la schématique et le routage, MATLAB et Simulink pour les simulations de l'asservissement, STM32CubeIDE pour l'informatique embarquée, mais aussi des sites web tels que Octopart pour la gestion des BOM. De plus, j'ai découvert et me suis formé au logiciel Mathcad, un outil mathématique très puissant et utile.

J'ai réalisé toutes ces tâches seul et non dans une équipe particulière, cependant j'ai eu un suivi hebdomadaire tout au long du stage afin de confirmer mes choix, de me réorienter dans la bonne direction si besoin et de me donner des conseils. Ce qui m'a permis d'avoir cette liberté au niveau de mes choix étant donné que je pouvais faire valider ceux-là par des

personnes d’expériences et donc être sûr de ne pas m’égarer, mais aussi d’apprendre de leur expérience en discutant avec eux chaque semaine.

Dans la suite nous allons nous attarder sur la partie plus technique de mon stage, dans laquelle je vais détailler les différentes étapes que j’ai traversées tout au long de mon stage.

CHAPITRE 3 REALISATION

1.1. Planning

Une des premières choses que j'ai faite a été de créer un planning. Pour ce faire j'ai utilisé quelques documents qui présentent un cycle de vie d'un projet en V comme on peut le voir sur la figure 7 ci-dessous.

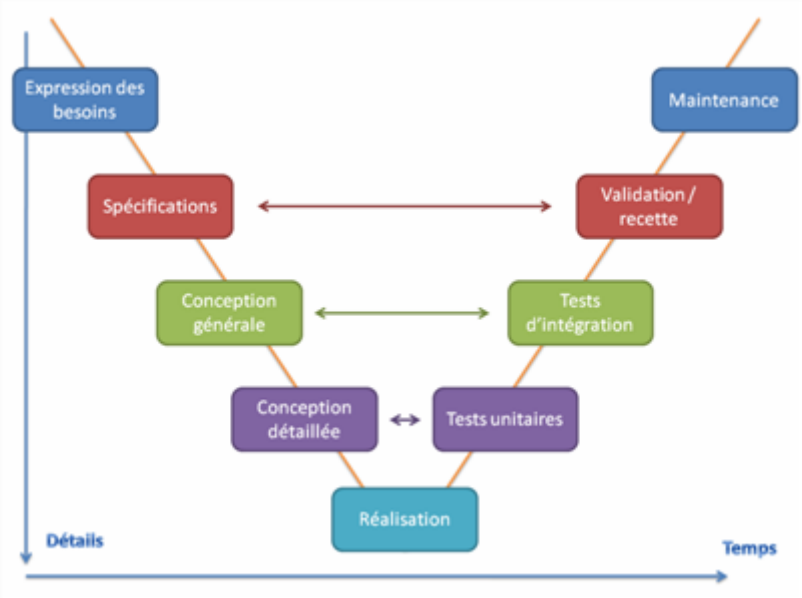


Figure 7 : Schéma du cycle de vie d'un projet

Dans un premier temps, j'ai décidé de faire mon planning au format GANTT. J'ai identifié à partir de la figure 7, les parties globales de ce projet puis j'ai essayé de trouver des sous-parties pour chacune d'entre elles. Ensuite, j'ai estimé le temps pour chaque sous-partie et chaque partie et vérifié si il était cohérent avec l'équipe qui m'encadrait sur ce projet. Enfin, j'ai obtenu ce que vous pouvez voir sur la figure suivante.

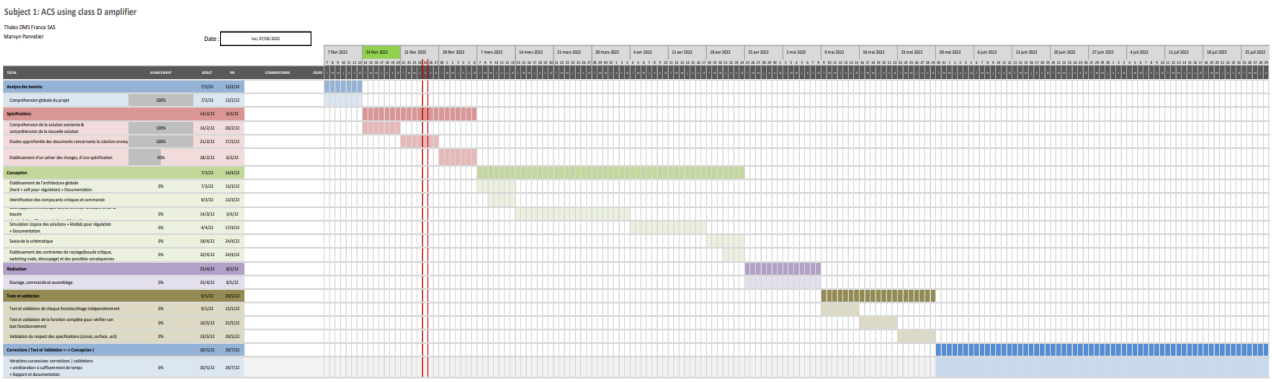


Figure 8: Vue globale du planning

Bien sûr, on ne pas voir grand-chose sur la figure 8, nous allons donc zoomer sur les différentes parties et les détailler.

TÂCHE	AVANCEMENT	DÉBUT	FIN
Analyse des besoins		7/2/22	13/2/22
Compréhension globale du projet	100%	7/2/22	13/2/22
Spécifications		14/2/22	6/3/22
Compréhension de la solution existante & compréhension de la nouvelle solution	100%	14/2/22	20/2/22
Etudes approfondie des documents concernants la colution envisagé	100%	21/2/22	27/2/22
Etablissement d'un cahier des charges, d'une spécification.	50%	28/2/22	6/3/22
Conception		7/3/22	24/4/22
Etablissement de l'architecture globale (hard + soft pour régulation) + Documentation	0%	7/3/22	13/3/22
Identification des composants critiques et commande		9/3/22	13/3/22
Développement théorique des différentes fonctions et de la boucle de régulation (Documentation + Matcad)	0%	14/3/22	3/4/22
Simulation Ltspice des solutions + Matlab pour régulation + Documentation	0%	4/4/22	17/4/22
Saisie de la schématique	0%	18/4/22	24/4/22
Etablissement des contraintes de routage(boucle critique, switching node, découpage) et des possibles conséquences	0%	20/4/22	24/4/22
Réalisation		25/4/22	8/5/22
Routage, commande et assemblage	0%	25/4/22	8/5/22
Tests et validation		9/5/22	29/5/22
Test et validation de chaque fonction/étage indépendemment	0%	9/5/22	15/5/22
Test et validation de la fonction complète pour vérifier son bon fonctionnement	0%	16/5/22	22/5/22
Validation du respect des specifications (conso, surface...ect)	0%	23/5/22	29/5/22
Corrections (Test et Validation <--> Conception)		30/5/22	29/7/22
Itérations successives: corrections | validations + amélioration si suffisemment de temps + Rapport et documentation	0%	30/5/22	29/7/22

Figure 9: Zoom sur le planning

Sur la figure 9, toutes les parties et sous-parties apparaissent. Vous trouverez ci-dessous une liste des différentes parties et une explication des tâches à accomplir, ainsi que du temps estimé pour chacune.

1. Analyse des besoins

Cette première partie consiste à appréhender le but du projet et ces enjeux, afin d'être efficace le plus tôt possible.

2. Spécification

Dans cette partie, mon objectif est de comprendre le projet dans sa globalité, d'avoir une première idée des circuits, composants qui seront nécessaires, et de faire des recherches sur les circuits principaux, comme l'amplificateur classe D. De plus, je dois faire un cahier des charges avec des valeurs, des besoins et des attentes clairs. Ainsi, j'aurai mes objectifs pour la fin du stage.

3. Conception

C'est une partie importante car elle inclut de développer une architecture globale, de trouver des composants critiques, de théoriser et de simuler des solutions. Quand tout cela sera fait correctement, il faut créer un schéma final et quelques contraintes de routage importantes pour passer à l'étape suivante.

4. Implémentation

L'objectif de cette partie est de prévoir du temps pour acheminer, commander et assembler la carte prototype qui sera utilisée dans les parties suivantes.

5. Tests and validation

Afin de tester correctement le PCB, mon idée est d'abord de tester chaque partie de circuit indépendamment et de les faire fonctionner, puis de tester tous le circuit pour s'assurer que les fonctions principales fonctionnent. Après cela, l'attention doit être portée sur le cahier des charges, répond-il à toutes les exigences ? Que peut-on encore améliorer ?

6. Corrections

Enfin, ce temps est là pour résoudre les problèmes s'il y en a, pour améliorer certaines choses, pour mieux répondre au cahier des charges en revenant à la conception par exemple, mais aussi pour donner une réponse concernant la viabilité de cette solution, si c'est une bonne idée pour les futurs produits ou non. Encore une chose, ce temps servira également à finaliser la documentation et à capitaliser les informations même si cela se fait aussi pendant le reste du stage.

1.2. Evolution du planning

Les dates visibles sur la Figure 9 sont celles fixées dès le début du stage. Bien sûr celles-ci ont évolué en fonction des difficultés rencontrées ou non au fur et à mesure. Cette partie a pour but d'expliquer de façon générale les changements qu'il y a eu sur ce projet et pourquoi.

Pour commencer, les deux premières parties se sont passé comme prévu et la 3ème partie sur la conception a même pu débuter fin février. Comme expliqué plus haut, ces deux premières parties consistent essentiellement en la compréhension du projet, des objectifs, la création du cahier des charges…etc. J'ai donc eu le temps nécessaire pour faire cela dans le temps que je m'étais fixé au tout début du stage.

Abordons à présent la partie conception, qui est sans aucun doute une étape majeure de ce stage. Pour essayer de mieux prévoir la durée de cette période, la conception a été divisés en plusieurs sous-parties et le temps global a été respecté. Seulement en réalité l'étape de développement théorique a duré moins longtemps que prévu, et l'étape de simulation LTspice plus longtemps, surtout dû à la simulation du filtre passe bande. Ce qui au final se compense et respecte relativement bien le planning initial.

L'étape suivante est dédiée au routage et assemblage du PCB. Cette étape était prévue sur deux semaines mais une formation a retardé le routage. De plus, nous avons décidé de faire le PCB en local, avec le Fablab présent à Thales plutôt que de le commander à un fabriquant spécialisé dans la fabrication de PCB mais qui aurait mis plus de temps à le livrer. Ce choix a aussi induit un retard car la technologie utilisée au Fablab ne permet pas de faire des pistes trop petites ou bien d'utiliser des composants avec des pas trop petits. Il a donc fallu chercher des alternatives à des composants déjà sélectionnés mais ne pouvant donc pas être utilisés, ce qui a pris une ou deux journées de plus que prévu. Après cela le PCB a pu être fabriqué et il a fallu souder chaque composant à la main et faire des tests de continuité, ce qui a aussi pris 2 ou 3 jours afin d'avoir fini d'assembler.

Enfin l'étape des tests a commencé par l'apparition de quelques courts-circuits qu'il a fallu identifier et corriger, ce qui a aussi pris quelques jours avant d'avoir une carte réellement fonctionnelle et pouvoir attaquer les tests. Au moment où ce rapport est rédigé les tests sont en cours et il reste environ 2 mois afin d'optimiser les résultats et de conclure sur cette conception.

Pour résumer, environ deux semaines de retard sont apparues par rapport au planning initial, ce qui n'est pas un problème étant donné qu'une période de plus d'un mois a été prévu pour résoudre ce genre de problème. Finalement le planning correspondant à l'évolution du stage jusqu'à aujourd'hui ressemble au suivant :

TÂCHE	AVANCEMENT	DÉBUT	FIN
Analyse des besoins		7/2/22	13/2/22
Compréhension globale du projet	100%	7/2/22	13/2/22
Spécifications		14/2/22	6/3/22
Compréhension de la solution existante & compréhension de la nouvelle solution	100%	14/2/22	20/2/22
Etudes approfondie des documents concernants la colution envisagé	100%	21/2/22	27/2/22
Etablissement d'un cahier des charges, d'une spécification regroupant les points importants.	100%	28/2/22	6/3/22
Conception		7/3/22	24/4/22
Etablissement de l'architecture globale (hard + soft pour régulation) + Documentation	100%	7/3/22	13/3/22
Identification des composants critiques et commande	100%	9/3/22	27/3/22
Développement théorique des différentes fonctions et de la boucle de régulation (Documentation + Matcad)	100%	14/3/22	3/4/22
Simulation Ltspice des solutions + Matlab pour régulation + Documentation	100%	4/4/22	17/4/22
Saisie de la schématique	100%	18/4/22	24/4/22
Etablissement des contraintes de routage(boucle critique, switching node, découpage) et des possibles conséquences	100%	20/4/22	24/4/22
Réalisation		25/4/22	22/5/22
Routage, choix des composants, commande et assemblage, et correction des court-circuit	100%	25/4/22	22/5/22
Tests et validation		23/5/22	19/6/22
Test et validation de chaque fonction/étage indépendemment	100%	23/5/22	29/5/22
Test et validation de la fonction complète pour vérifier son bon fonctionnement	100%	30/5/22	12/6/22
Validation du respect des specifications (conso, surface...ect)	10%	13/6/22	19/6/22
Corrections (Test et Validation <--> Conception)		20/6/22	29/7/22
Itérations successives: corrections | validations + amélioration si suffisemment de temps + Rapport et documentation	0%	20/6/22	29/7/22

Figure 10 : Planning final

À présent, nous allons passer à l'aspect plus technique de ce stage en commençant par décrire le cahier des charges puis par parler des architectures imaginées pour réaliser la fonction ACS demandée.

1.3. Cahier des charges

Ici va être décrit le cahier des charges qui a cadré le stage et a permis de définir des objectifs. Pour cela, je me suis appuyé sur deux types d'informations. La première étant les performances demandées à la fonction ACS dans les projets en général. La deuxième est la performance du circuit actuel, appelé ELAC, afin de fixer les objectifs à atteindre pour proposer une solution plus performante.

Tension différentielle de sortie aux bornes de la charge	7.0+-0.7V RMS
Forme du signal de sortie	Sinusoïde
Fréquence	Doit être capable de générer des sinusoïdes de fréquence comprise entre 2kHz et 5kHz
Précision de la fréquence du sinus de sortie entre les circuits ACS	Inférieur à 0.1Hz
Courant dans la charge	Entre 10mA et 40mA RMS
Caractéristique de la charge	Charge inductive avec résistance parasite ayant pour facteur de puissance cos(φ) = 0.2079 soit φ=78°
Surface	La surface actuelle pour un ACS ELAC est de 740mm2, il faut donc faire moins
Rendement	Un ACS ELAC a un rendement inférieur à 10%, il faut donc être supérieur à ce rendement et l'optimiser au maximum.
Ondulation sur le signal et stabilité	Inférieur à 100mV
THD du sinus de sortie	Inférieur à 1%
Coût	Pas de référence mais le plus bas possible tout en respectant les points ci-dessus.

1.4. Les architectures pour répondre au besoin

Pour rappel, le but est de remplacer un circuit réalisant la fonction ACS à base d'amplificateurs linéaires (AOP), par un circuit utilisant le phénomène de commutation et ses avantages. Pour cela, plusieurs solutions étaient possibles et j'ai donc décidé de les étudier et comparer afin de faire un choix et commencer la conception. Pour être plus précis, j'ai rapidement identifié 3 architectures possibles.

- La première utilise un composant dit COTS, c'est-à-dire un composant disponible sur le marché et prêt à l'emploi. Ce composant est un amplificateur audio de classe D qui prend en entrée un signal, qui serait un sinus dans notre cas, et qui a pour sortie un SPWM amplifiée représentant le sinus d'entrée. L'avantage de cette solution est qu'elle peut être bien en terme de surface utilisée car ce type de composants fonctionnent avec de la microélectronique et non des composants analogiques, ce qui permet d'obtenir des composants très peu encombrants. Le coût peut aussi être un atout de cette solution mais avec la crise des composants actuelle il ne l'est plus vraiment, car les seuls composants encore disponibles sont à des prix assez élevés. Concernant les aspects négatifs, on peut y trouver la consommation étant donné que ce sont des composants conçus pour une application audio et qui intègre plusieurs circuits de sécurité, filtrage, transformation du signal…etc, qui ne sont pas nécessaires à notre application et qui engendrent des consommations supplémentaires. De plus, il peut être intéressant de prendre un PWM en entrée du système global car il peut être généré avec un FPGA est donc peu coûteux en surface et consommation. Ce qui imposerait de prévoir une étape de filtrage avant de rentrer dans l'amplificateur, qui demande une entrée analogique. Ce qui peut perturber le système et rend la solution moins viable.

- La deuxième solution est plus simple à expliquer puisqu'elle consiste aussi à utiliser un composant COTS, mais cette fois-ci avec une entrée PWM directement. Cela permet d'éviter le problème évoqué à la fin du dernier point et de garder le point positif de la surface utilisée assez faible. Seulement ce type de composant est assez rare et cher, surtout avec la crise, ce qui rend cette solution beaucoup moins attrayante. Enfin, encore une fois cette solution pourrait répondre au besoin mais serait toujours surdimensionnée et sur-consommatrice étant donné que ces types de composants intègre des circuits de sécurité, traitement de signal…etc, et sont conçus pour fonctionner avec des signaux bien plus complexes qu'un sinus.

- Enfin la dernière architecture imaginée consiste à concevoir par moi-même un circuit à base d'étages push-pull ou demi-pont qui viendrait amplifier la PWM générée par le FPGA avec l'intermédiaire d'un driver de MOSFET pour commander correctement l'étage amplificateur. Cette solution à le bénéfice de s'adapter au cahier des charges sans en faire plus que besoin, elle permet aussi de diminuer la consommation du système de façon assez forte. Quant au coût ainsi qu'à la surface utilisée, ils semblent tous les deux pouvoir concurrencer les deux solutions ci-dessus, ou en être proche.

Bien sûr un filtre passe bas ou bande est nécessaire à la sortie de ces trois solutions afin de filtrer les PWM qui ont été amplifiées et récupérer le sinus souhaité. Le tableau ci-dessous compare les trois solutions en exposant des chiffres afin de se faire une idée plus précise quant à la solution à retenir.

	Composant COTS avec entrée analogique	Composant COTS avec entrée PWM	MOSFET driver + étage d'amplification
Consommation	35mA + courant tiré par la charge	7.3mA + courant tiré par la charge	Moins de 1mA + courant tiré par la charge
Surface	Sans le filtre, entre 50 et 70mm2	Sans le filtre, entre 60 et 90mm2	4 double NMOS: 4*4.2=17.2mm2 4 gate drivers: 4*9=36mm2 Total: 53.2mm2
Prix (pour 500 unités)	Entre 1 et 2€ le composant	Plus de 3€	NMOS : 4*0.266€ Driver: 4*0.243€ Total=2.036€
Disponibilité			

Bien Moyennement bien Mauvais

À partir de ces résultats il m'a semblé plus cohérent de choisir la troisième solution n'utilisant pas de composants COTS, car celle-ci permet de développer la meilleure solution en terme de rendement tout en rivalisant avec les deux autres sur la surface utilisée et le coût. De plus, malgré que ça ne soit pas un argument dans le choix de la meilleure solution, cette architecture m'a permis de développer des compétences plus approfondies autour de la commutation puisqu'il faut concevoir un étage d'amplification à base de MOSFET. Dans la suite de ce rapport nous considérerons cette architecture.

1.5. Conception

Dans cette partie nous allons aborder les différentes étapes de ma progression dans la conception du circuit. Mais avant ça, on peut observer ci-dessous un schéma plus précis de l'architecture choisie. J'ai fait ce schéma afin de définir l'allure globale du circuit et les différents points à développer et qui vont d'ailleurs être décrits dans les sous-parties suivantes.

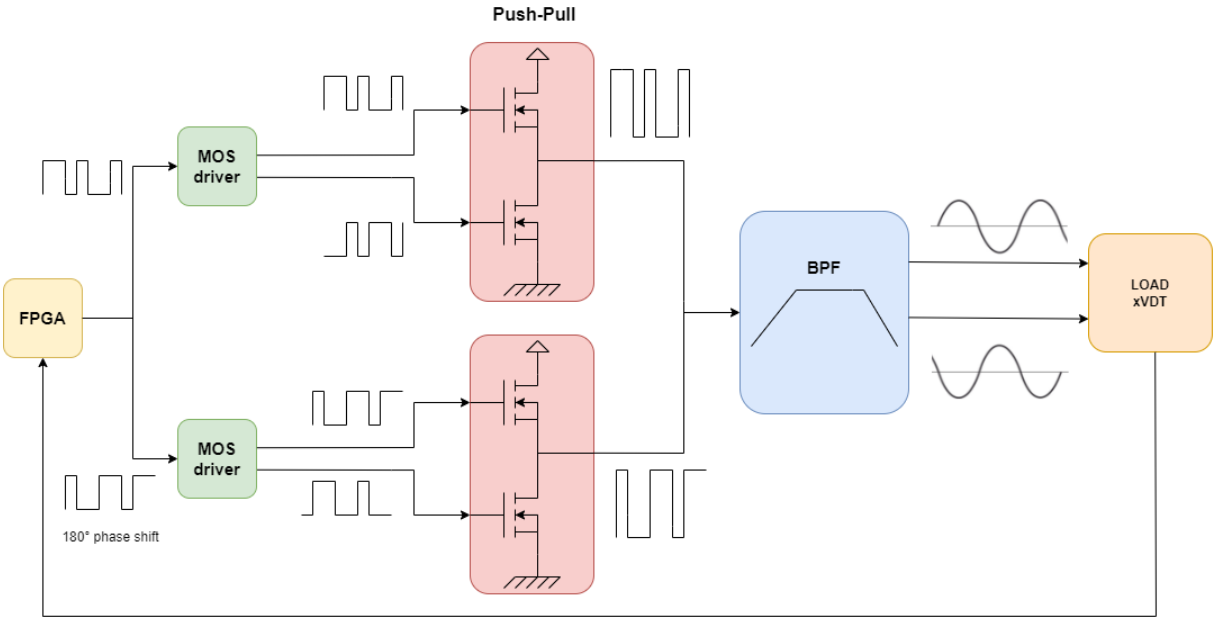


Figure 11 : Schéma du circuit à concevoir

Ce schéma a été réalisé au début du stage et pour le stage aucun FPGA ne sera utilisé mais la génération des PWM et l'asservissement de la tension de sortie seront réalisés à l'aide d'une carte NUCLEO-H743ZI2.

1.5.1. Etage d'amplification à base de MOSFET

Commençons par l'étage d'amplification. Afin de développer cet étage j'ai commencé par faire des recherches sur les différents moyens de réaliser cet étage. Il en est ressorti 2 montages différents. Le premier utilisant 2 NMOS et le deuxième utilisant un NMOS et un PMOS comme on peut le voir sur la figure ci-dessous.

