

Comment protéger les usagers vulnérables (cyclistes, piétons) lors de la fermeture de la lisse de barrières automatisées par l'intégration d'un capteur intelligent ?

Mémoire présenté par
Soufiane El Abdouni

en vue de l'obtention du grade de master en
Ingénieur civil électromécanicien
Option : mécatronique

Promoteurs
Jean Marie Delcambe, Laurent Francis, Yves Therasse

Lecteur
Christophe Craeye

Année académique 2017-2018.

Mémoire réalisé en collaboration avec Automatic Systems



Remerciements

J'adresse ici mes remerciements aux personnes qui m'ont aidé dans la réalisation de ce mémoire.

Je souhaite avant tout remercier mes promoteurs de mémoire : le Pr. Laurent Francis, Yves Therasse et Jean-Marie Delcambe pour leurs aides, leurs disponibilités, leurs encadrements et leurs conseils. Je remercie plus particulièrement Yves et Jean-Marie de m'avoir fourni un bureau au sein de leurs équipes ainsi que pour leurs soutiens logistiques.

Je remercie Thibault Delhayé pour son aide et ses conseils en ce qui concerne l'utilisation du logiciel Comsol.

J'adresse ensuite mes remerciements à Éric Beckers ainsi qu'à Yves Claudic pour l'emprunt de leurs matériels, leurs aides et leurs judicieux conseils en électronique.

Je remercie aussi Damiens Bertrand pour les informations et les conseils qui m'ont été fournis concernant les contraintes auxquels je devais faire attention pour ce type de lisse.

Je désire aussi remercier Bonouh Kada d'avoir mis à ma disposition un espace de travail dans l'atelier de prototypage, de m'avoir fourni les différents outils dont j'ai eu besoin et d'avoir mis à ma disposition son équipe de prototypage.

Je remercie aussi cette équipe de prototypage, qui est composé de Patrick Lefevre, Frederic Musette et de Karolczak Philippe, pour le partage de leurs matériels et pour la mise en place du prototype. Je tenais à remercier plus particulièrement Patrick pour le collage des électrodes sur la lisse, ce fût un travail difficile qu'il a réalisé avec soin et précision.

Je tiens aussi à remercier David, Souley et Nordine pour le prêt de composant électronique.

J'adresse un grand merci pour mon ami Sofyan Mimouni et à son épouse ainsi qu'à ma sœur Chaïmae et à mes promoteurs pour la relecture de mon mémoire.

Je tiens très fortement à remercier mes très chers parents, Mr et Mme El Abdouni, qui m'ont toujours encouragé, soutenu tout au long de mes études et qui m'ont partagé des valeurs telles la persévérance et la patience.

Je tiens aussi à remercier mes petites sœurs, Chaïmae et Yasmina, ainsi que mes petits frères, Jawad et Redouane, pour l'ambiance de travail à la maison et leurs encouragements.

Je remercie toutes les personnes d'Automatic Systems avec qui j'ai pu discuter, échanger durant cette année, ce fût une expérience très enrichissante.

Table des matières

1.	Introduction	1
2.	Cahier des charges	2
3.	État de l'art.....	4
3.1.	Capteurs capacitifs développés précédemment chez A.S.	4
3.2.	Capteurs industriels	5
	Capteur photoélectrique.....	5
	Capteur inductif.....	6
	Capteur ultrasonique.....	7
	Capteur capacitif	7
	Sélection des capteurs industriels	8
3.3.	Capteurs « exotiques »	9
	Système de tapis intelligent	10
	Caméra de détection empêchant les collisions.....	10
	Détection de présence par câble coaxial enterré.....	11
	La ' moustache virtuelle '	12
3.4.	Choix du capteur à étudier	13
	Choix des critères de sélection	13
	Tableau comparatif	14
	Conclusion	15
4.	Simulation Comsol	16
4.1.	Mise en place de la lisse sur Comsol.....	16
4.2.	Représentation de la tête humaine	17
4.3.	Modèles 2D.....	17
	Point de départ	18
	Analyse théorique	18
	Modèle 1 : Système à cinq électrodes	19
	Modèle 2 : Montage AD-HOC des électrodes.....	21
	Modèle 3 : Solution finale.....	23
4.4.	Modèle 3D	27
	Maillage	27
	Lignes de champs électriques	28
	Variation capacitive.....	29
4.5.	Piste d'amélioration	30
5.	Hardware	31
5.1.	Capteur	31
5.2.	Circuit oscillateur	31
5.3.	Circuit d'interface & d'alimentation.....	35
5.4.	FPGA : Deo-Nano.....	36
5.5.	Raspberry Pi 3	37
6.	Software	38
6.1.	Annulation des bruits environnementaux.....	38
6.2.	Filtre software	39