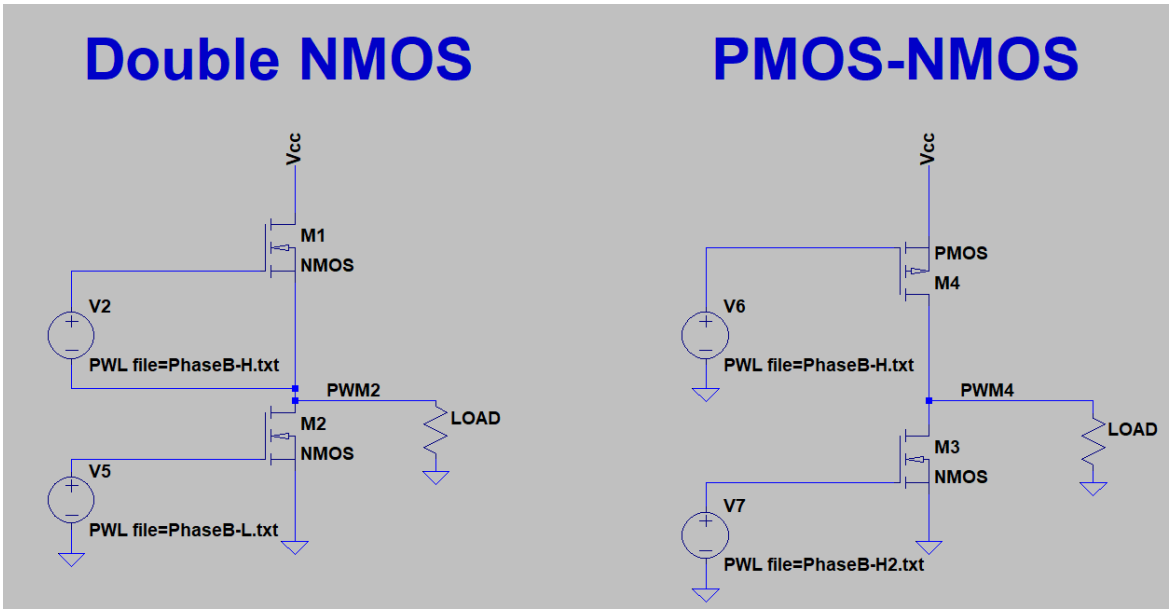


Figure 12 : Schéma des deux types de montage d'amplification

Sur cette figure on remarque, quelques différences. La première concerne la façon de commander les MOSFET. Pour rappel pour commander un MOSFET on agit sur la tension Vgs, celle-ci doit être positive et supérieure à un seuil imposé par le composant pour qu'un NMOS soit passant, et au contraire négative pour qu'un PMOS soit passant.

Sur la configuration à deux NMOS, le NMOS du bas a son émetteur à la masse, il suffit donc d'appliquer une tension positive et supérieure au seuil du NMOS pour qu'il devienne passant, et d'appliquer une tension nulle pour le bloquer, ce qui est fait avec une PWM entre 0 et 3.3V par exemple. Concernant le NMOS du haut, ce n'est pas la même histoire étant donné que le potentiel de son émetteur n'est pas connu à un instant précis. Ma première idée fût de commander ce MOSFET avec une PWM ayant un potentiel haut proche de celui de l'alimentation afin de m'assurer qu'il est au-dessus du potentiel de l'émetteur et par conséquent faire entrer le NMOS en conduction. Seulement après quelques recherches, je me suis rendu compte que cette méthode n'était pas optimale et qu'il valait mieux faire ce qu'on peut voir sur la figure 12. Cette méthode consiste à référencer le potentiel bas de la PWM au potentiel de l'émetteur, ce qui est fait dans les drivers destinés à commander ce genre de circuit.

Le NMOS du bas sur la configuration utilisant un PMOS et un NMOS fonctionne de la même façon que sur le montage précédent. Le PMOS lui fonctionne à l'inverse, je m'explique. Son émetteur étant à Vcc et ayant rappelé que sa tension Vgs doit être négative pour qu'il soit passant, il faut donc un potentiel sur la grille inférieure à Vcc moins la tension de seuil afin qu'il soit passant. En revanche, lorsque le potentiel de grille est à Vcc, la tension Vgs est nulle et le PMOS devient bloquant. Il faut donc le commander avec une PWM dont l'état haut est à Vcc et bas à 0V ou du moins en dessous du seuil.

Un autre point important est ce qui est appelé deadtime. Le deadtime comme son nom l'indique est un temps mort appliqué entre les 2 PWM de commande afin d'éviter des courts-circuits. En effet, si les deux transistors commutent en même temps il existe un temps durant lequel les deux transistors sont passants et créent un court-circuit. Ce phénomène est dû au temps de monter et descente des PWM qui ne sont jamais nuls dans la réalité. Un moyen simple de contrer cela et d'insérer un temps mort entre la commande des deux transistors, de cette façon lorsque le deuxième commutera, le premier l'aura déjà fait et donc pas de problème de court-circuit.

Finalement, j'ai fait le choix d'utiliser le montage utilisant deux NMOS car il est plus courant et on trouve donc plus facilement des drivers adaptés à ce montage, et deuxièmement un NMOS se base sur de la conduction d'électrons contrairement aux PMOS qui se base sur la conduction des trous, les électrons ayant une plus grande mobilité que les trous, cela rend les NMOS souvent plus performants. Ce type de montage a pour nom « demi-pont ».

Sur la figure 12 on peut voir que je génère les PWM à l'aide de générateurs de tension et de fichiers texte. En effet, afin de rendre plus simple les simulations j'ai utilisé un script Matlab trouvé sur internet, que j'ai ensuite modifié pour qu'il convienne à mon application. Ce script prend des paramètres en entrée, tels que la fréquence de la PWM, les temps de montée et descente des fronts, le pas, le deadtime, la tension haute et basse ainsi que l'indice de modulation, puis génère 4 fichiers textes avec les valeurs des PWM. Deux PWM représentant des sinus déphasés de 180° et les deux autres PWM sont les opposés de ceux-là. Ce qui permet de commander les NMOS dans LTspice, comme indiqué sur la Figure 11. Les deux figures suivantes représentent une simulation d'un des deux demi-ponts.

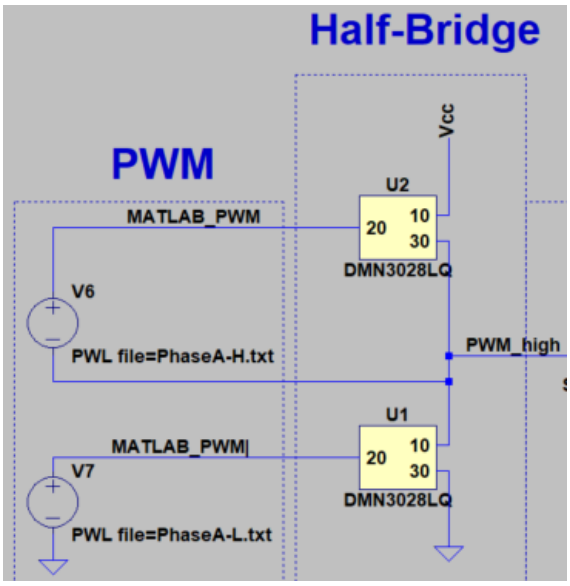


Figure 13 : Schéma LTspice d'un étage demi pont

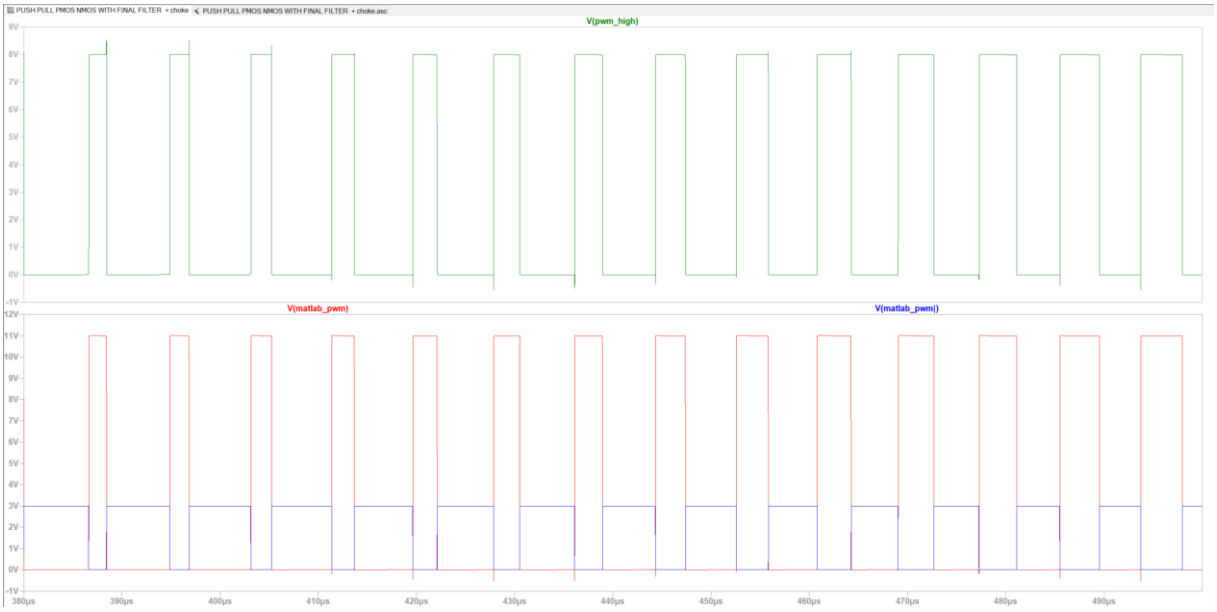


Figure 14 : Simulation des PWM d'un demi pont sur LTspice

À présent que le résultat en sortie des demi-ponts est satisfaisant, nous pouvons passer à un point pas des moins importants de ce projet, le filtrage !

1.5.2. Filtre passe bande

Justification des choix

Le filtrage est essentiel puisqu'il va permettre de passer d'un signal PWM à une sinusoïde. Dans un premier temps j'ai d'abord pensé à un filtre passe bas LC, car ce type de filtre contrairement aux filtres RC ne génère de pas ou peu de pertes. Bien sûr, en réalité il existera toujours les résistances parasites (ESR) qui vont créer des pertes mais moindres comparées à un filtre RC. Ensuite, j'ai eu rapidement à ajouter un filtre passe haut CR en série afin de supprimer la composante continue sur les deux sorties différentielles, les charges (xVDT) n'acceptant pas de composante continue à leurs bornes. Cette fois, un filtre CR n'est pas gênant puisque la résistance est placée en parallèle et très peu de courant ne va passer à l'intérieur grâce à sa valeur assez élevée.

Après ça, je me suis intéressé à un circuit appelé snubber. Le circuit LC a l'avantage de n'engendrer que peu de pertes, cependant ce type de filtre crée un phénomène de résonance autour de la fréquence de coupure du filtre. Pour atténuer le pic de résonance on utilise un snubber, ce circuit va filtrer le courant du pic et le dissiper via la résistance. Enfin on place ces filtres sur les deux sorties et on peut les relier via la charge. On obtient donc le schéma suivant.

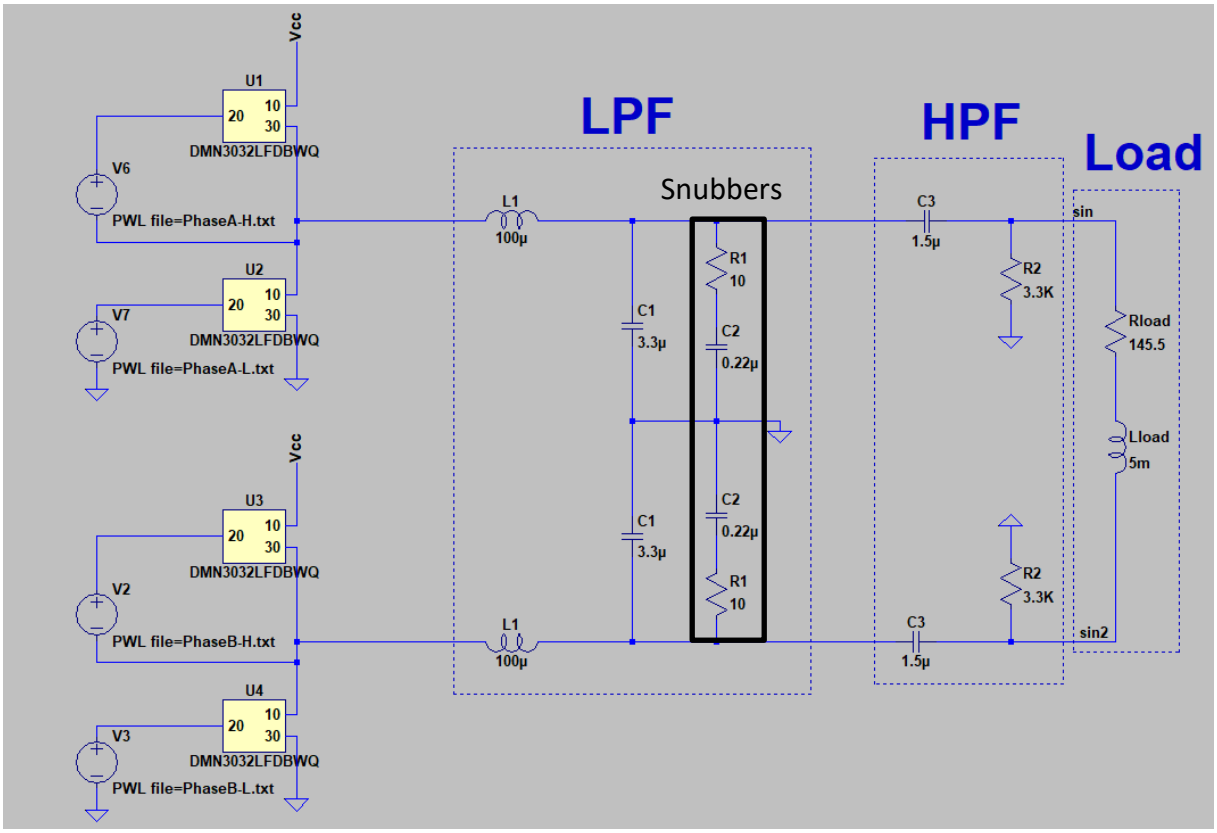


Figure 15 : Schéma LTspice demi pont + filtre

Une fois ce filtre conçu, il a fallu le dimensionner en trouvant les valeurs de chaque composant. Pour cela, j'ai utilisé LTspice pour faire des simulations ainsi que des diagrammes de bode de mon filtre, jusqu'à ce que le résultat obtenu convienne. Ce filtre est très important car il va définir le THD du sinus de sortie, qui doit être inférieur à 1% pour respecter le cahier des charges.

Rappel sur le THD

Pour rappel, le THD (total harmonic distortion en anglais) est un indicateur qui vient indiquer à quel point un sinus est parfait ou non dans notre cas. Son calcul peut se comprendre à l'aide de la figure suivante.

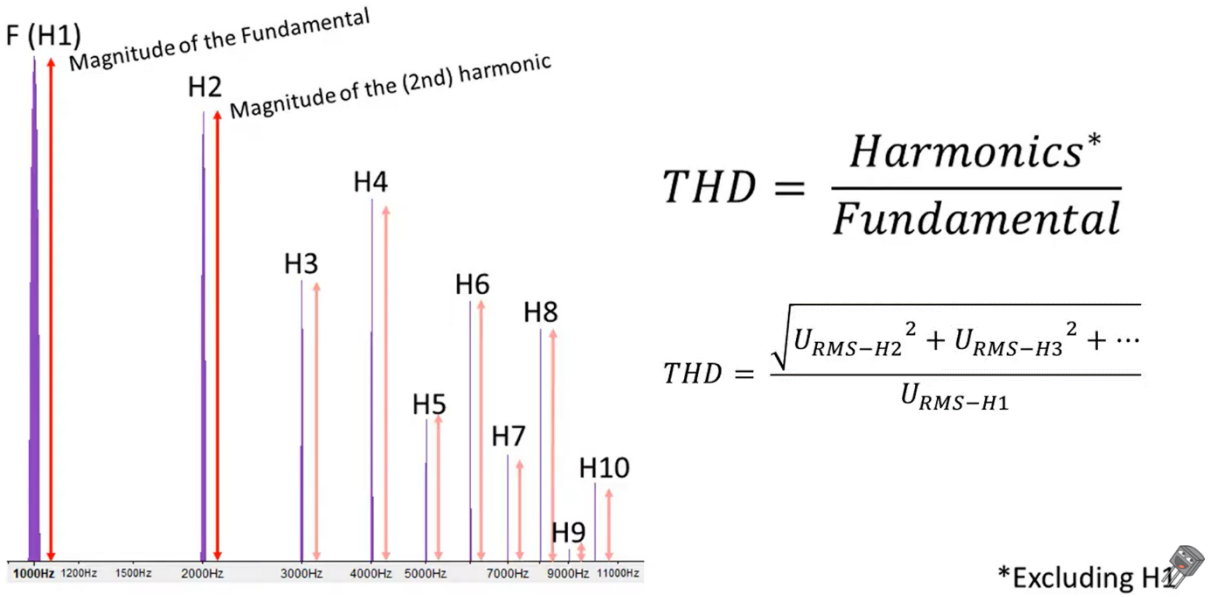


Figure 16 : Explication du calcul du THD

On remarque que le calcul du THD correspond à un rapport entre les harmoniques et la fondamentale du signal de sortie. Prenons l'exemple de ce stage, si la PWM qui porte le sinus a une fréquence de 300kHz, qui porte un sinus de fréquence 5kHz, on sait qu'après le filtrage on va retrouver un pique à la fréquence du sinus puis des harmoniques à tous les multiples pairs et impairs. De plus, on retrouve la fréquence de la PWM, 300kHz et ses harmoniques aussi, qui sont beaucoup plus hautes en fréquences et donc très peu gênantes.

Concernant le THD, les harmoniques qui vont le plus avoir d'influence sur celui-ci sont les harmoniques les plus proches de la fréquence de la fondamentale, puisque les harmoniques s'atténuent naturellement au fur et à mesure qu'elles s'éloignent de la fondamentale. Pour un sinus à 5kHz il faut donc atténuer correctement le 10kHz et 15kHz par

exemple. Cependant, si le signal PWM est propre on ne doit trouver que l'harmonique du sinus à 5kHz et les harmoniques de la PWM qui sont trop hautes en fréquence pour être gênantes après filtrage.

Le cahier des charges indique qu'il faut être capable de générer des sinus entre 2kHz et 5kHz. J'ai donc fait le choix de concevoir un filtre passe bande qui laisse passer les fréquences un peu inférieures à 2kHz et au moins supérieur à 5kHz. De cette façon je pourrais filtrer les harmoniques au-dessus 5kHz si elles existent et garantir un bon THD. C'est pourquoi nous allons maintenant détailler le dimensionnement des composants.

Justification du dimensionnement

En ce qui concerne le premier filtre la fréquence de coupure est la suivante :

$$f_o := \frac{1}{2\pi \cdot \sqrt{L1 \cdot C_1}} = 15.915 \times 10^3 \frac{1}{s}$$

Soit une fréquence de coupure à 16kHz.

Sur la Figure 15, on remarque les circuits snubber en parallèle des condensateurs du premier filtre. Le dimensionnement des snubbers a été fait à l'aide des formules suivantes et on peut voir les résultats obtenus ci- dessous.

$$R_{snub} = \sqrt{L_p / C_p}$$

$$C_{snub} = 1/(2\pi R_{snub} F_{ring})$$

$$R_{snub} := \sqrt{\frac{100 \cdot 10^{-6} H}{3.3 \cdot 10^{-6} F}} = 5.505\Omega \qquad C_{snub} := \frac{1}{2\pi \cdot R_{snub} \cdot f_o} = 3.3 \times 10^{-6} F$$

J'ai cependant décidé de prendre d'autres valeurs après plusieurs simulations, qui sont R=10 et C=0.22uF. Ce choix est dû à des contraintes de pertes de puissance qui seront décrites ultérieurement.

La partie suivante est composée d'un condensateur et d'une resistance en parallèle de la charge comme on peut le voir sur la figure 15. L'idée est de faire un filtre CR passe-haut. Celui-ci coupait entre 700Hz et 2500Hz selon la charge. Les résistances de 3.3kΩ sont là pour reprendre la masse comme référence qui va par conséquent effacer la composante continue, mais elles n'ont pas réellement d'impact sur le filtrage état donné qu'elles sont bien plus grande que la résistance de la charge, ce qui donne une résistance équivalente dépendante surtout de la charge. Ensuite, on peut identifier simplement un circuit série RLC ou (CRL ici). Voici la fonction de transfert de ce circuit :

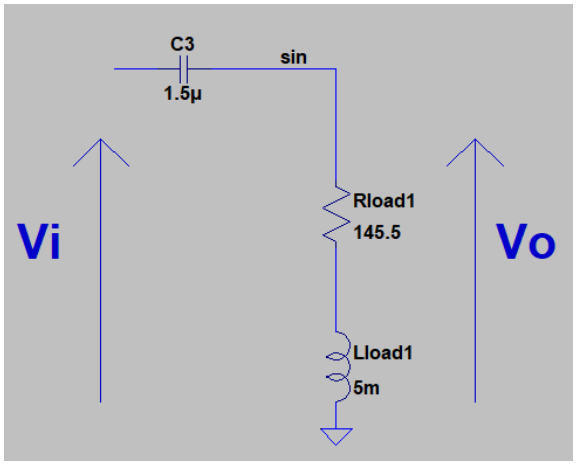


Figure 17 : Circuit CRL de sortie

$$\frac{Vo}{Vi}(s) := \frac{R + Ls}{\frac{1}{Cs} + R + Ls}$$

$$\frac{Vo}{Vi}(s) := \frac{RCs + LCs^2}{1 + RCs + LCs^2}$$

$$\frac{Vo}{Vi}(jw) := \frac{RCjw - LCw^2}{1 + RCjw - LCw^2}$$

On retrouve une fréquence propre similaire à celle d'un passe haut du second ordre : $\omega_o := \frac{1}{\sqrt{LC}}$

Nous verrons plus loin que pour un courant de charge compris entre 10mA et 40mA, on a respectivement Lload=5mH min et Lload=60mH max. La valeur des condensateurs a été fixé à 1.5uF mais Ceq est la capacité équivalente aux 2 condensateurs en série due à la sortie différentielle, soit 0.75uF.

Pour L=5mH on a :

$$\omega_o := \frac{1}{\sqrt{5\cdot 10^{-3}\,H\cdot 0.75\cdot 10^{-6}\,F}} = 16.33\times 10^3\,\frac{1}{s} \qquad F_o := \frac{\omega_o}{2\pi} = 2.599\times 10^3\,\frac{1}{s}$$

Et L=60mH on a :

$$\omega_o := \frac{1}{\sqrt{60\cdot 10^{-3}\,H\cdot 0.75\cdot 10^{-6}\,F}} = 4.714\times 10^3\,\frac{1}{s} \qquad F_o := \frac{\omega_o}{2\pi} = 750.264\,\frac{1}{s}$$

Ce qui donne des fréquences propres comprises entre 750Hz et 2.6kHz. La fréquence propre de 2.6kHz n'est pas un problème car on verra que la fréquence du sinus est toujours supérieure à la fréquence propre. En effet la fréquence du sinus va jouer sur la valeur de L pour conserver le facteur de puissance cos(φ) indiqué dans le cahier des charges.

D'ailleurs nous allons maintenant calculer les valeurs de la résistance et inductance de charge afin de respecter le cahier des charges.

Calcul des valeurs de la charge

Premièrement, le circuit doit fonctionner avec un courant de 10mA à 40mA dans la charge. Deuxièmement, la charge a un caractère inductif et par conséquent introduit de la puissance réactive et son facteur de puissance cos(φ). Le capteur xVDT que le circuit ACS doit piloter à un φ égal à 78° environ. De plus nous connaissons les deux équations suivantes avec R et L la résistance et la capacité de charge.

$$\left|Z_{load}(s)\right| := \sqrt{R^2 + (L\omega)^2} \qquad \arg\left(Z_{load}(s)\right) := \arctan\left(\frac{L\omega}{R}\right) = \blacksquare\cdot\varphi$$

Le capteur xVDT doit être excité par une sinusoïde de 7V RMS. Nous pouvons donc calculer une plage d'impédance à partir de la plage actuelle que nous avons :

$$Z_{load.max} := \frac{7V}{10\cdot 10^{-3}A} = 700\,\Omega \qquad Z_{load.max} := \frac{7V}{40\cdot 10^{-3}A} = 175\,\Omega$$

Maintenant, trouvons la valeur R et L avec les données dont nous disposons. On peut avant tout exprimer L en fonction de R :

$$\arctan\left(\frac{L\omega}{R}\right) := \varphi = \blacksquare\cdot 78° \quad \Rightarrow \quad \frac{L\omega}{R} := \tan(78°) \quad \Rightarrow \quad L := \tan(78°)\cdot\frac{R}{\omega}$$

Pour un courant dans la charge de 10mA :

$$\left|Z_{load}\right| = \blacksquare\cdot 700 \quad \Rightarrow \quad \sqrt{R^2 + (L\omega)^2} = \blacksquare\cdot 700$$

$$\Rightarrow \quad R^2 + (L\omega)^2 := 700^2 \quad \text{On remplace L}$$

$$\Rightarrow \quad R^2 + (\tan(78))^2\cdot R^2 := 700^2$$

$$\Rightarrow \qquad R := \sqrt{\frac{700^2}{1 + (\tan(78°))^2}} = 145.538$$

Ensuite, nous pouvons trouver L pour la fréquence dont nous avons besoin. Pour nous la gamme de fréquences va de 2kHz à 5kHz

Pour 2kHz :

$$L_{2kHz} := \tan(78°) \cdot \frac{145.54\Omega}{2 \cdot \pi \cdot f_{2kHz}} = 54.488 \times 10^{-3} H$$

Pour 5Khz :

$$L_{5kHz} := \tan(78°) \cdot \frac{145.54\Omega}{2 \cdot \pi \cdot f_{5kHz}} = 21.795 \times 10^{-3} H$$

Même raisonnement pour un courant dans la charge de 40mA :

$$\left|Z_{load}\right| = \blacksquare \cdot 175 \qquad \Rightarrow \qquad R := \sqrt{\frac{175^2}{1 + (\tan(78°))^2}} = 36.385$$

Pour 2kHz :

$$L_{2kHz} := \tan(78°) \cdot \frac{36.385\Omega}{2 \cdot \pi \cdot f_{2kHz}} = 13.622 \times 10^{-3} H$$

Pour 5kHz :

$$L_{5kHz} := \tan(78°) \cdot \frac{36.385\Omega}{2 \cdot \pi \cdot f_{5kHz}} = 5.449 \times 10^{-3} H$$

Nous avons maintenant des valeurs concrètes pour la charge en fonction du courant, du facteur de puissance fixé et de la fréquence souhaitée. Bien sûr ici j'ai calculé les valeurs extrêmes, il y a autant de valeurs de charge que de courant possible.

Avant de passer à la partie suivante nous allons déterminer la fonction de transfert du filtre et enfin afficher des courbes du diagramme de bode, des sinus et les calculs de leur THD en sortie du filtre.

Calcul de la fonction de transfert du filtre

Dans un premier temps, nous allons trouver la fonction de transfert en utilisant une représentation utilisant des impédances. Ci-dessous une représentation simplifiée du filtre.

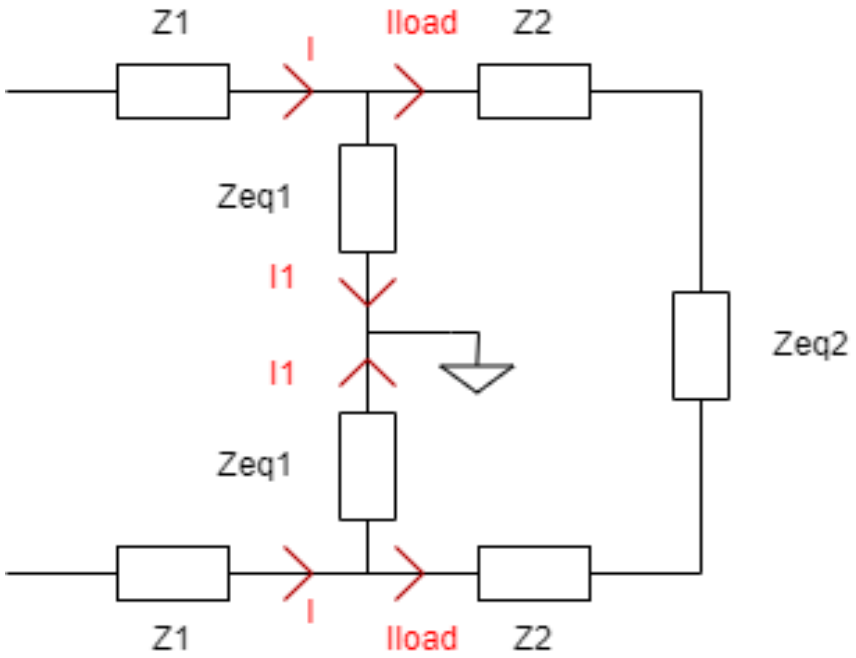


Figure 18 : Représentation simplifiée du filtre

Ou on a $Z_{eq2}(s) := \frac{Z_{Load}(s)\cdot 2\cdot Z_5(s)}{Z_{Load}(s) + 2\cdot Z_5(s)}$ avec Z5 = R2=3.3kΩ

On souhaite trouver $\frac{Vo(s)}{Vi(s)} = \blacksquare\cdot H(s)$ $V(s) := I_{Load}\cdot Z_{eq2}(s)$ $I_{Load} := \frac{Vo(s)}{Z_{eq2}(s)}$

$Vi(s) := 2\cdot Z_1(s)\cdot I + 2\cdot Z_2(s)\cdot I_{Load} + I_{Load}\cdot Z_{eq2}(s)$ avec $I := I_1 + I_{Load}$

$Vi(s) := 2\cdot Z_1(s)\cdot\left(I_1 + I_{Load}\right) + 2\cdot Z_2(s)\cdot I_{Load} + I_{Load}\cdot Z_{eq2}(s)$

Il faut maintenant exprimer les courants en fonction de Vo afin de trouver la fonction de transfert

$$I_1 := \frac{V_{int}(s)}{2 Z_{eq1}(s)}$$

Le pont diviseur de tension suivant peut aussi nous aider

$$Vo(s) := \frac{Z_{eq2}(s)}{Z_{eq2}(s) + 2\cdot Z_2(s)}\cdot V_{int}(s) \qquad V_{int}(s) := \frac{Z_{eq2}(s) + 2\cdot Z_2(s)}{Z_{eq2}(s)}\cdot Vo(s)$$

On injecte et on a :

$$I_1 := \frac{Z_{eq2}(s) + 2\cdot Z_2(s)}{2\cdot Z_{eq1}(s)\, Z_{eq2}(s)}\cdot Vo(s)$$

Nous avons donc :

$$Vi(s) := 2\cdot Z_1(s)\cdot\left(\frac{Z_{eq2}(s) + 2\cdot Z_2(s)}{2\cdot Z_{eq1}(s)\, Z_{eq2}(s)}\cdot Vo(s) + \frac{Vo(s)}{Z_{eq2}(s)}\right) + \frac{2\cdot Z_2(s)\cdot Vo(s)}{Z_{eq2}(s)} + Vo(s)$$

Finalement on peut diviser le tout par Vo pour obtenir :

$$\frac{Vi(s)}{Vo(s)} := 2\cdot Z_1(s)\cdot\left(\frac{Z_{eq2}(s) + 2\cdot Z_2(s)}{2\cdot Z_{eq1}(s)\, Z_{eq2}(s)} + \frac{1}{Z_{eq2}(s)}\right) + \frac{2\cdot Z_2(s)}{Z_{eq2}(s)} + 1 = \blacksquare\cdot H(s)$$

J'ai appelé cette fonction H, il suffit de prendre son inverse pour obtenir la fonction de transfert du filtre :

$$F(s) := \frac{1}{H(s)}$$

À l'aide de Mathcad, le logiciel qui m'a permis de mener tous mes calculs, j'ai obtenu cette application numérique du filtre avec les valeurs réelles des composants.

$$F(s) := \frac{1}{\left[2\cdot Z_1(s)\cdot\left(\frac{Z_{eq2}(s) + 2\cdot Z_2(s)}{2\cdot Z_{eq1}(s)\, Z_{eq2}(s)} + \frac{1}{Z_{eq2}(s)}\right) + \frac{2\cdot Z_2(s)}{Z_{eq2}(s)} + 1\right]} \left|\begin{array}{l}\text{simplifier}\\ \text{développer}\end{array}\right. \rightarrow \frac{10\times 10^{9}\cdot s^{3} - 0\cdot s^{4} + 4.612\times 10^{15}\cdot s^{2} + 30.342\times 10^{18}\cdot s + 15.259\times 10^{-6}}{s^{5} + 561.726\times 10^{3}\cdot s^{4} + 14.566\times 10^{9}\cdot s^{3} + 4.918\times 10^{15}\cdot s^{2} + 33.72\times 10^{18}\cdot s + 1.118\times 10^{24}}\blacksquare$$

Diagramme de bode

Ci-dessous on peut observer les diagrammes de bode associés aux deux fréquences extrêmes pour un courant de 10mA et 40mA. Ces diagrammes m'ont été fortement utiles afin de faire des tests de valeurs sur LTspice et observer les conséquences sur l'allure du filtrage.

D'abord pour 10mA :

Sinus de fréquence 2kHz :

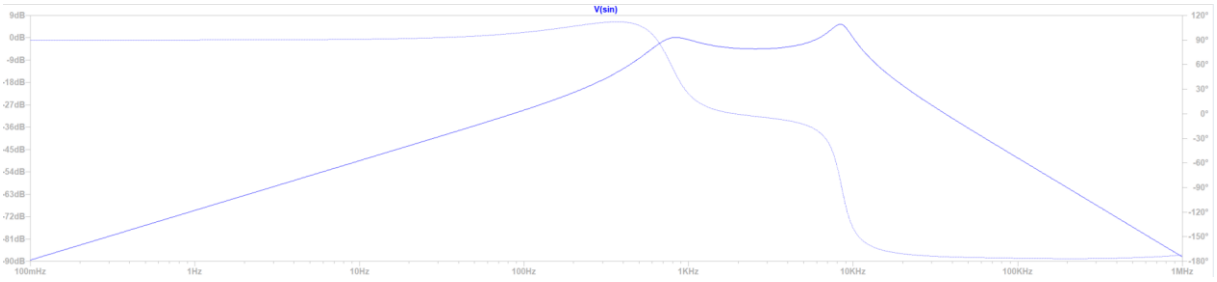


Figure 19 : Rload=145.5Ω et Lload=54.5mH

Sinus de fréquence 5kHz :

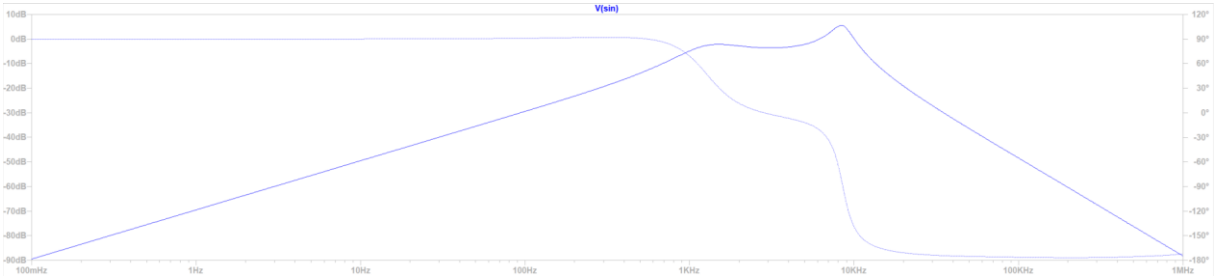


Figure 20 : Rload=145.5Ω et Lload=21.8mH

Et pour un courant de 40mA :

Sinus de fréquence 2kHz :

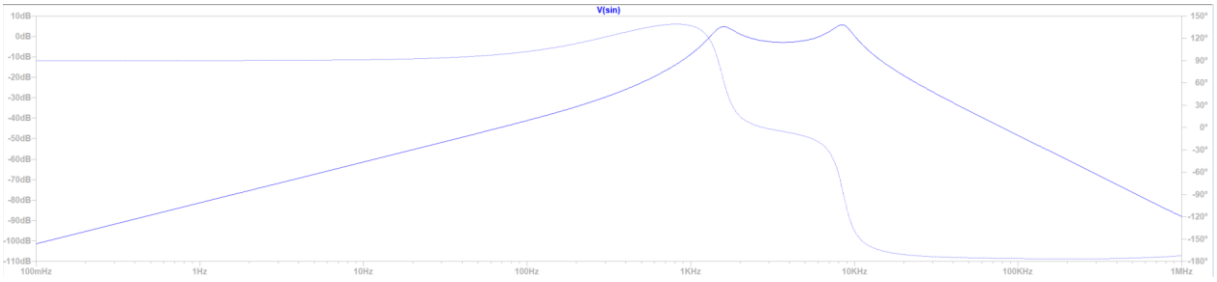


Figure 21 : Rload=36.38Ω et Lload=13.6mH

Sinus de fréquence 5kHz :

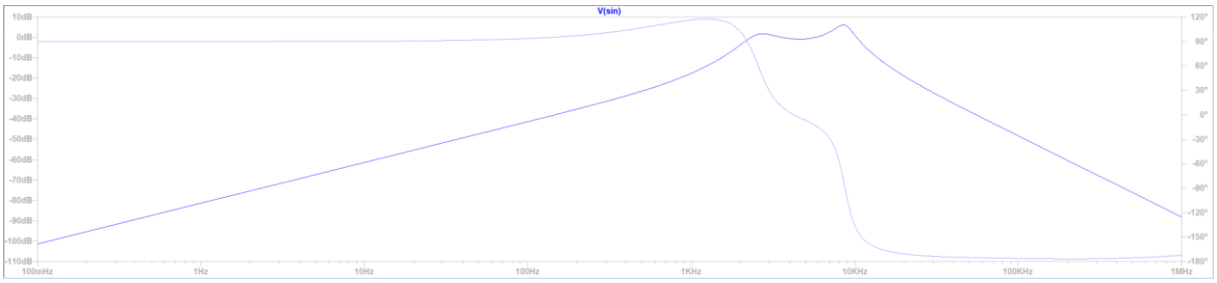


Figure 22 : Rload=36.38Ω et Lload=5.5mH

J'ai aussi tracé le diagramme avec MATLAB, et à partir de l'application numérique de la fonction de transfert donnée à la fin de là sous partie précédente. Pour cela j'ai utilisé les mêmes valeurs de charges que la dernière figure, soit R=36.38Ω et L=5.5mH, et j'obtiens :

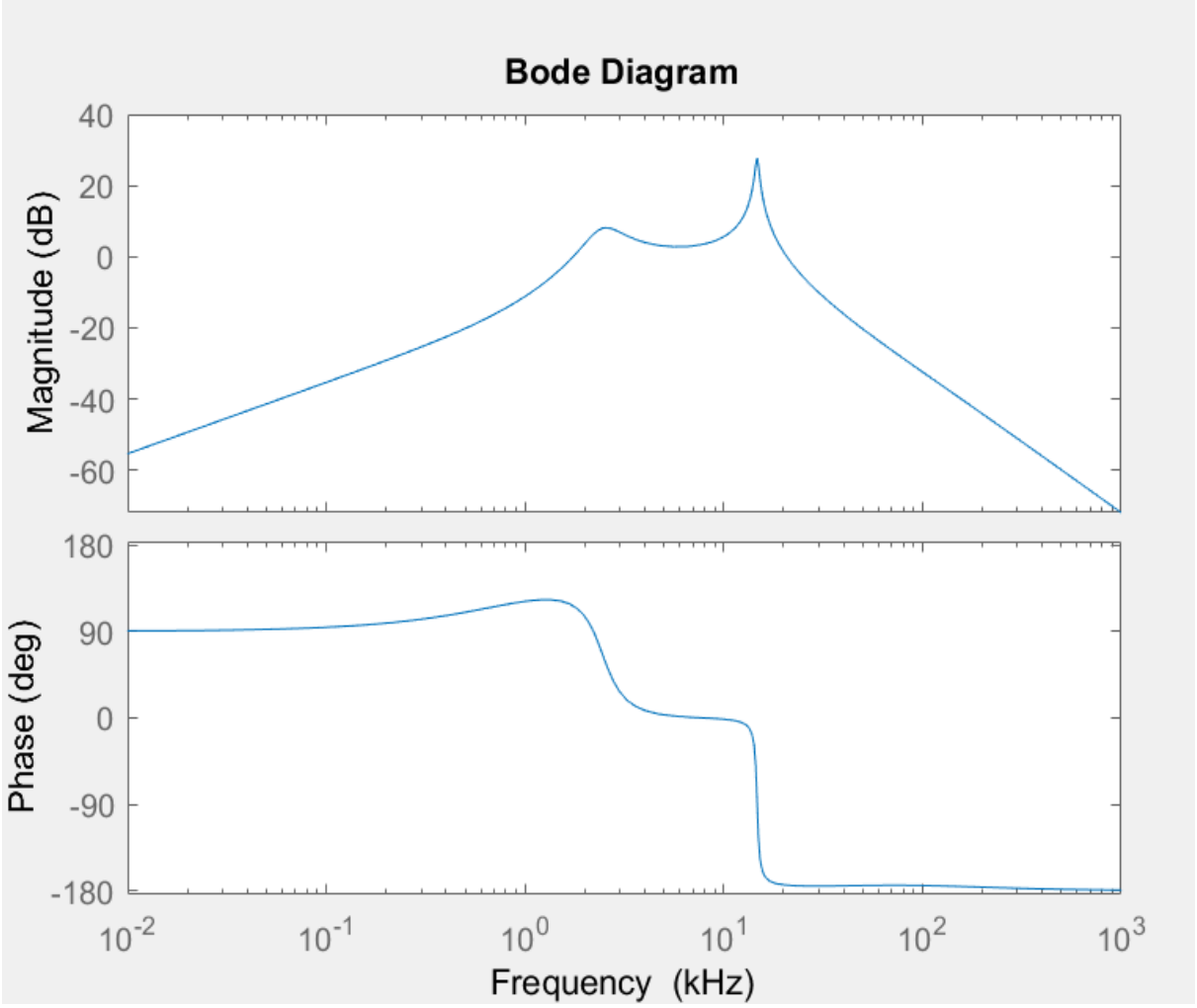


Figure 23 : Diagramme de bode du filtre théorique tracé avec MATLAB

On constate que les diagrammes sont similaires à peu de chose près ce qui veut dire que le calcul de la fonction de transfert du filtre est correct. Nous pouvons passer à la sous-partie suivante qui traite du calcul du THD.

Calcul du THD sur LTspice

LTspice offre la possibilité via une commande, de calculer le THD d'un signal, il suffit de lui indiquer le signal par un label et de lui donner la fréquence fondamentale souhaitée pour ce signal. Plus bas on peut trouver les simulations LTspice représentant les sinus appliqués aux bornes de la charge pour un sinus de 2kHz et 5kHz. De plus on peut aussi trouver une copie écran de la fenêtre LTspice qui indique le THD pour chacune des fréquences. Bien sûr, j'ai lancé la simulation sur une durée assez longue pour m'assurer d'avoir assez d'échantillons et ne pas fausser le calcul du THD. De plus, LTspice n'est pas capable de calculer le THD du signal appliqué aux bornes de la charge directement mais il peut calculer celui des sinus de phases opposées qui arrivent à chacune des bornes, et qui donne une très bonne idée du THD du signal différentiel. C'est pour cela que deux THD sont donnés pour chaque fréquence.