6.3.	Détection.....	42
	Technique de la variance.....	42
	Technique finale de détection	44
6.4.	Conclusion	46
7.	Description des expériences & Résultats.....	47
7.1.	Prototype 0 : Capacité maison	47
7.2.	Prototype 1 : Mini-capalisse.....	47
	Résultats avec l'Arduino	48
	Résultats avec le FPGA + RPI.....	51
7.3.	Prototype 2 : capalisse finale	54
	Signaux de détection : rien VS pastèque.....	55
	Signaux de détection en fonction de la distance	57
	Erreur de détection	60
	Test avec l'épaule	60
	Humidification des électrodes et détection d'une personne	62
	Conclusion	63
8.	Recommandation.....	64
9.	Conclusion	65
10.	Annexes	66
10.1.	Annexe A : Réalisation du modèle Comsol	66
	La lisse.....	66
	L'environnement	67
	Les électrodes et leur fixation	68
	La pastèque	69
	Conseil.....	69
10.2.	Annexe B : Circuits électroniques	69
	Carte d'interface	69
	Carte oscillateurs	70
10.3.	Annexe C : Les codes informatiques	71
	Code python tournant sur le Raspberry-Pi.....	71
	Programmation FPGA : Code SystemVerilog.....	74
11.	Bibliographie	82

1. Introduction

Une grande majorité des barrières sur nos parkings représentent un danger pour les usagers qualifiés de vulnérables. En effet, lorsque ces derniers passent en dessous d'une barrière, celle-ci risque de ne pas les détecter et donc de se refermer violemment sur un usager vulnérable. De tels incidents ont déjà eu lieu et peuvent avoir de graves conséquences. Afin de remédier à ce problème, nous pouvons apercevoir régulièrement, dans nos parkings, la mise en place d'une simple signalétique [1] interdisant à nos piétons de circuler sous la lisse. De plus, un passage est également ajouté sur le côté pour que les usagers puissent circuler en toute sécurité.

Dans le cadre de ce mémoire, nous avons travaillé avec A.S.¹[2] dans le but de trouver et tester une solution, au moindre coût, permettant de détecter sans contact, les personnes passant sous la barrière. Il nous a été proposé lors de cette expérience de continuer sur le modèle du capteur capacitif qui avait été développé par mon prédécesseur. Cependant, nous ne nous sommes pas contentés de simplement reprendre le travail réalisé précédemment, mais nous nous sommes également intéressés aux nouvelles technologies présentes sur le marché et ce de manière à affiner notre recherche. Nous nous sommes posé la question suivante : « *comment protéger les usagers vulnérables (cyclistes, piétons) lors de la fermeture de la lisse de barrières automatisées en intégrant un capteur intelligent ?* »

Nous organiserons ce mémoire de la manière suivante : tout d'abord, nous présenterons une sélection de capteurs tout en analysant les avantages et inconvénients de chacun. Les nouvelles technologies ne faisant qu'évoluer, nous nous sommes également intéressés à de nouveaux projets développés depuis mon prédécesseur et son capteur capacitif. Ensuite, nous analyserons le choix des composants hardware, leurs mises en place et la manière dont nous les avons programmées. Enfin, nous commenterons les résultats obtenus et présenterons quelques recommandations avant de terminer par une brève conclusion.

¹ Automatic Systems

2. Cahier des charges

La mission consiste à mettre en place un capteur capable de détecter les usagers vulnérables sous la lisse. Ce capteur doit être intelligent, car il doit être capable de s'adapter à son environnement et de s'autodiagnostiquer. Pour cela, nous avons réalisé un cahier des charges avec les ingénieurs d'A.S. afin de satisfaire leurs attentes. Voici ci-dessous la dernière version du cahier des charges.

	Cahier des charges d'un détecteur sans contact de personnes sous la lisse	Version 2 02/11/2017
Contexte	Détection sans contact de personnes sous la lisse afin d'éviter des accidents dus à la collision.	
Fonctions	F.1 : Éviter le contact entre la lisse et l'utilisateur vulnérable lors de la fermeture F.2 : Autodétections des défaillances F.3 : Envoie un bit de détection s'il y'a un usager en dessous de la lisse	
Fonctions requises	FR.1.1 : Usager vulnérable : (Personne non détectée via la boucle d'induction) - Adultes – enfants > 90cm - Motocyclistes - Cyclistes FR.2.1 : Check automatiquement son état. FR.2.2 : Envoie un bit de défaillance sur l'output1 (durée minimum du bit 10ms). FR.2.3 : Tension de sortie du bit entre 5-24V FR.3.1 : Envoie un bit de détection sur l'output2 (durée minimum du bit 10ms). FR.3.2 : Tension de sortie du bit entre 5-24V FR.3.3 : Mise de la barrière en mode sécurité.	
Contraintes	C.1 : Système robuste (fonctionnement en conditions extrêmes) C.2 : Utilisation de l'alimentation disponible dans le boîtier C.3 : Système bon marché C.4 : Sûr pour tous les utilisateurs et l'électronique C.5 : Antivol C.6 : Utilisation de la position angulaire C.7 : Corps fantôme (Pastèque) C.8 : Respect des normes EMC	
Contraintes requises	C.R.1.1 : Température de fonctionnement de -25°C à +75°C. C.R.1.2 : Fonctionnement en environnement très humide ou pluvieux. C.R.1.3 : Fonctionnement normal en cas d'orage ou de brouillard. C.R.1.4 : Fonctionnement dans un environnement chargé en éléments perturbateurs. Exemple : un camion ou une voiture qui passe à côté de la barrière ne doit pas être détecté. C.R.2.1 : Alimentation DC disponible dans le boîtier : 24V. C.R.3.1 : Prix acceptable : environ 15% du prix de la barrière qui est d'environ 700€. C.R.4.1 : Pas de contact électrique externe afin d'éviter les chocs électriques. C.R.4.2 : Sécurité électronique (fusible ...). C.R.5.1 : Intégré dans le boîtier ou a une valeur économique très faible. C.R.6.1 : Données allant de -5° à 90° C.R.7.1 : Utilisation d'une pastèque pour la représentation de la tête. C.R.8.1 : Normes EMC : 61000-6-1 ; 61000-6-2 ; 61000-6-3 ; 61000-6-4	