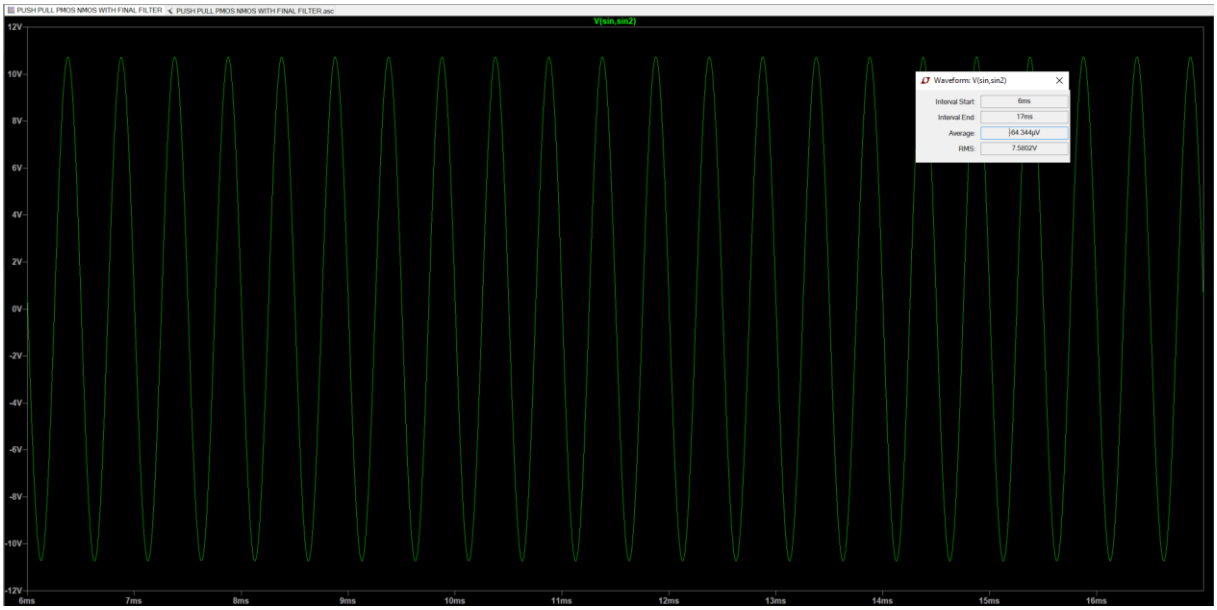


Figure 24 : Sinus de fréquence 2kHz obtenu par simulation LTspice

16	3.200e+04	5.598e-05	1.218e-05
17	3.400e+04	8.094e-05	1.761e-05
18	3.600e+04	4.455e-05	9.691e-06
19	3.800e+04	5.863e-05	1.275e-05
20	4.000e+04	4.772e-05	1.038e-05

Total Harmonic Distortion: 0.184328%(3.778811%)

N-Period=all
Fourier components of V(sin2)
DC component:-0.00258494

Harmonic Number	Frequency [Hz]	Fourier Component	Normalized Component
1	2.000e+03	4.597e+00	1.000e+00
2	4.000e+03	8.346e-04	1.815e-04
3	6.000e+03	6.210e-03	1.351e-03
4	8.000e+03	1.898e-03	4.129e-04
5	1.000e+04	1.613e-03	3.508e-04
6	1.200e+04	5.002e-04	1.088e-04
7	1.400e+04	5.483e-04	1.193e-04
8	1.600e+04	2.169e-04	4.719e-05
9	1.800e+04	4.323e-04	9.403e-05
10	2.000e+04	1.492e-04	3.246e-05
11	2.200e+04	1.478e-04	3.215e-05
12	2.400e+04	4.326e-05	9.411e-06
13	2.600e+04	1.324e-04	2.879e-05
14	2.800e+04	1.321e-04	2.874e-05
15	3.000e+04	1.265e-04	2.752e-05
16	3.200e+04	1.026e-04	2.231e-05
17	3.400e+04	7.187e-05	1.563e-05
18	3.600e+04	8.091e-05	1.760e-05
19	3.800e+04	4.578e-05	9.959e-06
20	4.000e+04	5.229e-05	1.137e-05

Total Harmonic Distortion: 0.148140%(3.506214%)

Figure 25 : copie écran LTspice du calcul du THD à 2kHz

On peut observer un THD de 0.18% et 0.14%, ce qui est largement acceptable par rapport aux 1% fixés dans le cahier des charges.

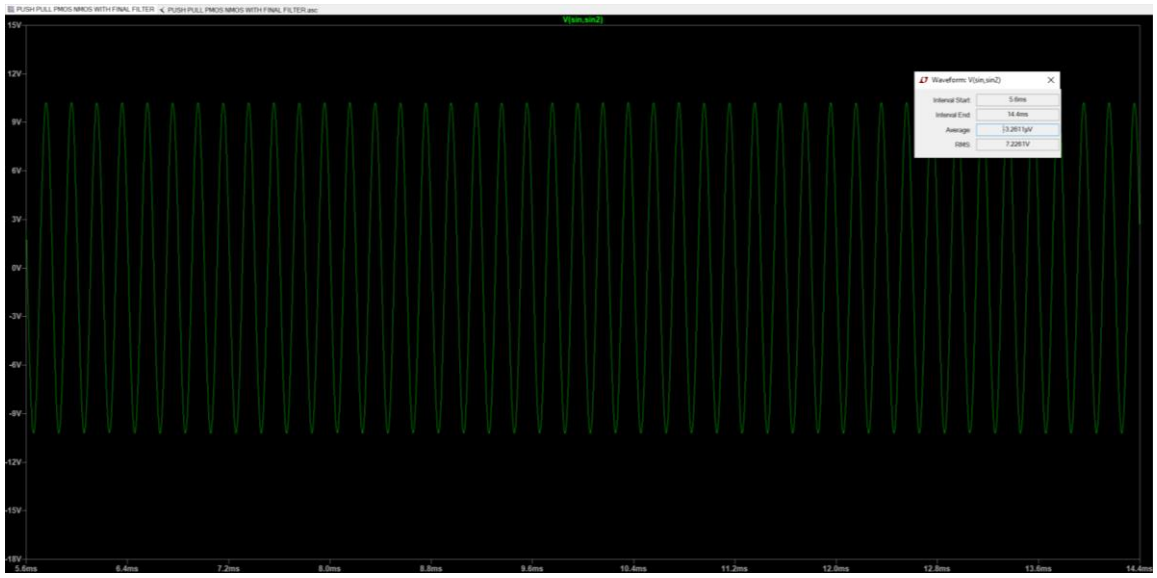
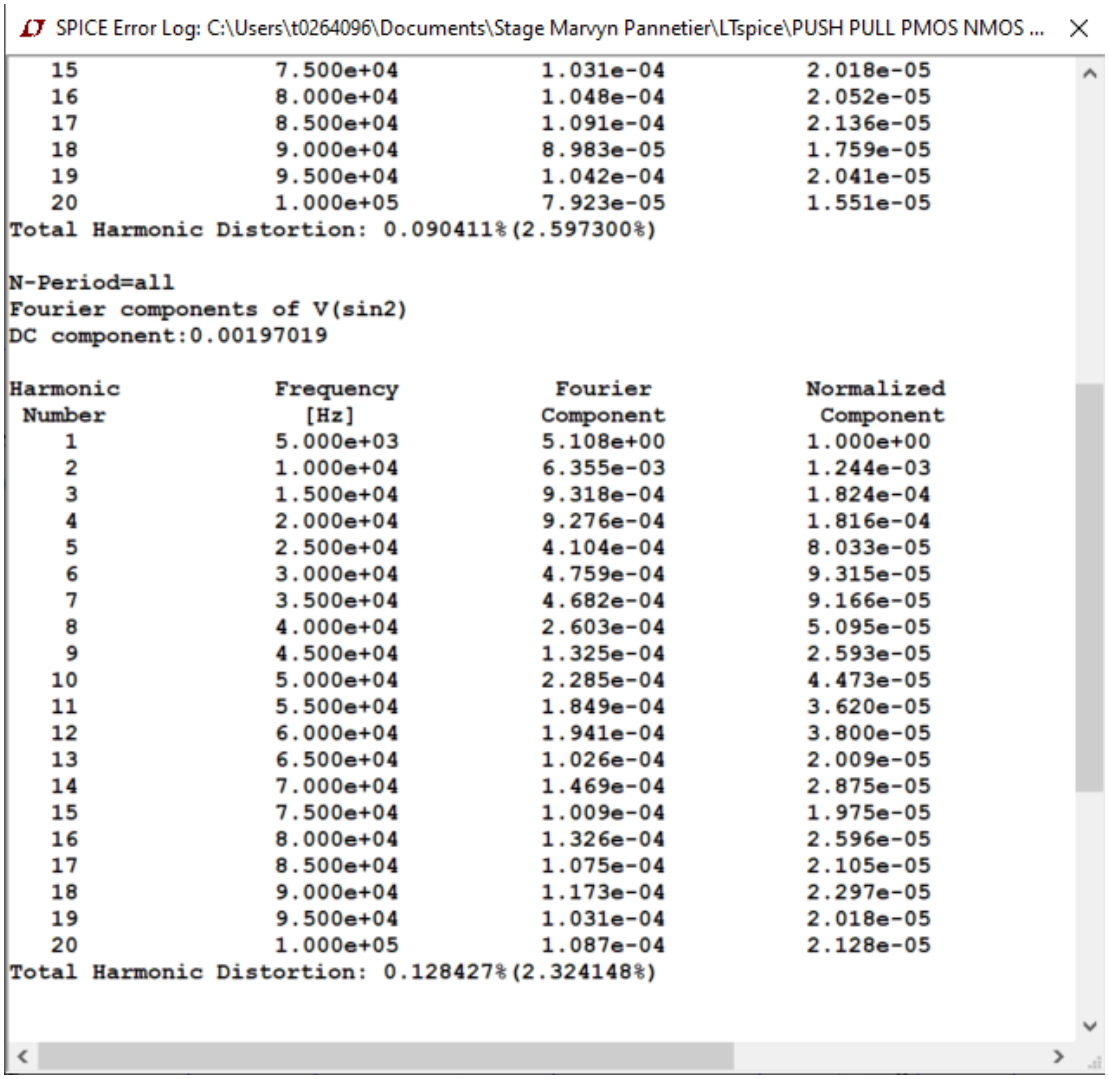


Figure 26 : Sinus de fréquence 5kHz obtenu par simulation LTspice




```
   15               7.500e+04           1.031e-04           2.018e-05
   16               8.000e+04           1.048e-04           2.052e-05
   17               8.500e+04           1.091e-04           2.136e-05
   18               9.000e+04           8.983e-05           1.759e-05
   19               9.500e+04           1.042e-04           2.041e-05
   20               1.000e+05           7.923e-05           1.551e-05
Total Harmonic Distortion: 0.090411%(2.597300%)

N-Period=all
Fourier components of V(sin2)
DC component:0.00197019

Harmonic            Frequency            Fourier            Normalized
 Number               [Hz]              Component             Component
    1               5.000e+03           5.108e+00           1.000e+00
    2               1.000e+04           6.355e-03           1.244e-03
    3               1.500e+04           9.318e-04           1.824e-04
    4               2.000e+04           9.276e-04           1.816e-04
    5               2.500e+04           4.104e-04           8.033e-05
    6               3.000e+04           4.759e-04           9.315e-05
    7               3.500e+04           4.682e-04           9.166e-05
    8               4.000e+04           2.603e-04           5.095e-05
    9               4.500e+04           1.325e-04           2.593e-05
   10               5.000e+04           2.285e-04           4.473e-05
   11               5.500e+04           1.849e-04           3.620e-05
   12               6.000e+04           1.941e-04           3.800e-05
   13               6.500e+04           1.026e-04           2.009e-05
   14               7.000e+04           1.469e-04           2.875e-05
   15               7.500e+04           1.009e-04           1.975e-05
   16               8.000e+04           1.326e-04           2.596e-05
   17               8.500e+04           1.075e-04           2.105e-05
   18               9.000e+04           1.173e-04           2.297e-05
   19               9.500e+04           1.031e-04           2.018e-05
   20               1.000e+05           1.087e-04           2.128e-05
Total Harmonic Distortion: 0.128427%(2.324148%)
```

Figure 27 : copie écran LTspice du calcul du THD à 5kHz

Cette fois-ci on a un THD de 0.09% et 0.12% ce qui est tout autant acceptable vis-à-vis du cahier des charges.

Une dernière chose concernant le filtrage, une inductance de mode commun appelée inductance choke a été ajouté plus tard. Le circuit fonctionnant avec de la commutation ce qui crée des perturbations électromagnétiques et la charge peut avoir des courants de fuites ou des couplages à la masse mécanique qu'il faut filtrer pour tenir les gabarits d'émission conduite. On peut observer comment a été inséré ce composant dans le schéma dans l'annexe 1 et 2. Sur le schéma on peut aussi voir que je peux choisir de court-circuiter ou non ce composant afin de faire des tests avec et sans. Le but étant si possible de faire des mesures en CEM afin de quantifier l'impact de l'inductance choke sur les émissions conduites.

À ce point le filtre n'a plus de secret pour nous, il est temps de s'intéresser à une partie très importante qui est celle de la consommation, des pertes et du rendement.

1.5.3. Calcul théorique des pertes et du rendement

Dans cette partie, le but est d'estimer les pertes ainsi que le rendement du circuit développé. Bien sûr les deux sont liés et c'est pourquoi nous allons commencer par définir le rendement. Le rendement correspond au rapport de la puissance utile sur la puissance consommée ou encore de la puissance de sortie sur la puissance d'entrée

$$\eta := \frac{P_{OUT}}{P_{IN}} \qquad \text{ou} \qquad \eta := \frac{\text{Useful\_power\_output}}{\text{Total\_power\_input}}$$

Sur la figure ci-dessous sont encadrés en rouge les composants qui vont engendrer des pertes, il y a en plus les pertes dans les MOSFET ainsi que celles dans les drivers. Le rectangle vert lui identifie la résistance de la charge qui va aussi dissiper de la puissance.

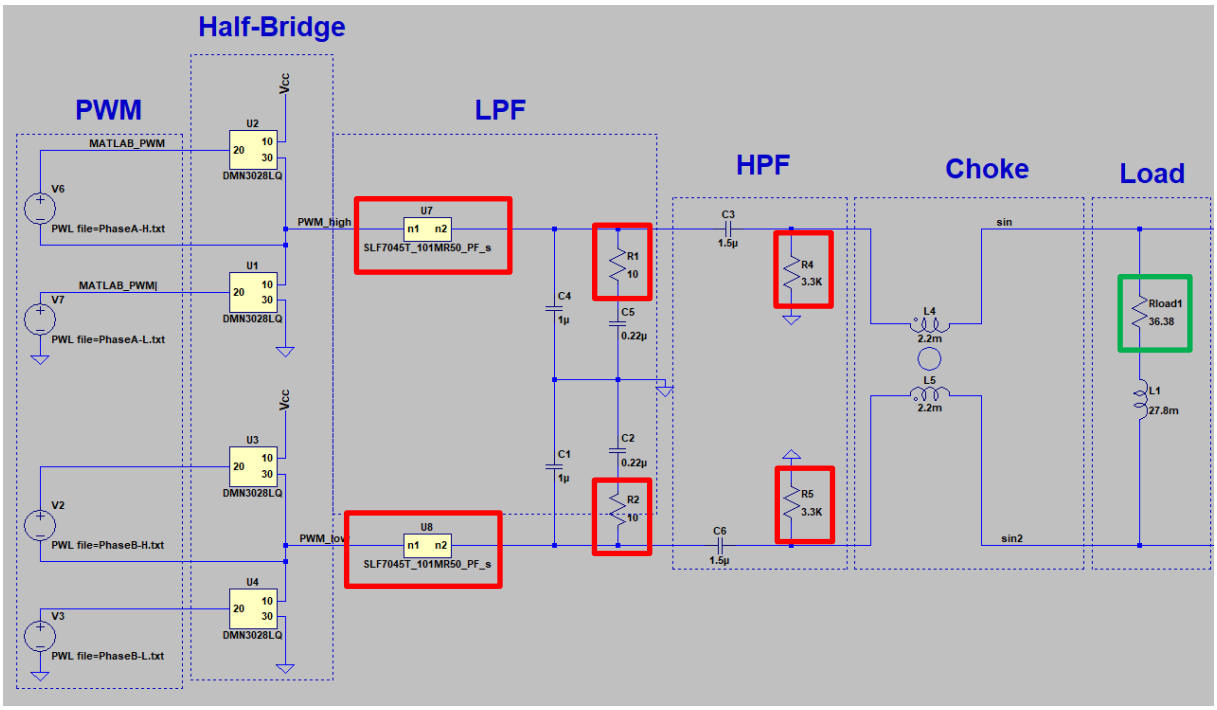


Figure 28 : Identification des pertes sur le schéma LTspice

J'ai ensuite défini chacune de ces puissances afin d'arriver à une forme plus précise du rendement :

$$\frac{P_L}{P_{MOSFET} + P_{snubber} + P_{inductor} + P_{3.3k} + P_L + P_{driver} + P_{ripple}}$$

On peut donc voir que la puissance de sortie correspond à la puissance dans la charge et la puissance totale consommée est l'addition des puissances qu'on peut voir ci-dessus. Je vais ensuite décrire chacune de ces puissances, mais avant cela il faut savoir que j'ai rédigé un

document sur Mathcad, qui est très complet et reprend toutes les étapes et preuve de chaque calcul de rendement. Pour des raisons de place et d'intérêt pour le rapport je mettrai en annexe des copies écran de ce document et je vais uniquement sélectionner les points les plus importants dans cette partie.

Commençons par PL, la puissance dans la charge. On a les équations suivantes :

$$P_L := R_L \cdot I_{RMS}^2 \qquad P_L = \frac{Vo^2}{|Z_L|} \cdot \cos(\varphi) \qquad \text{avec PL = Pout}$$

Avec RL et ZL liés bien sûr, on a vu plus haut que pour ZL = 700Ω on a RL = 145.5Ω

Ensuite on a Pdriver : $P_{driver} := f_{SW} \cdot Cg \cdot Vcc^2$

Avec Cg la capacité de grille des MOSFET à commander, 500pF dans notre cas, Fsw la fréquence de la PWM et Vcc la tension d'alimentation du driver. Ce qui donne pour une fréquence de 300kHz, Pdriver=9.6mW.

Pour Psnubber j'ai cherché à estimer sa valeur le plus précisément possible. Pour ça j'ai d'abord calculé la résistance équivalente du snubber, puis la tension RMS aux bornes du snubber pour en déduire le courant et il ne reste plus qu'à multiplier la résistance du snubber par le courant au carré.

D'abord le calcul de l'impédance

$$Z_{eq} := \frac{1 + R_2 \cdot C_2 \cdot w \cdot j}{C_2 \cdot w \cdot j} \qquad Z_{eq.norm} := \frac{\sqrt{1 + (R_2 \cdot C_2 \cdot w)^2}}{C_2 \cdot w}$$

Ainsi, nous pouvons approximer le courant. Si on prend une tension d'alimentation de 8V, on peut supposer qu'il s'agit approximativement de l'amplitude maximale du sinus et en déduire par conséquent la valeur RMS :

$$V_{sin.RMS} := \frac{8V}{\sqrt{2}} \qquad I_{RMS} := \frac{V_{sin.RMS}}{Z_{eq.norm}}$$

Il est maintenant possible de calculer la puissance estimée des snubbers :

$$P_{snubber} := R_2 \cdot I_{RMS}^2 \qquad P_{2snubber} := 2 \cdot P = 7.424 \times 10^{-3} W$$

Une autre perte est due aux résistances de 3.3KΩ. On sait que nous voulons 7V RMS aux bornes de la charge, soit 3.5V RMS aux bornes de chaque résistance de 3.3KΩ. Le calcul est donc plus simple :

$$P := \frac{3.5^2 V^2}{3.3 \cdot 10^3 \Omega} = 3.712 \times 10^{-3} W$$

Sachant qu'il y a deux résistances, on a une puissance totale dissipée par les résistances de 7.4mW.

L'inductance est une source de pertes à ne pas négliger étant donné que son ESR peut être assez conséquente et qu'elle est traversée par le courant maximum en sortie des étages demi-ponts. J'ai donc cherché à estimer le courant dans l'inductance. Après réflexion, le courant dans l'inductance dépend essentiellement du courant dans la charge et du courant tiré par les condensateurs C1 à la masse. Ce courant peut être estimé de la manière suivante :

$$C := 1 \cdot 10^{-6} F \qquad \Delta u := 8V \qquad \Delta t := \frac{1}{2 \cdot 5 \cdot 10^3 Hz} \qquad I := C \cdot \frac{\Delta u}{\Delta t} = 0.08A$$

Si on se met dans le cas ou 40mA traverse la charge on aurait donc un courant dans les inductances de : Iinductances = 120mA soit 60mA par inductance due à la configuration différentielle. Il ne reste qu'à multiplier le courant au carré par la résistance ESR de l'inductance, qui dans notre cas vaut 250mΩ.

$$P_{inductor} := ESR \cdot I_{inductor}^2 = 9.321 \times 10^{-4} W$$

Soit environ 1mW.

Les dernières pertes à calculer sont dû aux MOSFET et aux ondulations (ripple). Les pertes dans les MOSFET sont divisées en deux, les pertes de conduction et de commutation, on remarque un facteur 4 dans la dernière formule étant donné qu'il y a 4 MOSFET dans un circuit ACS.

$$P_{Cond} = I_{mosfet}^2 R_{DSon} = 4 \times 10^{-4} W \qquad P_{SW} := \frac{1}{2} \cdot Vo \cdot I_{Load} \cdot \left(t_{c.on} + t_{c.off}\right) \cdot f_{SW}$$

$$P_{MOSFET} := 4\left(P_{SW} + P_{cond}\right) = 2.644 \times 10^{-3} W$$

Enfin les pertes dues au ripple s'expriment de cette façon :

$$I_{Ripple\_RMS} = \frac{Vo}{\sqrt{3}4 \cdot L \cdot f_{SW}} = 0.027A \qquad P_{Ripple} = 2I_{ripple\_RMS}{}^{2} \cdot \left(R_{DSon} + ESR\right) = 4.1 \times 10^{-4} W$$

Pour finir cette partie, on peut à présent calculer le rendement qui correspond au rendement dans les conditions suivantes. Icharge = 40mA, Fsinus = 5kHz et Fsw=300kHz.

$$\eta := \frac{P_L}{Pmosfets + P_{snubber} + P_{inductor} + P_{3.3k} + P_L + 2 \cdot P_{driver}} = 0.572$$

Soit un rendement de 57.2% ce qui est beaucoup plus que la fonction actuelle qui ne dépasse pas les 10% dans les meilleurs cas. De plus il est possible d'améliorer ce rendement en diminuant la fréquence de commutation des MOSFET, ce qui va cependant induire un plus grand ripple, il faut donc trouver le bon compromis sur le circuit réel.

Pour rappel, un document beaucoup plus détaillé sur les pertes se trouve en annexe et j'ai décidé de me concentrer uniquement sur les points-clés de ce circuit dans ce rapport.

Pour conclure cette partie, LTspice permet de faire des mesures de puissance dissipée à travers les composants, j'ai donc mesuré les pertes dans la charge puis dans chacun des composants entourés sur la Figure 28 plus celle dans les MOSFET et en y ajoutant les pertes théoriques des drivers, en respectant les mêmes conditions que dans la théorie et j'obtiens finalement un rendement de 58%, ce qui est assez proche du résultat malgré les approximations de la théorie.

La suite du rapport va aborder la génération des PWM ainsi que l'asservissement de la tension de sortie.

1.5.4. Implémentation de la génération des PWM

Pour générer les PWM, une carte NUCLEOH743ZI2 m'a été confiée, celle-ci permet de générer facilement des PWM avec les timers de base, allant jusqu'à 200kHz et en gardant une résolution de 10bits. Elle possède aussi un timer fonctionnant jusqu'à 480MHz et permettant de générer des PWM à plus de 300kHz tout en gardant une très bonne résolution.

La particularité ici va être de générer un PWM d'un le rapport cyclique est variable afin de représenter un sinus. Pour cela on utilise un tableau ayant des valeurs représentatives des amplitudes d'une sinusoïde, en plus d'un timer et de la DMA, afin de récupérer les valeurs dans la mémoire RAM et de les appliquer au registre influant sur le rapport cyclique de la PWM. On règle le timer permettant d'interroger la RAM afin de d'obtenir la fréquence du sinus voulu grâce à la formule suivante.

$$SinFreq = \frac{TriggerFreq}{Ns}$$

Avec TriggerFreq la fréquence à laquelle le timer va interroger la mémoire et NS le nombre de points utilisés pour représenter le sinus.

Une autre formule importante est celle de la résolution :

$$Resolution = \frac{\log\left(\frac{Fclk}{Fpwm}\right)}{\log(2)}[\text{Bits}]$$

Cette formule permet de calculer le nombre de bits de rendement en fonction de la fréquence de la PWM et de l'horloge. Ces fréquences se choisissent dans le deuxième timer chargé de la génération de la PWM. Sa fréquence d'horloge peut être réglée à 240MHz et on peut choisir une fréquence de PWM de 200kHz en jouant sur la variable ARR de la fonction suivante par exemple :

$$Fpwm = \frac{Fclk}{(ARR + 1) \times (PSC + 1)}$$

À partir de là je me suis fixé une résolution de 10bits minimum et j'ai fait les applications numériques pour différentes fréquences de PWM afin de réaliser des tests. La figure ci-dessous est une bonne représentation du fonctionnement décrit plus tôt.

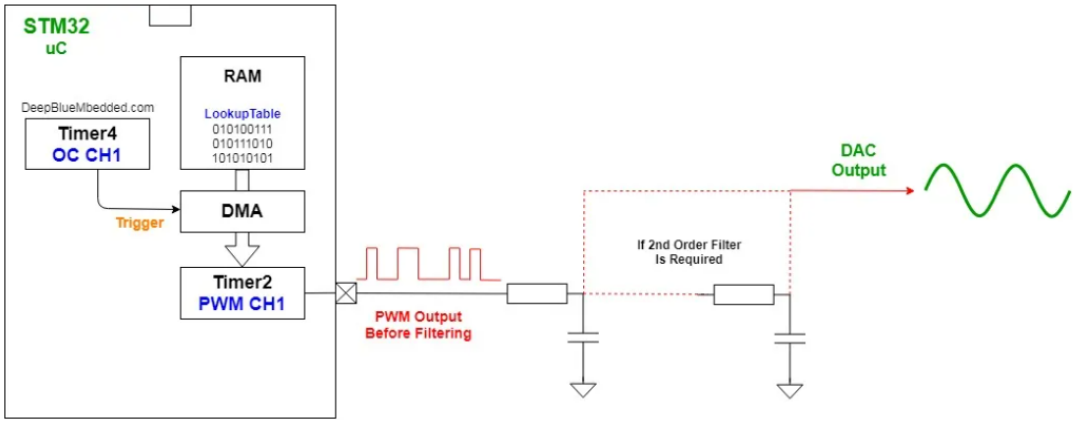


Figure 29 : Fonctionnement de la génération des PWM

Pour finir, il m'a suffi de répéter les manipulations en utilisant un deuxième tableau avec les mêmes valeurs de sinus mais déphasées de 180° afin de générer une deuxième PWM.

Ci-dessous une capture écran d'un PWM de fréquence 200Khz, généré par la carte nucleo.

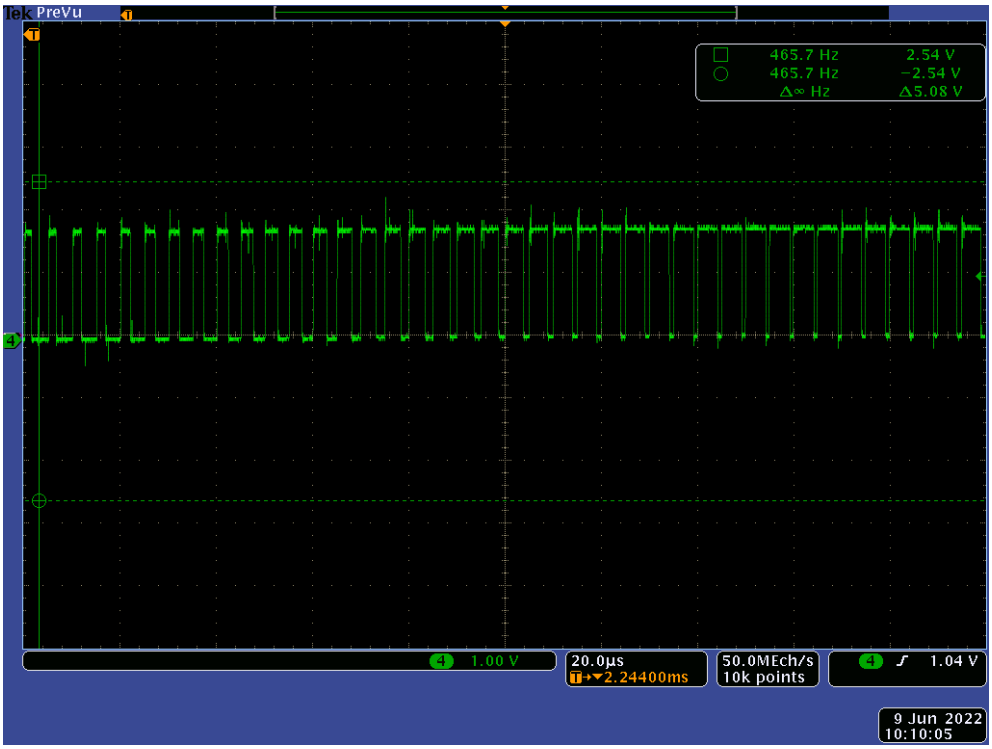


Figure 30 : SPWM générée à partir de la carte nucleo

Maintenant il est temps de s'intéresser à la partie plus automatique de mon stage, l'asservissement de la tension de sortie.

1.5.5. Asservissement de la tension de sortie

Un des buts du projet est de garantir la stabilité de la tension de sortie mais aussi et surtout d'atteindre la valeur fixée par la commande en modifiant l'indice de modulation des PWM. En effet, en fonction de la fréquence du sinus, pour une même tension d'alimentation, l'amplitude du signal de sortie va changer. Ce phénomène est dû aux atténuations ou amplifications, eux-mêmes dues aux phénomènes de résonnance introduit par le filtre en fonction de la fréquence du signal à filtrer. Ce qui veut dire qu'il va falloir se placer dans le cas atténuant le plus afin de fixer une valeur d'alimentation suffisamment grande, puis il faudra corriger les amplifications par rapport à ce cas en diminuant l'indice de modulation.

La première étape est de mesurer la sortie, ensuite il faut appliquer un correcteur qui va venir corriger la commande et donc la sortie. Afin de simuler ce système j'ai pu utiliser MATLAB et Simulink, ci-dessous on peut voir le système Simulink conçu pendant le stage.

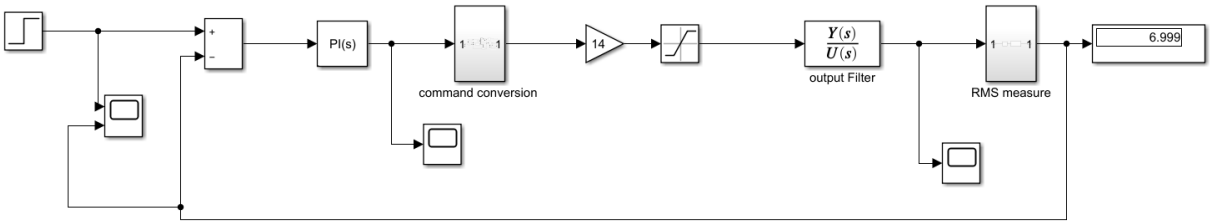


Figure 31 : Modèle de simulation Simulink de l'asservissement de la sortie

La première chose que j'ai faite a été de créer un bloc contenant la fonction de transfert du circuit., c'est-à-dire la fonction de transfert du filtre plus un gain dû à l'étage d'amplification. Ensuite j'ai créé la boucle et la commande puis j'ai inséré le correcteur PI. Enfin, j'ai ajouté un bloc permettant la conversion de la mesure en une commande. En effet une des problématiques de ce système et que la mesure correspond à la valeur RMS de la tension mais que le paramètre sur lequel il faut jouer pour changer la commande est l'indice de modulation de la PWM, autrement dit la plage sur laquelle le rapport cyclique varie. Par exemple de 3% à 97% de la période ou encore 10% à 90% de la période. Cela peut être aussie de passer de 5%-100% à 20%-100%, ce qui importe c'est que la plage sur laquelle le rapport cyclique varie diminue pour diminuer l'amplitude du sinus après filtrage, et inversement pour augmenter l'amplitude. C'est pour cela qu'il a fallu créer se bloque de conversion dont le contenu est le suivant :

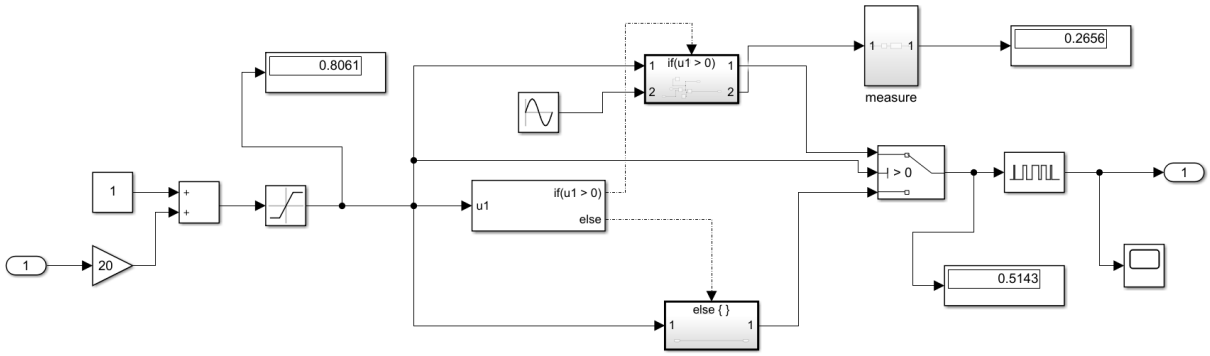


Figure 32 : Contenu du bloc de conversion de la mesure en commande

Il est un peu complexe à première vue. Ce bloc récupère la différence entre la valeur RMS souhaitée et la valeur mesurée à la sortie, puis il multiplie ou divise cette valeur afin de modifier l'indice de modulation en conséquence. Ce bloc est un peu complexe car il n'est pas possible de changer l'indice de modulation directement, j'ai dû jouer sur l'amplitude d'une sinusoïde qui influe sur les rapports cycliques de la PWM, c'est une contrainte uniquement due à la simulation sur Simulink puisqu'avec le correcteur qui sera implémenté sur la carte nucleo il suffira de multiplier les valeurs du tableau contenant les valeurs du sinus pour modifier l'indice de modulation.

Concernant le dimensionnement du correcteur j'ai utilisé la méthode de Ziegler Nichols pour trouver les valeurs du correcteur PI, en mesurant le gain maximal Ku=0.4 et la période d'oscillation du signal Tu=600µs et en appliquant les formules de cette méthode. Au final je trouve un Kp=0.18 et Ki=360.

Finalement j'ai pu lancer des simulations en testant avec une commande de 5V à 8V. Les résultats sont satisfaisants et corrige bien la sortie de façon précise. Sur la Figure 27 on peut d'ailleurs voir dans l'afficheur de droite la valeur RMS de la sortie pour une commande de 7V, qui est de 6.999V, soit 1mV d'écart.

Afin de réaliser l'asservissement sur le PCB réel il a fallu concevoir un circuit. Ma première idée était d'utiliser un amplificateur d'instrumentation avec entrée différentielle, qui permettrait d'adapter la tension aux caractéristiques d'entrée de la carte nucleo. Seulement, il s'est avéré compliqué de trouver un amplificateur avec des caractéristiques satisfaisantes, sans que son prix soit excessif. J'ai donc choisi un amplificateur opérationnel avec lequel j'ai fait un circuit soustracteur simple avec un offset car la carte nucleo n'accepte pas les tensions négatives. Le résultat est forcément moins précis que celui d'un amplificateur d'instrumentation mais je n'ai pas besoin d'une valeur très précise pour voir si j'arrive à mettre en place la régulation. Dans la réalité, il y a un circuit déjà développé par THALES pour faire cette mesure de façon très précise sur les cartes finales.

1.5.6. Schématique et routage du PCB

Après la théorie et les simulations finies, la suite logique des choses est de fabriquer la carte électronique afin de faire des tests réels. Pour cela il a fallu faire la schématique et le routage pour pouvoir générer les fichiers gerber nécessaires à la fabrication d'un PCB. De plus, j'ai réalisé mon PCB dans le FABLAB de THALES directement, qui est équipé d'une technologie permettant de réaliser des PCB à deux couches uniquement et avec des largeurs des pistes et d'isolation de 0.5mm au minimum. J'ai donc fait le schéma et le routage avec le logiciel Kicad EDA en respectant les contraintes de fabrication du FABLAB.

Avant de réaliser la schématique, j'ai sélectionné les composants afin d'importer directement les bon symboles et empreintes. Pour choisir les composants je me suis basé sur les critères suivants :

- Performance
- Taille
- Prix
- Disponibilité

Les composants passifs ont été relativement simples à trouver mais en ce qui concerne les driver de MOSFET et les MOSFET par exemple j'ai sélectionné 3 composants et j'ai ensuite fait des tableaux comparatifs afin de déterminer lequel des 3 composants est le meilleur.

Une fois les composants sélectionnés, j'ai pu faire la schématique et m'attaquer au routage. J'ai inséré beaucoup de points de tests et de jumper afin de pouvoir observer les signaux qui m'intéressent, de faire des mesures de courant et de déconnecter des parties du circuit afin de faire des tests plus spécifiques. Concernant le routage, j'ai eu une contrainte sur la taille des pistes d'au moins 0.5mm mais aussi sur le nombre de vias car leur technologie disponible ne sait pas métalliser les vias, ce qui m'a donc obligé à faire le lien entre les deux couches du PCB moi-même et ceux-ci pour chaque via.

Enfin, j'ai porté une attention plus particulière sur les étages d'amplification, la commutation créant des pics et variations de courant et tension autour du nœud appelé switching node assez conséquentes, j'ai privilégié les plans à la place des pistes pour cette partie du circuit. Je suis conscient que ce PCB pourrait être amélioré en le routant sur 4 couches, en faisant des retours à la masse plus intelligents, ou en optimisant encore plus le routage lié aux étages d'amplification. Pour plus de détails, le schéma, le routage et la BOM sont présents en annexe du rapport.

1.5.7. Tests et mesures sur le PCB

Avant tout, ce rapport est fini d'être rédigé début juin, il me reste donc un mois et demi de stage durant lesquels je vais essentiellement me concentrer sur les tests, mesures et l'amélioration du PCB actuel. J'ai cependant fait en sorte d'avoir fait quelques mesures afin d'exposer les premiers résultats dans ce rapport.

Pour commencer, voici à quoi ressemble la carte après assemblage

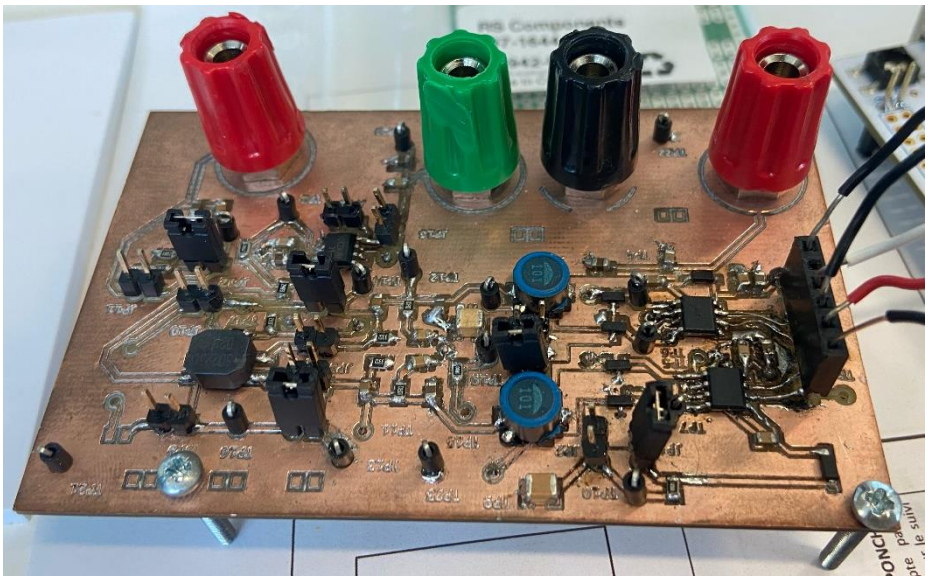


Figure 33 : Vue de la face supérieur du PCB

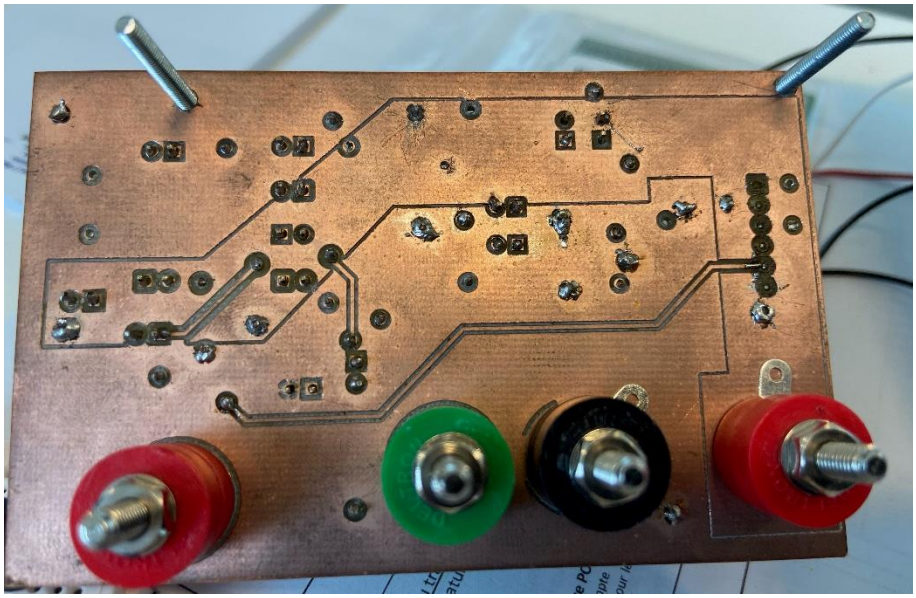


Figure 34 : Vue de la face inférieur du PCB

Une fois le PCB assemblé j'ai pu envoyer les PWM avec la nucleo, alimenter la carte et faire mes premières observations à l'oscilloscope.

La première chose que j'ai vérifiée est l'allure des PWM et le résultat est plutôt satisfaisant puisque toutes les PWM sont propres, avant et après les drivers. Ci-dessous une copie écran de l'allure d'une des PWM.

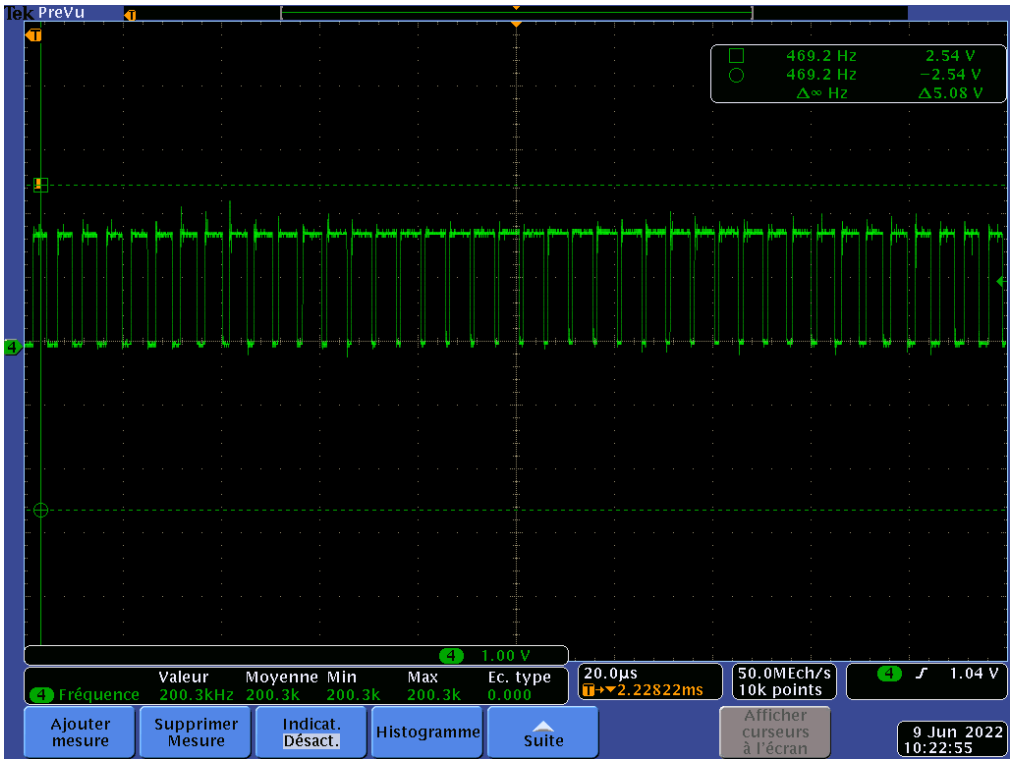


Figure 35 : Allure d'une des PWM sur le PCB

J'ai aussi vérifié que la fréquence était la bonne, ce qui est bon puisqu'on peut lire 200kHz sur la figure ci-dessus, qui est la valeur que j'ai programmée dans le code. La prochaine étape est de regarder l'allure des signaux de sortie, qui est la suivante.

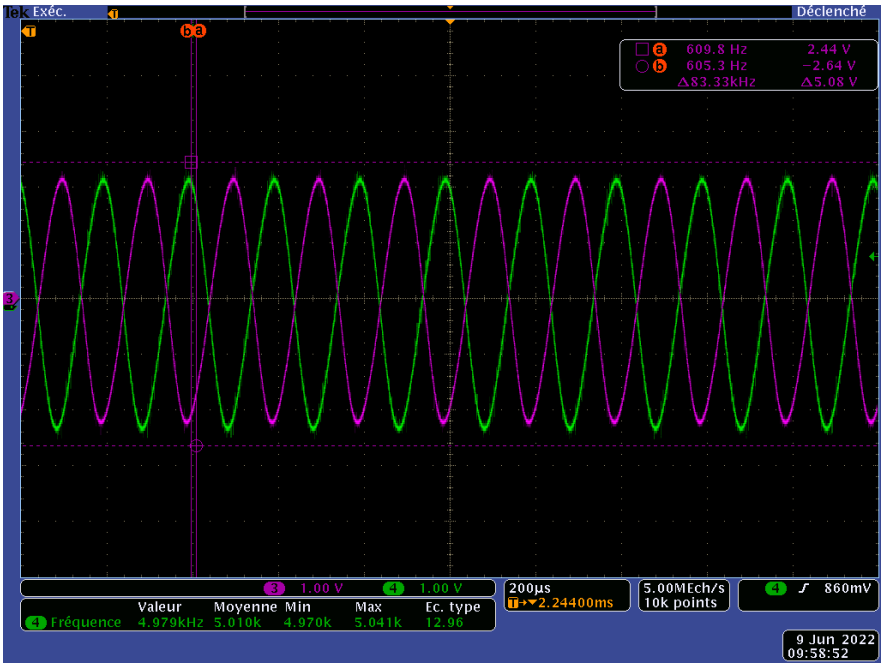


Figure 36 : Allure des sinusoïdes en sortie

J'ai ensuite fait la différence des deux pour observer le signal différentiel aux bornes de la charge. Ces sinus sont programmés pour avoir une fréquence de 5kHz et on peut voir sur la figure suivante que c'est bien le cas.

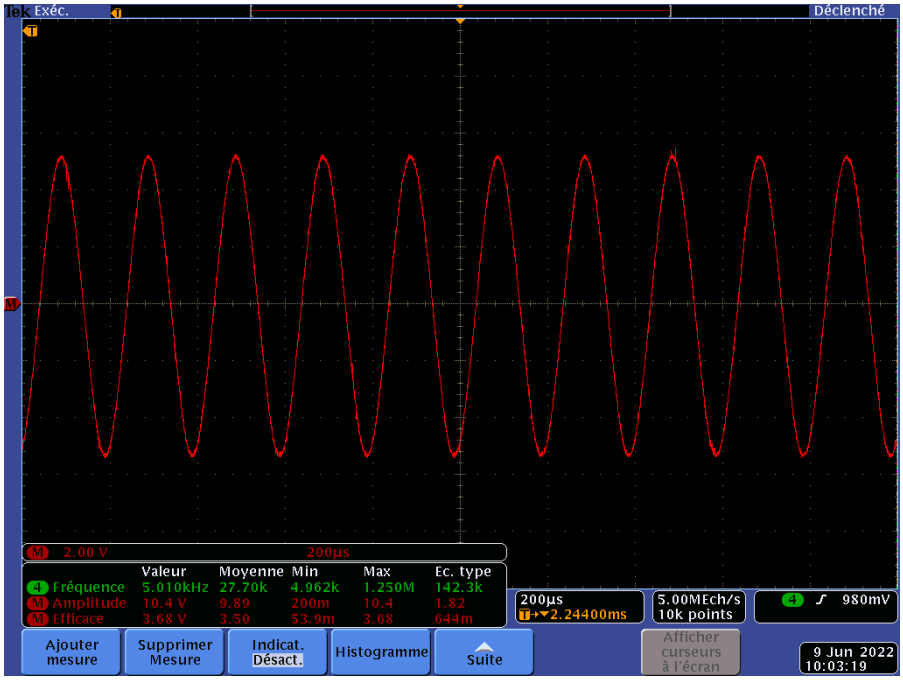


Figure 37 : Sinus différentiel à 5kHz

La même manipulation a été faite pour un sinus à 2kHz et on obtient ce résultat pour le sinus différentiel :

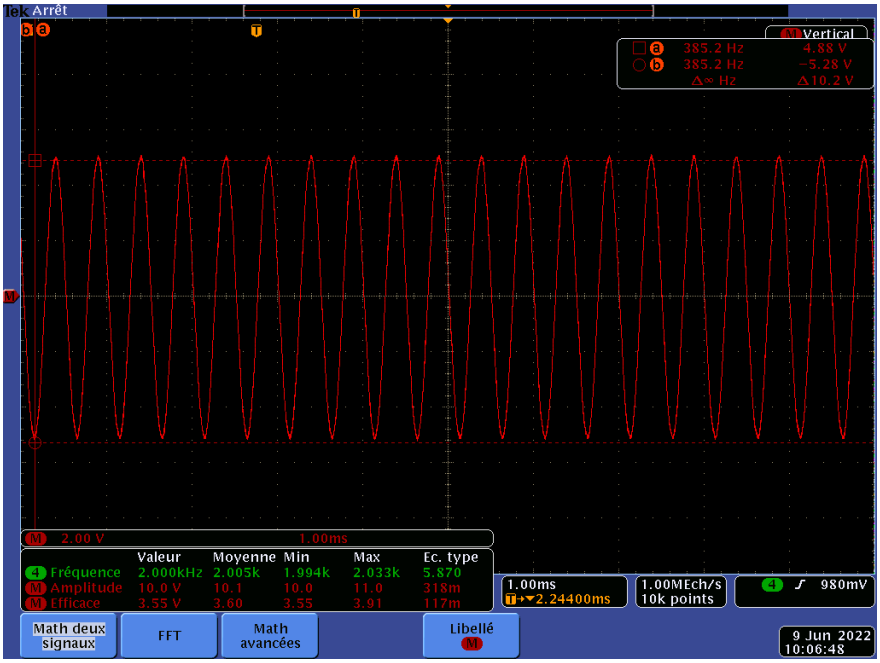


Figure 38 : Sinus différentiel à 2kHz

On peut aussi remarquer que l'amplitude des sinus différentiels est d'environ 10V soit 7.07V RMS. Bien sûr, j'ai utilisé une carte de prototypage pour changer la valeur de la charge en fonction des fréquences et travailler avec quelque chose de cohérent.

L'oscilloscope permet aussi d'afficher la FFT d'un signal, ce que j'ai donc fait pour les deux signaux précédents.

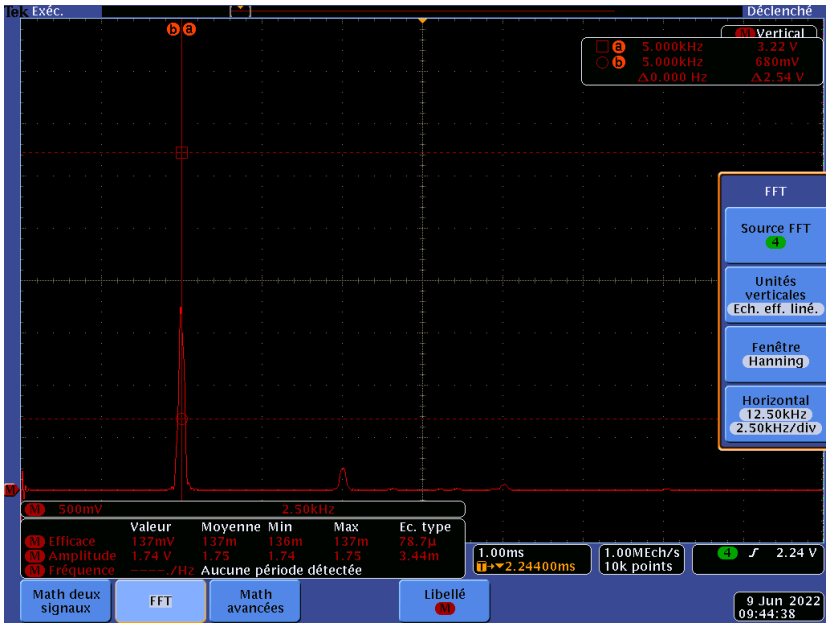


Figure 39 : FFT du sinus différentiel à 5kHz

Figure 40 : FFT du sinus différentiel à 2kHz

On observe bien la fondamentale dans les deux cas et des pics plus faibles pour les harmoniques. Le signal à 5kHz à sa deuxième harmonique un peu plus forte que j'explique par le pic du filtre passe bas, une solution est de remplacer le condensateur de 1uF par 3.3uF pour diminuer la fréquence de coupure, ou d'améliorer la qualité de la PWM, je ferai donc ces tests.

Pour conclure, le circuit fonctionne correctement, il semble qu'avec un peu plus de temps je vais réussir à obtenir des résultats vraiment satisfaisants. Cependant il me reste quand même à implémenter l'asservissement et le tester mais aussi à définir la tension minimum d'alimentation en prenant en compte les pires cas en fréquence et charge afin d'assuré que le circuit fonctionne pour tous les cas définis dans le cahier des charges. Je vais aussi pouvoir faire des mesures de consommations et les comparer à la théorie et aux simulations, mais les premières valeurs de courant consommé données par l'alimentation semblent cohérentes voir plus faible que la théorie.

En ce qui concerne la surface utilisée j'ai pu calculer à partir des composants sélectionnés pour le circuit final, une surface d'environ 150mm2 sans routage, et je pense ne pas prendre trop de risques en affirmant qu'avec un bon routage la surface finale sera inférieur à 400mm2 et puisse se rapprocher des 300mm2 au vu de ce que j'ai pu faire sur mon PCB avec beaucoup de composants de tests en plus et des contraintes de taille des pistes et autres, bien supérieur à ce qui se fait industriellement. Si le temps qu'il me reste suffit, je tenterai de réaliser un routage en important les composants sélectionnés pour le futur circuit afin de vérifier cela.

En terme de coût ma BOM sans les composants ajoutés pour le test varie entre 4€ et 5€. Je n'ai pas le recul nécessaire pour analyser ce chiffre mais il ne semble pas mauvais au vu des discussions que j'ai pu avoir. Enfin le rendement théorique semble varier entre 30% et 65% dans les pires et meilleurs cas, ce qui reste bien plus élevé que la fonction actuelle, et les courants consommés lors des tests semblent encourageant pour augmenter ces rendements sur le PCB.

De plus, des améliorations dans le software ou sur des valeurs de condensateurs seront peut-être trouvées pour arriver à la version finale de la carte et conclure sur la viabilité de cette solution pour l'avenir.

CHAPITRE 4. CONCLUSIONS ET PERSPECTIVES

Au final, ce stage est vraiment très enrichissant pour moi. J'ai pu appliquer énormément de compétences développées à l'école, avec une grosse partie sur l'électronique analogique mais aussi une part non négligeable d'asservissement et de programmation, ce qui correspond exactement aux domaines que j'ai étudiés ces dernières années.

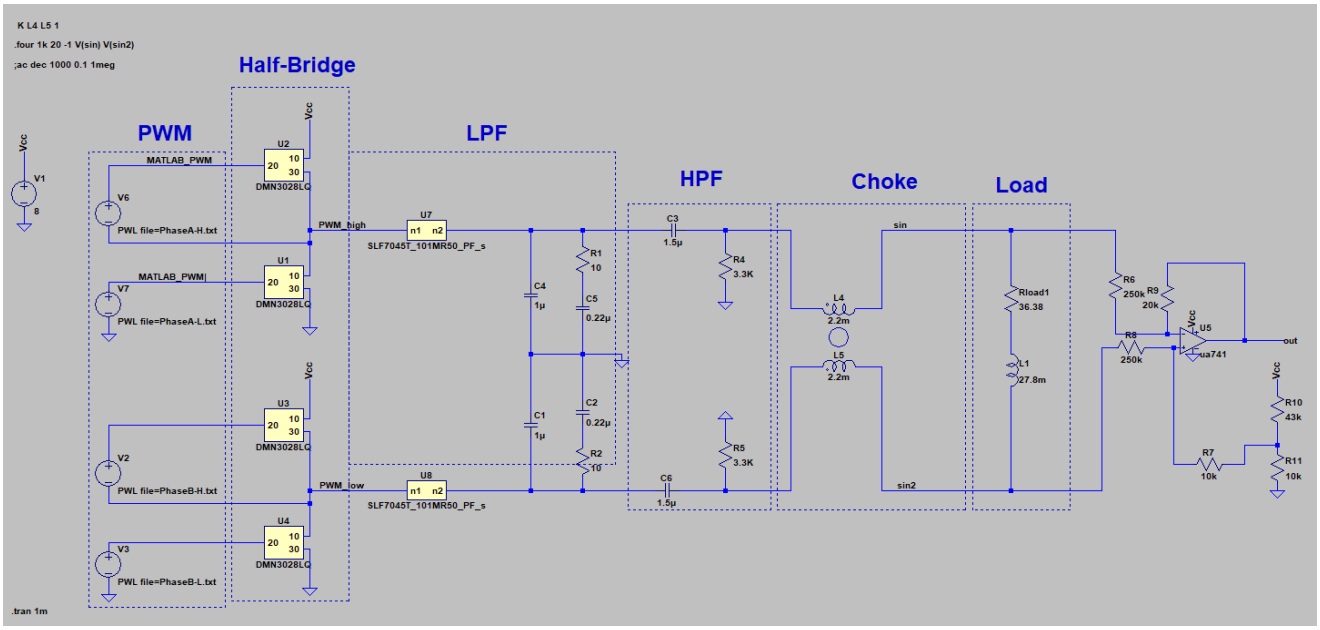
De plus, j'ai eu la chance de travailler sur un sujet très complet et challengeant, qui a donné une dimension très intéressante à ce stage en m'obligeant à développer de nouvelles compétences pour obtenir les résultats souhaités. Le but étant d'évaluer la viabilité ou non d'une solution technologique qui pourrait bien être utilisée dans les produits THALES du futur, si les résultats finaux sont concluants. Le sujet m'ayant intéressé dès le début du stage, j'ai vraiment cherché à obtenir le meilleur résultat possible et faire avancer THALES sur ce point.

Ce stage est une continuité parfaite de mes études à l'INSA de mon point de vue, j'ai retrouvé beaucoup des sujets abordés en cours et des conseils donnés par les professeurs tout au long de la formation. Le fait de voir que l'INSA m'a permis de développer des compétences solides et qui me permettent de ne pas être perdu dans mon stage, m'a apporté de la confiance au fur et à mesure du stage. Pratiquer dans le monde professionnel est aussi quelque chose de très enrichissant et différent des études, c'est un projet de six mois, ce qui est assez rare à l'école, il a fallu gérer mon temps correctement, la notion de responsabilité apparaît, même si elle n'est pas très grande dans mon cas, elle existe. Il faut apporter des résultats aux personnes qui m'ont confié cette tâche et qui espèrent avancer sur ce sujet, alors qu'à l'école on travaille souvent pour apprendre dans un premier temps mais essentiellement pour soi-même.

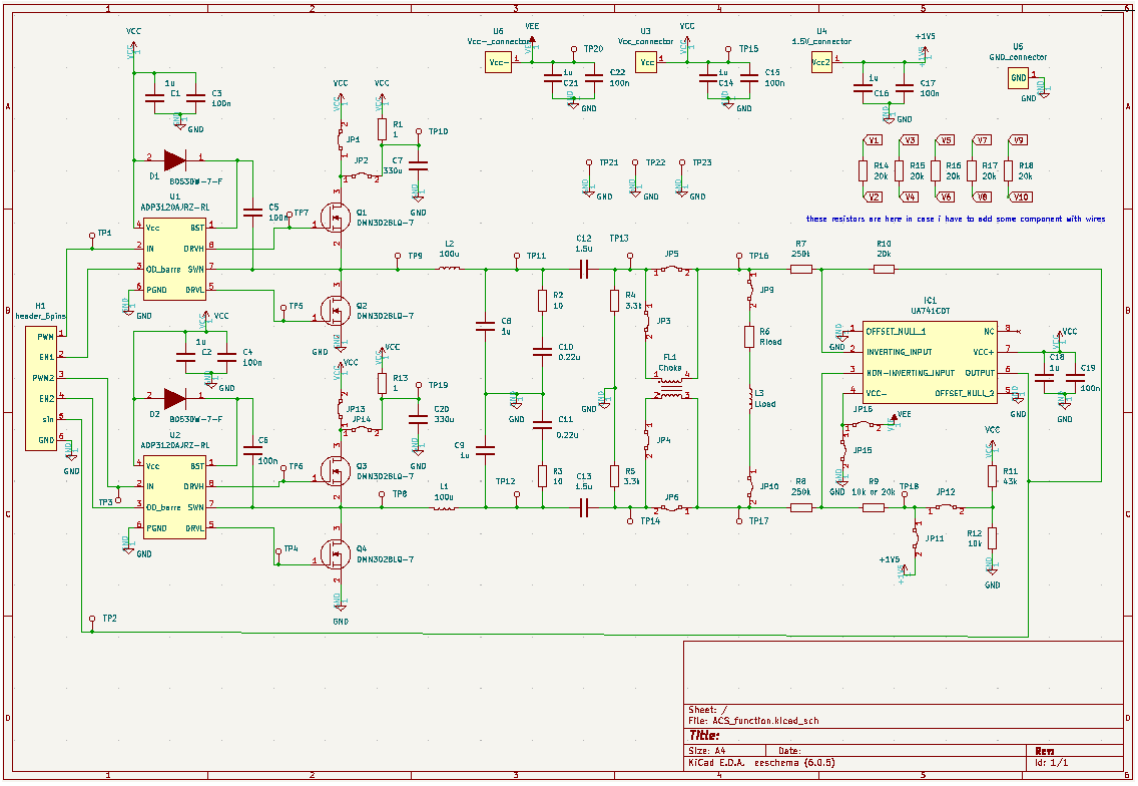
Je suis arrivé à THALES avec l'ambition de découvrir le métier de concepteur en électronique analogique et je ne suis vraiment pas déçu. J'ai pu découvrir non seulement ce que c'est que de travailler dans l'électronique analogique mais aussi tous les autres métiers de l'électronique, le numérique, la carte, le système…etc. Je suis aujourd'hui convaincu de vouloir continuer mon parcours dans l'électronique analogique qui est un domaine qui continu de m'intriguer et me passionner autant dans ma vie professionnelle que personnelle.

ANNEXES

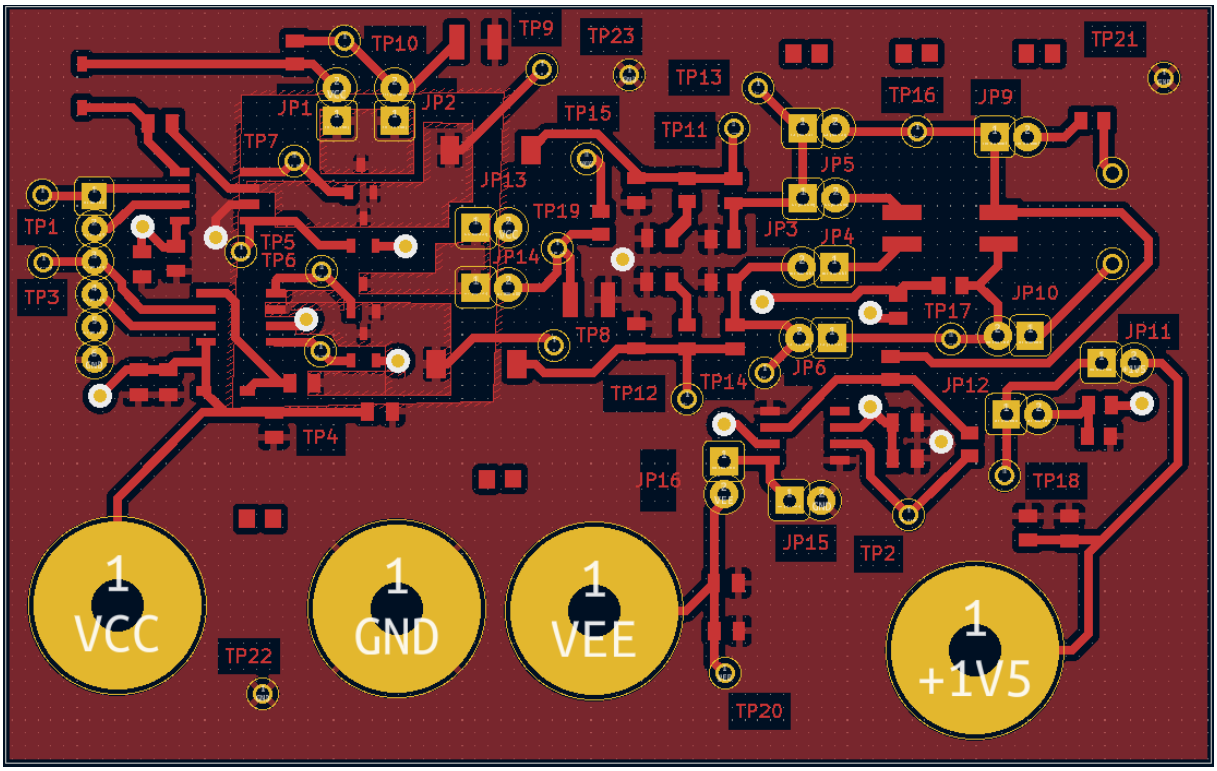
Schématique complète LTspice



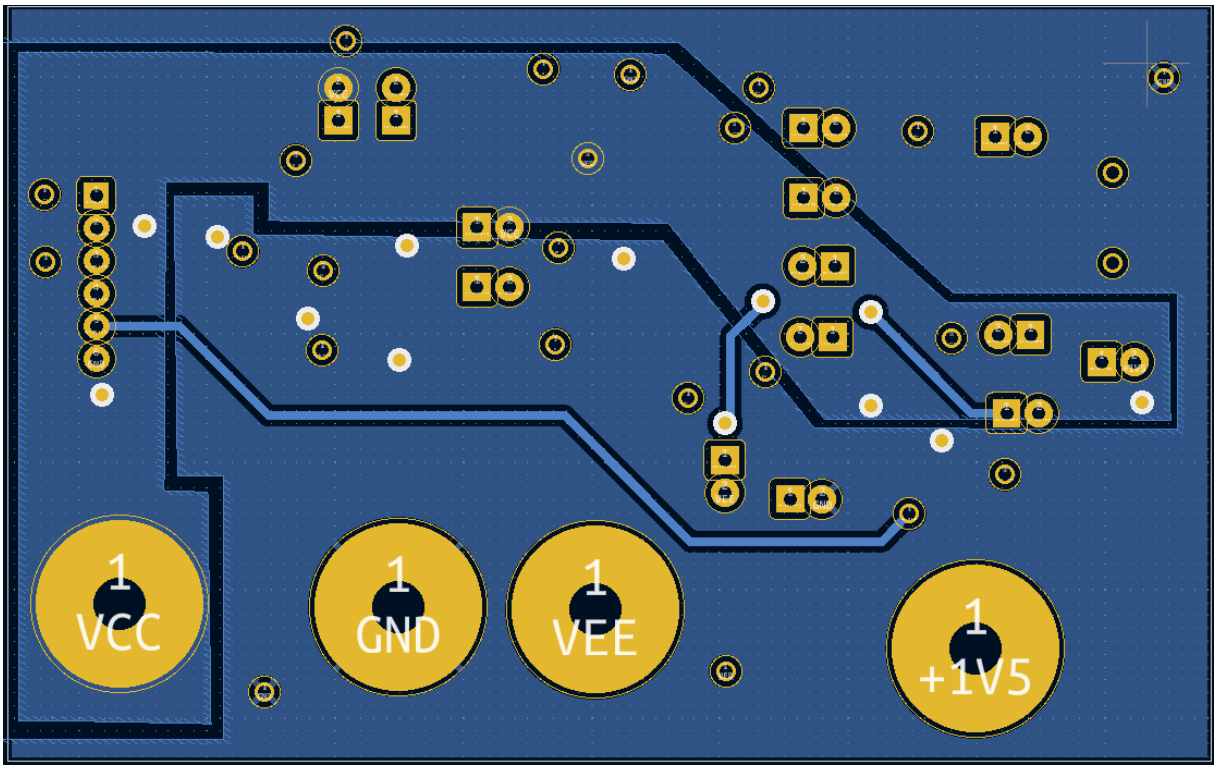
Schématique et routage



Vue du TOP



Vue du BOTTOM



Code main.c

```
/* Includes ------------------------------------------------------------
-------*/
#include "main.h"

/* Private includes ----------------------------------------------------
-------*/
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------
-------*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define ------------------------------------------------------
-------*/
/* USER CODE BEGIN PD */
#define NS       128
#define TIM4CLK  240000000
#define F_SIGNAL 2000
/* USER CODE END PD */

/* Private macro -------------------------------------------------------
-------*/
/* USER CODE BEGIN PM */
uint32_t Wave_LUT[NS] = {
    512, 537, 562, 587, 612, 637, 661, 685, 709, 732, 754, 776, 798,
818, 838,
    857, 875, 893, 909, 925, 939, 952, 965, 976, 986, 995, 1002, 1009,
1014, 1018,
    1021, 1023, 1023, 1022, 1020, 1016, 1012, 1006, 999, 990, 981, 970,
959, 946, 932,
    917, 901, 884, 866, 848, 828, 808, 787, 765, 743, 720, 697, 673,
649, 624,
    600, 575, 549, 524, 499, 474, 448, 423, 399, 374, 350, 326, 303,
280, 258,
    236, 215, 195, 175, 157, 139, 122, 106, 91, 77, 64, 53, 42, 33, 24,
    17, 11, 7, 3, 1, 1, 1, 2, 5, 9, 14, 21, 28, 37, 47,
    58, 71, 84, 98, 114, 130, 148, 166, 185, 205, 225, 247, 269, 291,
314,
    338, 362, 386, 411, 436, 461, 486, 511
};
//deuxième table de valeurs correspondant à un sinus déphasé de 180°
par rapport au premier
uint32_t Wave_LUT2[NS] = {
    499, 474, 448, 423, 399, 374, 350, 326, 303, 280, 258,
    236, 215, 195, 175, 157, 139, 122, 106, 91, 77, 64, 53, 42, 33, 24,
    17, 11, 7, 3, 1, 1, 1, 2, 5, 9, 14, 21, 28, 37, 47,
    58, 71, 84, 98, 114, 130, 148, 166, 185, 205, 225, 247, 269, 291,
314,
    338, 362, 386, 411, 436, 461, 486, 511, 512, 537, 562, 587, 612,
637, 661, 685, 709, 732, 754, 776, 798, 818, 838,
```

```
    857, 875, 893, 909, 925, 939, 952, 965, 976, 986, 995, 1002, 1009,
1014, 1018,
    1021, 1023, 1023, 1022, 1020, 1016, 1012, 1006, 999, 990, 981, 970,
959, 946, 932,
    917, 901, 884, 866, 848, 828, 808, 787, 765, 743, 720, 697, 673,
649, 624,
    600, 575, 549, 524
};

//Wave_LUT=Wave_LUT1*0.8;

//Wave_LUT2=Wave_LUT21*0.8;

/* USER CODE END PM */

/* Private variables ---------------------------------------------------
-------*/

HRTIM_HandleTypeDef hhrtim;

TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim4;
DMA_HandleTypeDef hdma_tim4_ch1;
DMA_HandleTypeDef hdma_tim4_ch2;

/* USER CODE BEGIN PV */
uint32_t DestAddress = (uint32_t) &(TIM2->CCR1); //registre à modifer
pour influer sur le rapport cyclique
uint32_t DestAddress2 = (uint32_t) &(TIM2->CCR2);
uint32_t TIM4_Ticks = TIM4CLK / (NS * F_SIGNAL); // calcul de la
frequence du trigger
/* USER CODE END PV */

/* Private function prototypes -----------------------------------------
-------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_TIM2_Init(void);
static void MX_TIM4_Init(void);
static void MX_HRTIM_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------
-------*/
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
```

```
  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------
-------*/

  /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_TIM2_Init();
  MX_TIM4_Init();
  MX_HRTIM_Init();

  //implémentation de la modification de l'indice de modulation
  for (int i=0; i<128;i++)
  {
  Wave_LUT[i]=Wave_LUT[i]*0.9;

  Wave_LUT2[i]=Wave_LUT2[i]*0.9;
  }

  /* USER CODE BEGIN 2 */
  HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
  HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2);

  HAL_TIM_OC_Start(&htim4, TIM_CHANNEL_1);
  HAL_DMA_Start_IT(&hdma_tim4_ch1, (uint32_t)Wave_LUT, DestAddress,
NS);
  __HAL_TIM_ENABLE_DMA(&htim4, TIM_DMA_CC1);

  HAL_TIM_OC_Start(&htim4, TIM_CHANNEL_2);
  HAL_DMA_Start_IT(&hdma_tim4_ch2, (uint32_t)Wave_LUT2, DestAddress2,
NS);
  __HAL_TIM_ENABLE_DMA(&htim4, TIM_DMA_CC2);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
```

```
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
  RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};

  /** Supply configuration update enable
  */
  HAL_PWREx_ConfigSupply(PWR_LDO_SUPPLY);
  /** Configure the main internal regulator output voltage
  */
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE0);

  while(!__HAL_PWR_GET_FLAG(PWR_FLAG_VOSRDY)) {}
  /** Initializes the RCC Oscillators according to the specified
parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_DIV1;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 60;
  RCC_OscInitStruct.PLL.PLLP = 2;
  RCC_OscInitStruct.PLL.PLLQ = 2;
  RCC_OscInitStruct.PLL.PLLR = 2;
  RCC_OscInitStruct.PLL.PLLRGE = RCC_PLL1VCIRANGE_3;
  RCC_OscInitStruct.PLL.PLLVCOSEL = RCC_PLL1VCOWIDE;
  RCC_OscInitStruct.PLL.PLLFRACN = 0;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2

|RCC_CLOCKTYPE_D3PCLK1|RCC_CLOCKTYPE_D1PCLK1;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.SYSCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB3CLKDivider = RCC_APB3_DIV2;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_APB1_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_APB2_DIV2;
  RCC_ClkInitStruct.APB4CLKDivider = RCC_APB4_DIV2;
```

```
  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_4) !=
HAL_OK)
  {
    Error_Handler();
  }
  PeriphClkInitStruct.PeriphClockSelection = RCC_PERIPHCLK_HRTIM1;
  PeriphClkInitStruct.Hrtim1ClockSelection = RCC_HRTIM1CLK_TIMCLK;
  if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
}
static void MX_TIM2_Init(void)
{

  /* USER CODE BEGIN TIM2_Init 0 */

  /* USER CODE END TIM2_Init 0 */

  TIM_ClockConfigTypeDef sClockSourceConfig = {0};
  TIM_MasterConfigTypeDef sMasterConfig = {0};
  TIM_OC_InitTypeDef sConfigOC = {0};

  /* USER CODE BEGIN TIM2_Init 1 */

  /* USER CODE END TIM2_Init 1 */
  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 0;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim2.Init.Period = 1199;
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
  if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
  if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_PWM_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) !=
HAL_OK)
  {
    Error_Handler();
  }
  sConfigOC.OCMode = TIM_OCMODE_PWM1;
  sConfigOC.Pulse = 0;
  sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
  sConfigOC.OCFastMode = TIM_OCFAST_ENABLE;
  if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_1) !=
HAL_OK)
```

```
  {
    Error_Handler();
  }
  sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
  if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_2) !=
HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN TIM2_Init 2 */

  /* USER CODE END TIM2_Init 2 */
  HAL_TIM_MspPostInit(&htim2);

}

/**
  * @brief TIM4 Initialization Function
  * @param None
  * @retval None
  */
static void MX_TIM4_Init(void)
{

  /* USER CODE BEGIN TIM4_Init 0 */

  /* USER CODE END TIM4_Init 0 */

  TIM_ClockConfigTypeDef sClockSourceConfig = {0};
  TIM_MasterConfigTypeDef sMasterConfig = {0};
  TIM_OC_InitTypeDef sConfigOC = {0};

  /* USER CODE BEGIN TIM4_Init 1 */

  /* USER CODE END TIM4_Init 1 */
  htim4.Instance = TIM4;
  htim4.Init.Prescaler = 0;
  htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
  //ligne permettant d'assurer la freq du sin
  htim4.Init.Period = TIM4_Ticks-1;
  htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim4.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_ENABLE;
  if (HAL_TIM_Base_Init(&htim4) != HAL_OK)
  {
    Error_Handler();
  }
  sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
  if (HAL_TIM_ConfigClockSource(&htim4, &sClockSourceConfig) != HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_OC_Init(&htim4) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
```

```
  if (HAL_TIMEx_MasterConfigSynchronization(&htim4, &sMasterConfig) !=
HAL_OK)
  {
    Error_Handler();
  }
  sConfigOC.OCMode = TIM_OCMODE_TIMING;
  sConfigOC.Pulse = 0;
  sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
  sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
  if (HAL_TIM_OC_ConfigChannel(&htim4, &sConfigOC, TIM_CHANNEL_1) !=
HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_OC_ConfigChannel(&htim4, &sConfigOC, TIM_CHANNEL_2) !=
HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN TIM4_Init 2 */

  /* USER CODE END TIM4_Init 2 */

}

/**
  * Enable DMA controller clock
  */
static void MX_DMA_Init(void)
{

  /* DMA controller clock enable */
  __HAL_RCC_DMA1_CLK_ENABLE();

  /* DMA interrupt init */
  /* DMA1_Stream0_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(DMA1_Stream0_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(DMA1_Stream0_IRQn);
  /* DMA1_Stream1_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(DMA1_Stream1_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(DMA1_Stream1_IRQn);

}

/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOC_CLK_ENABLE();
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();
```

```
__HAL_RCC_GPIOD_CLK_ENABLE();
__HAL_RCC_GPIOG_CLK_ENABLE();
__HAL_RCC_GPIOE_CLK_ENABLE();

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0|GPIO_PIN_14, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_10, GPIO_PIN_RESET);

/*Configure GPIO pin Output Level */
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1, GPIO_PIN_RESET);

/*Configure GPIO pin : PC13 */
GPIO_InitStruct.Pin = GPIO_PIN_13;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

/*Configure GPIO pins : PC1 PC4 PC5 */
GPIO_InitStruct.Pin = GPIO_PIN_1|GPIO_PIN_4|GPIO_PIN_5;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
GPIO_InitStruct.Alternate = GPIO_AF11_ETH;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

/*Configure GPIO pins : PA1 PA2 PA7 */
GPIO_InitStruct.Pin = GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_7;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
GPIO_InitStruct.Alternate = GPIO_AF11_ETH;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pins : PB0 PB14 */
GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_14;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pin : PB13 */
GPIO_InitStruct.Pin = GPIO_PIN_13;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
GPIO_InitStruct.Alternate = GPIO_AF11_ETH;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pins : PD8 PD9 */
GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
GPIO_InitStruct.Alternate = GPIO_AF7_USART3;
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
```

```
  /*Configure GPIO pin : PD10 */
  GPIO_InitStruct.Pin = GPIO_PIN_10;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

  /*Configure GPIO pin : PG7 */
  GPIO_InitStruct.Pin = GPIO_PIN_7;
  GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOG, &GPIO_InitStruct);

  /*Configure GPIO pins : PA8 PA11 PA12 */
  GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_11|GPIO_PIN_12;
  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  GPIO_InitStruct.Alternate = GPIO_AF10_OTG1_FS;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

  /*Configure GPIO pins : PG11 PG13 */
  GPIO_InitStruct.Pin = GPIO_PIN_11|GPIO_PIN_13;
  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  GPIO_InitStruct.Alternate = GPIO_AF11_ETH;
  HAL_GPIO_Init(GPIOG, &GPIO_InitStruct);

  /*Configure GPIO pin : PE1 */
  GPIO_InitStruct.Pin = GPIO_PIN_1;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return
state */

  /* USER CODE END Error_Handler_Debug */
}

#ifdef  USE_FULL_ASSERT
/**
```

```
  * @brief  Reports the name of the source file and the source line
number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and
line number,
     tex: printf("Wrong parameters value: file %s on line %d\r\n",
file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/************************ (C) COPYRIGHT STMicroelectronics *****END OF
FILE****/
```

Script MATLAB de génération de SPWM

```
%% INPUTS
f=300000;                          % switching Frequency
n=350;                               % total number of cycles
Tr=10e-9;                           % rise time
Tf=10e-9;                           % fall time
timeStep =5e-9;                   % minimum time step

waveType = 1 ;                     % 0 = Constant duty
                                   % 1 = Sine

duty = 0.3 ;                       % for the constant duty waveType
fundamental = 5000 ;               % fundamental frequency for the sine
deadBand = 25e-9 ;                 % deadtime

Vdd = 3;                           % 12V supply
Vcc = 0;                           % reference of the Vdd
modulation = 0.35;                  % modulation index
range = 1-2*deadBand*f ;           % duty range to limit duty due deadtime
phaseA = 180 ;                     % phase difference in degree
phaseB = 210 ;
phaseC = 0 ;
% phase difference in degree


%% Calculations
totalTime=n/fundamental;                              % Total simulation time
time = 0:timeStep:totalTime-timeStep;                 % time array in seconds

trigRef= 0.5*sawtooth(2*pi*f*time,1/2)+0.5;         % Triangle reference
sineRefA = modulation*0.5*range*sin(2*pi*fundamental*time+phaseC*pi/180)+modulation*0.5+deadBand*f;
sineRefB = modulation*0.5*range*sin(2*pi*fundamental*time+phaseA*pi/180)+modulation*0.5+deadBand*f;
sineRefC = modulation*0.5*range*sin(2*pi*fundamental*time+phaseB*pi/180)+modulation*0.5+deadBand*f;

arrayLength = n*f*4/fundamental +2;


%% Initilization

time1  = zeros(1,arrayLength);
value1 = zeros(1,arrayLength);
time2  = zeros(1,arrayLength);
value2 = zeros(1,arrayLength);

Ref = sineRefA;

t=3;                                % time counter

%% PWL Generation

for p=1:3
    for i=2:length(time)
        if ((trigRef(i)-Ref(i)>=0) && (trigRef(i-1)-Ref(i-1)<=0)) % positive slope triangle part
            time1(t)=time(i)+deadBand/2;
            time1(t-1)=time(i)-Tr+deadBand/2;        %high side - rising edge
            value1(t)=Vdd;
```

```
            value1(t-1)=Vcc;
            time2(t-1)=time(i)-deadBand/2;
            time2(t)=time(i)+Tf-deadBand/2;          %low side - falling edge
            value2(t)=Vcc;
            value2(t-1)=Vdd;
            t=t+2;

        end
        if((trigRef(i)-Ref(i)<0) && (trigRef(i-1)-Ref(i-1)>0))  % negative slope triangle part
            time1(t-1)=time(i)-deadBand/2;
            time1(t)=time(i)+Tf-deadBand/2;         %high side - falling edge
            value1(t)=Vcc;
            value1(t-1)=Vdd;
            time2(t)=time(i)+deadBand/2;
            time2(t-1)=time(i)-Tr+deadBand/2;       %low side - rising edge
            value2(t)=Vdd;
            value2(t-1)=Vcc;
            t=t+2;
        end
    end
    t=3;   % reset the time counter

    time1(arrayLength) = totalTime;                  % deal with last point
    value1(arrayLength) = value1(arrayLength-1);

    time2(arrayLength) = totalTime;
    value2(arrayLength) = value2(arrayLength-1);
```

```
    if p == 1
        PhaseA_H  = [time1.' value1.'];
        PhaseA_L  = [time2.' value2.'];

        figure;
        plot(time1,value1);
        hold on
        plot(time,Ref);
        plot(time2,value2);

        Ref = sineRefB;
    end
    if p == 2
        PhaseB_H = [time1.' value1.'];
        PhaseB_L = [time2.' value2.'];


        figure;
     plot(time1,value1);
        hold on
        plot(time,Ref);
        plot(time2,value2);

        Ref = sineRefC;
    end
    if p == 3
```

```
        figure;
        plot(time1,value1);
        hold on
        plot(time,Ref);
        plot(time2,value2);
        PhaseC_H = [time1.' value1.'];
        PhaseC_L = [time2.' value2.'];

        Ref = sineRefA;
    end

end

%% Writing PWL files
writematrix(PhaseA_H,'PhaseA-H.txt');
writematrix(PhaseA_L,'PhaseA-L.txt');
writematrix(PhaseB_H,'PhaseB-H.txt');
writematrix(PhaseB_L,'PhaseB-L.txt');
writematrix(PhaseC_H,'PhaseC-H.txt');
writematrix(PhaseC_L,'PhaseC-L.txt');
```

Document Mathcad sur le rendement

Efficiency

In this document, the efficiency of a class D amplifier will be detailed. Efficiency is an important part of all projects and it is why, it must be uderstood properly to make sure that the theory is coherent and conception good. Moreover, I chose to focus on the main loss causes on this document.

In a first time we will consider the architecture below and then the real architecture of the project.

Class D power calculation

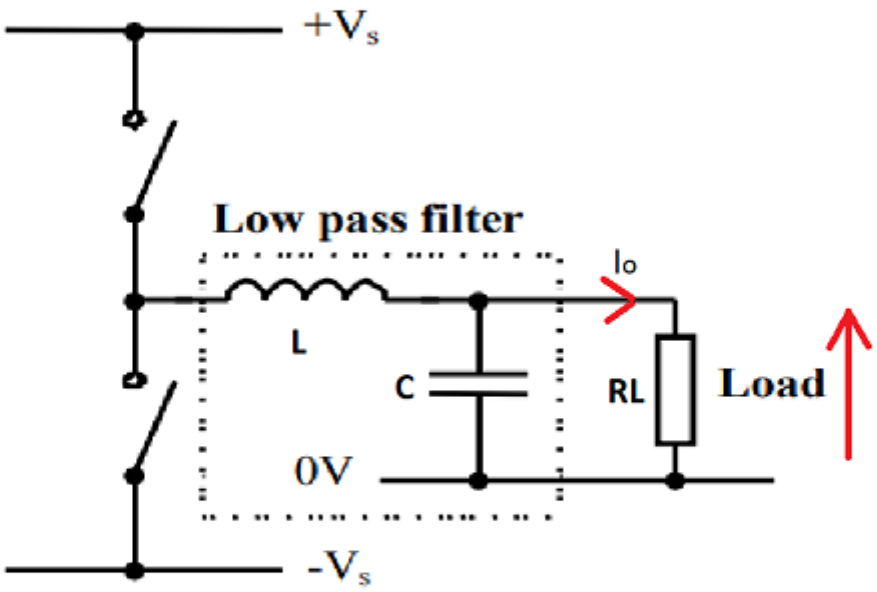


Figure 1: output stage class D amplifier

Let's start with the load power. Io and Vo are considered as the AC current and volatge below:

$$Io = \frac{Vo}{R_L} \quad \text{and} \quad I_{RMS} = \frac{Io}{\sqrt{2}} \quad \text{and} \quad P_L = I_{RMS}^2 \cdot R_L$$

Now, we can calculate the power due to the MOSFET ON resistance and the capacitor's ESR

$$P_{cond} = I_{RMS}^2 \left(R_{DSon} + ESR + R_L\right)$$

Finally, the power losses due to the current ripple must be calculated. We will start by calculating Δi through the inductor as defined below:

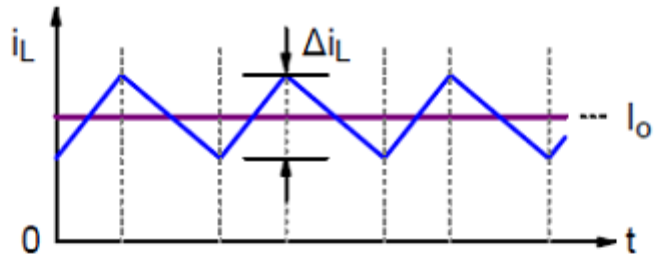


Figure 2: Δi current ripple

Let's use the inductor equation to start:

$$V_L = L \cdot \left(\frac{d}{dt} i_L\right) \quad \Rightarrow \quad \Delta i_L = \frac{V_L}{L} \cdot \Delta t \quad \text{with} \quad V_L = Vin - Vo$$

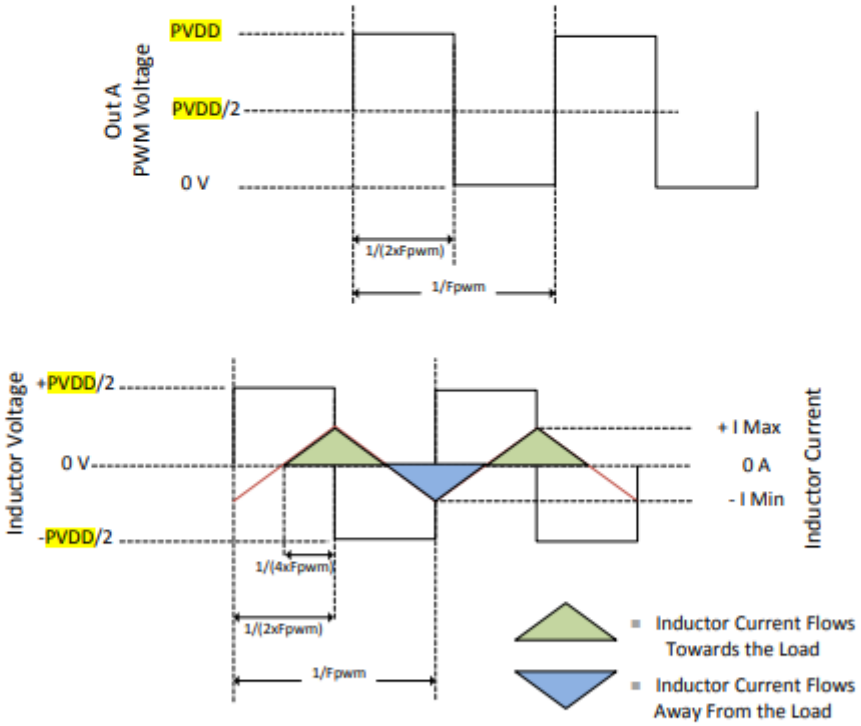


Figure 3: PWM and ripple link representation

$$P_{ripple} = I_{ripple\_RMS}^{2} \cdot \left(R_{DSon} + ESR\right) \qquad I_{ripple\_RMS} = \frac{I_{ripple\_peak}}{\sqrt{3}}$$

$$I_{ripple\_peak} = \frac{PVDD}{2 \cdot L} \cdot \delta t \quad \Rightarrow \quad I_{ripple\_peak} = \frac{PVDD}{2 \cdot L} \cdot \frac{1}{4 \cdot F_{PWM}}$$

$$\Rightarrow \quad I_{ripple\_peak} = \frac{PVDD}{8 \cdot L \cdot f_{PWM}} \quad \Rightarrow \quad I_{ripple\_RMS} = \frac{PVDD}{\sqrt{3}\,8 \cdot L \cdot f_{PWM}}$$

Then we will use these equations to determine the efficiency. The global formula is the next one:

$$\eta := \frac{P_{OUT}}{P_{IN}} \quad \text{or} \quad \eta := \frac{Useful\_power\_output}{Total\_power\_input}$$

We have:

$$P_{OUT} := P_L$$

and

$$P_{IN} := P_{cond} + P_{ripple}$$

$$\eta := \frac{\frac{Vo^2}{2R_L}}{\frac{Vo^2}{2R_L^2} \cdot (R_{DSon} + ESR + R_L) + I_{ripple\_RMS}^2 (R_{DSon} + ESR)}$$

$$\eta := \frac{R_L}{R_{DSon} + ESR + R_L + \frac{\left(2R_L^2 I_{ripple\_RMS}\right)^2}{Vo^2} (R_{DSon} + ESR)}$$

This equation works only for resistive load. If the load is reactive we will have active and reactive power in the process and cos(φ) too.So, the efficiency will change and can easily be found by introducing cos(φ) in Pload and Pcond:

$$P_{cond} = \frac{Vo^2}{2\left(|Z_L|\right)^2} \cdot \left(R_{DSSon} + ESR + |Z_L| \cdot \cos(\varphi)\right) \quad \text{and} \quad P_L = \frac{Vo^2}{2|Z_L|} \cdot \cos(\varphi)$$

Indeed, with load reactive impedance, all load terms must be multiplied by cos(φ). Efficiency is now:

$$\eta := \frac{|Z_L| \cdot \cos(\varphi)}{R_{DSon} + ESR + |Z_L| \cdot \cos(\varphi) + \frac{2\left(|Z_L|\right)^2 I_{ripple\_RMS}^2}{Vo^2} (R_{DSon} + ESR)}$$

It could interesting now, to add the switching losses. The formula of switching losses is the next one:

$$P_{SW} := \frac{1}{2}\cdot PVDD\cdot Io\cdot\left(t_{c.on} + t_{c.off}\right)\cdot F_{SW}$$ For the rest, we will set: $\left(t_{c.on} + t_{c.off}\right)\cdot F_{SW} = T$

$$P_{IN} := P_{cond} + P_{ripple} + P_{SW}$$

$$\Rightarrow\ P_{IN} := \frac{Vo^2}{2R_L}\cdot\left(R_{DSSon} + ESR + R_L\right) + I_{ripple\_RMS}^{\ 2}\cdot\left(R_{DSon} + ESR\right) + \frac{1}{2}PVDD\cdot Io\cdot T$$

and so:

$$\eta := \frac{\left|Z_L\right|\cdot\cos(\varphi)}{R_{DSon} + ESR + \left|Z_L\right|\cdot\cos(\varphi) + \frac{2\left(\left|Z_L\right|\right)^2 I_{ripple\_RMS}^{\ 2}}{Vo^2}\left(R_{DSon} + ESR\right) + \frac{PVDD\cdot Io\cdot\left(\left|Z_L\right|\right)^2}{Vo^2}\cdot T}$$

with $\frac{Io\cdot\left(\left|Z_L\right|\right)^2}{Vo^2} = \frac{\left|Z_L\right|}{Vo}$

$$\eta := \frac{\left|Z_L\right|\cdot\cos(\varphi)}{R_{DSon} + ESR + \left|Z_L\right|\cdot\cos(\varphi) + \frac{2\left(\left|Z_L\right|\right)^2 I_{ripple\_RMS}^{\ 2}}{Vo^2}\left(R_{DSon} + ESR\right) + \frac{PVDD\cdot\left|Z_L\right|}{Vo}\cdot T}$$

To be sure that the switching losses are negligible we will plot the efficiency and compare with and without. Below are the parameters I chose by looking at some datasheets and documents:

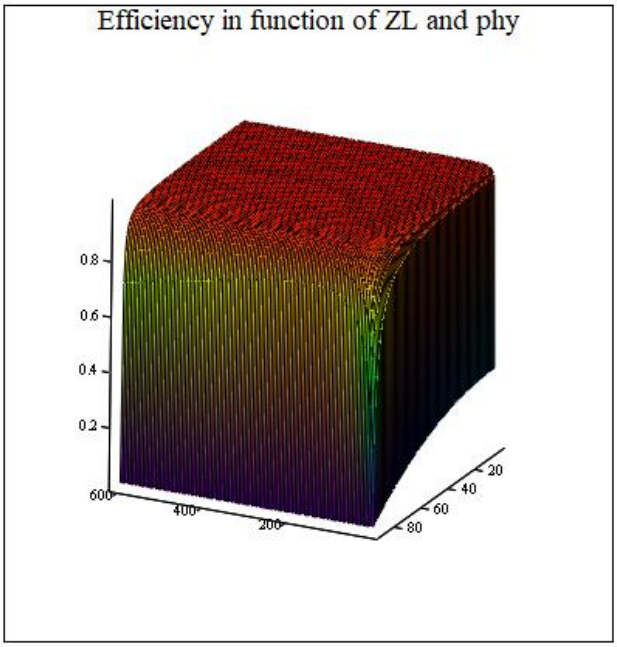
$PVDD := 8V$ $R_{DSon} := 0.03\Omega$ $ESR := 0.25\Omega$ $L := 100\cdot10^{-6}H$ $f_{SW} := 380\cdot10^{3}Hz$

$t_{c.on} := 2.2\cdot10^{-9}s$ $t_{c.off} := 9.7\cdot10^{-9}s$ $Vo := 7V$

$$I_{ripple\_RMS} = \frac{PVDD}{\sqrt{3}4\cdot L\cdot f_{SW}}$$ $$T := \left(t_{c.off} + t_{c.on}\right)\cdot f_{SW}$$

Efficiency without switching losses:

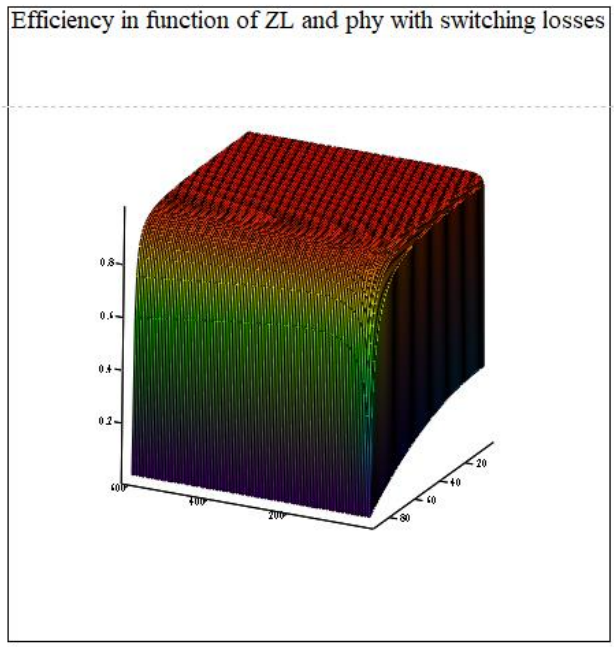
$$\eta\left(Z_L, \varphi\right) := \frac{\left|Z_L\right|\cdot\cos\left(\varphi\cdot\frac{\pi}{180}\right)}{R_{DSon} + ESR + \left|Z_L\right|\cdot\cos\left(\varphi\cdot\frac{\pi}{180}\right) + \frac{2\left(\left|Z_L\right|\right)^2 I_{ripple\_RMS}^{\ 2}}{Vo^2}\left(R_{DSon} + ESR\right)}$$



η

Efficiency with switching losses:

$$\eta_{sw}(Z_L,\varphi) := \frac{|Z_L|\cdot\cos\left(\varphi\frac{\pi}{180}\right)}{R_{DSon}+ESR+|Z_L|\cdot\cos\left(\varphi\frac{\pi}{180}\right)+\frac{2(|Z_L|)^2 I_{ripple\_RMS}{}^2}{Vo^2}(R_{DSon}+ESR)\ldots+\frac{PVDD\cdot|Z_L|}{Vo}\cdot T}$$



η_{sw}

As we can see the results are very similar, which means that switching losses are negligible. Below are the results at a point near to the maximum efficiency:

$\eta(700\Omega, 78) = 0.964$ $\eta_{sw}(700\Omega, 78) = 0.941$

However, if we increase the switching frequency at 5Hz, which is something existing for other application like DC/DC converter, for example. Advantages of using higher frequencies include reduced component's sizes, faster transient load response time, and lower output ripple. However, higher frequencies have reduced efficiencies, increased power losses, RF noise and EMI. I got the following results:

$$f_{SW} := 5\cdot 10^{6}\,Hz \qquad I_{ripple\_RMS} = \frac{PVDD}{\sqrt{3}\,8\cdot L\cdot f_{SW}} \qquad T := \left(t_{c.off} + t_{c.on}\right)\cdot f_{SW}$$

$$\eta(Z_L,\varphi) := \frac{|Z_L|\cdot\cos\left(\varphi\cdot\frac{\pi}{180}\right)}{R_{DSon} + ESR + |Z_L|\cdot\cos\left(\varphi\cdot\frac{\pi}{180}\right) + \frac{2\cdot\left(|Z_L|\right)^2\cdot I_{ripple\_RMS}^{\,2}}{Vo^2}\left(R_{DSon} + ESR\right)}$$

$$\eta(700\Omega, 78) = 0.998$$

$$\eta_{sw}(Z_L,\varphi) := \frac{|Z_L|\cdot\cos\left(\varphi\frac{\pi}{180}\right)}{R_{DSon} + ESR + |Z_L|\cdot\cos\left(\varphi\frac{\pi}{180}\right) + \frac{2\left(|Z_L|\right)^2 I_{ripple\_RMS}^{\,2}}{Vo^2}\left(R_{DSon} + ESR\right) \ldots + \frac{PVDD\cdot|Z_L|}{Vo}\cdot T}$$

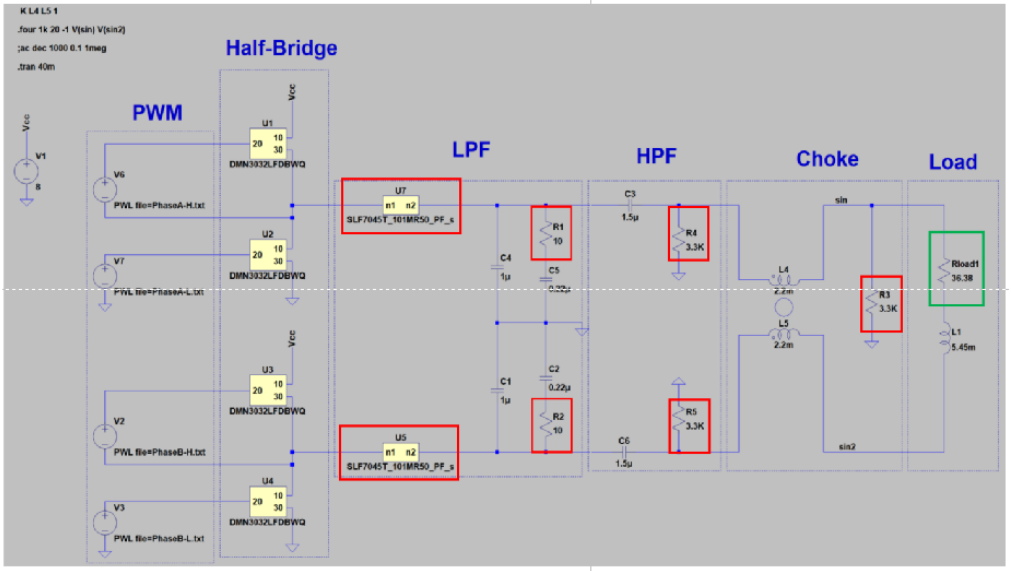
$$\eta_{sw}(700\Omega, 78) = 0.752$$

Here the results are pretty different! The efficiency is 17% lower considering the switching losses.

We can conclude that for our case, with a frequency around the kHz, the switching losses will have almost no influences on the efficiency and so can be neglected.

Real architecture efficiency

This part aims to give a theoretical efficiency of the whole circuit including the other parts (snubber, tracks losses, damping circuits... etc.).



$$\eta := \frac{P_{OUT}}{P_{IN}} \quad \text{or} \quad \eta := \frac{\text{Useful\_power\_output}}{\text{Total\_power\_input}}$$

Pout here is the power in the Load, and it is equal to:

$R_L := 36.38\Omega \qquad I_{RMS} := 40mA \qquad Z_L := 175\Omega \qquad \varphi := 78°$

$$P_L := R_L \cdot I_{RMS}^2 \quad \text{or} \quad P_L = \frac{Vo^2}{|Z_L|} \cdot \cos(\varphi) \quad \text{and PL = Pout}$$

Concerning the 3.3kΩ resistors, all have the same voltage across, 3.5V RMS. So we have:

$$P := \frac{3.5^2 V^2}{3.3 \cdot 10^3 \Omega} \quad \text{and for the 3 resistors:} \quad P_{3.3k} := 3 \cdot P$$

The method for the 10Ω resistor is pretty similar but it include the inductor impedance.

$R_2 := 10\Omega \qquad C_2 := 0.11 \cdot 10^{-6} F \qquad f := 5 \cdot 10^3 Hz \qquad w := 2\pi \cdot f$

$$Z_{eq} := \frac{1 + R_2 \cdot C_2 \cdot w \cdot j}{C_2 \cdot w \cdot j} \qquad Z_{eq.norm} := \frac{\sqrt{1 + (R_2 \cdot C_2 \cdot w)^2}}{C_2 \cdot w}$$

So, we can approximate the current. Indeed, the supply voltage here is 8V, we can assume that it is approximately the maximum amplitude of the sinus and consequently deduced the RMS value:

$$V_{sin.RMS} := \frac{8V}{\sqrt{2}} \quad \text{then} \quad I_{RMS} := \frac{V_{sin.RMS}}{Z_{eq.norm}}$$

And $P := R_2 \cdot I_{RMS}^2$

that's the power disspated by one of the 10Ω snubber resistor

So as there are 2 snubbers, the total power disspated is:

$$P_{snubber} := 2 \cdot P$$

The last rectangle on the left of the figure represent the power disspated by the inductor's ESR. Indeed, the current in the inductor is the strongest of the circuit as the inductors are directly in series with the half bridge amplifier stage.

The first thing to do is to find the current through the inductors. This current is in a huge part due to the capacitors of the LC filters. To be a bit more precise, the current consume by the load is 40mA in this case as we fixed the load values corresponding, but of course, more current will be asked to the power supply to make all the parts of the project work.

We also know that capacitors will pull current to follow the wanted voltage shape. In our case we decided to fixed the sinus frequency to 5kHz, which is the maximum sinus frequency in my specification. In this case, the capacitors linked to the ground in the LC filters, will require more current than if the frequency was smaller. That is logical regarding the capacitor equation we will use now:

$$i := C \cdot \frac{d}{dt} u$$ If the frequency increase, dt will decrease and i increase too.

A.N.: $C := 1 \cdot 10^{-6} F$ $\Delta u := 8V$ as it is the sinus maximum amplitude at this part of the circuit

$$\Delta t := \frac{1}{2 \cdot 5 \cdot 10^{3} Hz}$$

$$I := C \cdot \frac{\Delta u}{\Delta t}$$

We just have to add the other currents calculated earlier to find the current through the inductors

$I_{c4} := I$ $I_{R1} := I_{RMS}$ $I_{R4} := \frac{3.5V}{3.3 \cdot 10^{3} \Omega}$ $I_{R3} := I_{R4}$

$I_{Load} := 40 \cdot 10^{-3} A$

$I_{inductors} := I_{c4} + I_{R1} + I_{R4} + I_{R3} + I_{Load}$

That is the current for the two inductors, so for one: $I_{inductor} := \frac{I_{inductors}}{2}$

In addtition, the ESR of the inductor is: $ESR := 0.25\Omega$

Finally: $P_{inductor} := ESR \cdot I_{inductor}^{2}$

At this point we can add the losses due to the mosfets. Below are the differents types of losses:

$R_{DSon} := 0.04\Omega$ $I_{mosfet} := 0.1A$ $Vo := 7V$

$$P_{cond} = I_{mosfet}^{2} R_{DSon}$$

$L := 100 \cdot 10^{-6} H$ $f_{SW} := 380 \cdot 10^{3} Hz$

$$I_{ripple\_RMS} = \frac{Vo}{\sqrt{3}\,4 \cdot L \cdot f_{SW}}$$

$$P_{ripple} = I_{ripple\_RMS}^{2} \cdot (R_{DSon} + ESR)$$

$t_{c.on} := 2.2 \cdot 10^{-9} s$ $t_{c.off} := 9.7 \cdot 10^{-9} s$ $f_{SW} := 380 \cdot 10^{3} Hz$

$$P_{SW} := \frac{1}{2} \cdot Vo \cdot I_{Load} \cdot (t_{c.on} + t_{c.off}) \cdot f_{SW}$$

$$Pmosfets := 4(P_{SW} + P_{cond}) + 2 \cdot P_{ripple}$$

$$P_{driver} := f_{SW} \cdot 500 \cdot 10^{-12} F \cdot 8^{2} V^{2}$$

At the end we can calculate the efficiency of the global circuit in this case:

$$\eta := \frac{P_L}{Pmosfets + P_{snubber} + P_{inductor} + P_{3.3k} + P_L + P_{driver}} = 0.613$$

All the result above are really similar to the results given by LTspice, except the mosfet losses where there are some mW of difference.

BIBLIOGRAPHIE

Adams, Jun Honda and Jonathan. s.d. «Class D Audio Amplifier Basics.» Application note.

ALter, David M. 2008. «Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Controller.» https://www.ti.com/lit/an/spraa88a/spraa88a.pdf?ts=1655219384393&ref_url=https%253A%252F%252Fwww.qwant.com%252F.

Arnold, Knott. 2007. «Introduction to Class-D Audio Amplifiers.»

Center, eCircuit. s.d. *Push-Pull Output Stage.* http://www.ecircuitcenter.com/Circuits/pushpull/pushpull.htm.

Chirila, Cezar. 2018. *How to Build a Class-D Power Amp.* 29 Aout. https://www.allaboutcircuits.com/projects/how-to-build-a-class-d-power-amplifier/.

D.Grant, S.Burrow and. 2001. «Efficiency of Low Power Audio Amplifiers and Loudspeakers.»

Digi-Key Electronics. 2018. *Simplifier la conception de circuits audio portables basse consommation grâce aux amplificateurs de classe D améliorés.* 31 10. https://www.digikey.fr/fr/articles/simplify-portable-low-power-audio-circuit-design-class-d-amplifiers.

Gaalaas, Eric. 2006. *Class D Audio Amplifiers: What, Why, and How.* Juin. https://www.analog.com/en/analog-dialogue/articles/class-d-audio-amplifiers.html.

Goudswaard, Jeroen. 2019. «Switching ripple reduction in a class-D amplifier using a MOSFET in series with the filter capacitor.» BSc Thesis.

Hubbard, Dylan. 2018. «Reducing noise on the output of a switching regulator.» Design journal.

Keim, Robert. 2016. *Low-Pass Filter a PWM Signal into an Analog Voltage.* 11

Avril. https://www.allaboutcircuits.com/technical-articles/low-pass-filter-a-pwm-signal-into-an-analog-voltage/.

Knol, Eva J. 2018. «Analyzing and reducing switching frequency ripple of class D amplifiers by active filtering.»

Lakkas, George. 2016. «MOSFET power losses and how they affect power-supply efficiency.» Applications journal.

Magdy, Khaled. 2020. *Convert PWM To A DAC – Using PWM To Generate Analog Waveforms.* 1 Juillet. https://deepbluembedded.com/convert-pwm-to-a-dac-using-pwm-to-generate-analog-waveforms/.

McGrady, Chris. s.d. *What is a Snubber and Why do You Want One?* https://www.arrow.com/en/research-and-events/articles/what-is-a-snubber.

Musicarius. s.d. *Les Classes d'Amplificateurs expliquées (A, AB, B, D, H).* http://www.musicarius.com/blog/conseils-techniques/conseils-techniques-ampli-effets/les-classes-damplificateurs-expliquees-a-ab-b-d-h.

1999. «Reducing and Eliminating the class-D Output Filter.» Application report.

STmicroelectronics. s.d. *STM32H743ZxI datasheet.* https://www.st.com/en/microcontrollers-microprocessors/stm32h743zi.html.

Todd, Philip C. 1993. «Snubber Circuits: Theory, Design and Application.»

trevortjes. 2020. *LTSPICE #15: Measuring Total Harmonic Distortion (THD).* 24 Novembre. https://www.youtube.com/watch?v=YPO3DEVzk90.

Wikipedia. s.d. *Push-pull output.* https://en.wikipedia.org/wiki/Push%E2%80%93pull_output.

Yashar, John Satterla and Avi. 2020. «How to Choose a Class-D Audio Amplifier.» White paper.

LISTE DES ILLUSTRATIONS