Ce cahier des charges couche sur feuille toutes les propriétés que le capteur devra satisfaire. Afin d'avoir un aperçu de ce qui existe déjà comme capteur de détection dans le domaine de l'industrie et dans le domaine de la recherche, nous allons réaliser un état de l'art dans la section suivante. Une fois cet état de l'art réalisé nous devrons faire un choix de technologie et l'adapter afin qu'il respecte ce cahier des charges.

3. État de l'art

Le domaine de la détection se développe de plus en plus, car de nombreuses compagnies s'y intéressent afin de permettre à leurs produits de s'adapter à l'environnement qui les entourent. Comme on peut le voir depuis quelques années, de nombreux projets doivent être capables de détecter les objets et les humains qui les entourent afin d'effectuer des tâches en toute sécurité. L'un des exemples le plus commun est le secteur de l'automobile [4]. Depuis quelque temps, ce domaine essaye d'améliorer la détection autour des véhicules afin de réduire le risque d'accidents ou de faciliter la vie des conducteurs. Par exemple, l'ouverture des portes sans retirer la clé de sa poche ou encore la détection des piétons qui permet à la voiture de freiner bien plus rapidement que l'homme. Les capteurs de détection continueront fortement à se développer avec l'arrivée de voitures autonomes ou encore des robots en contact direct avec l'environnement humain.

Dans cette section, nous vous présenterons les différentes technologies trouvées durant le processus de recherche. Afin de trouver le maximum de capteurs liés au projet, nous avons décidé de nous concentrer sur les mots clés principaux suivants : le capteur de détection humain, le capteur d'approche et le capteur de proximité. Ces mots clés nous ont permis de trouver de nombreux capteurs ainsi que de nous rediriger vers d'autres recherches fructueuses.

Nous avons partagé "*L'état de l'art*" en 3 parties : la première partie reprend les conclusions du travail effectué par mon prédécesseur ; la seconde partie sera davantage orientée sur les capteurs industriels existants et situés au niveau 9 de l'échelle TRL²; enfin, la troisième partie portera sur les capteurs qui sont « exotiques » et qui se situent souvent en dessous du niveau 5 de l'échelle TRL.

3.1. Capteurs capacitifs développés précédemment chez A.S.

Lors de la première rencontre avec les ingénieurs de A.S., il nous a été présenté le projet sur lequel travailler et où A.S. avait en 2002 initié un projet [3].

En effet, en 2002 A.S. avait mis en place un capteur capacitif utilisant *l'effet sol*. Cette technologie consiste à réaliser un capteur capacitif donc l'électrode supérieure est un fil de cuivre intégré à la lisse. L'électrode inférieure est le sol et le diélectrique est représenté par l'air qui sépare les 2 électrodes. Lorsqu'un humain se trouve sous la lisse, la constante diélectrique change, ce qui fait varier la capacité et donc la fréquence mesurée. La présence d'un microcontrôleur comptant le nombre de tics, permet de détecter les variations de fréquences.

² Technology readiness level : qu'on appelle en français aussi niveau de maturité technologique est une échelle allant de 1 à 9. Lorsqu'on a un capteur avec un niveau 1 cela signifie qu'on est encore au principe théorique et lorsqu'on est au niveau 9 cela signifie qu'on est arrivé à montrer que le capteur est opérationnel et utilisable sur le marché.

Afin que cette technique puisse détecter un usager sous la lisse, il faut générer une courbe de référence qui représente la valeur d'un compteur en fonction du temps. Il faut savoir que la fréquence change tout au long de la fermeture de la barrière, car la distance séparant les 2 électrodes diminue et donc la capacité change. Lorsqu'il y a la présence d'un usager sous la lisse, la courbe mesurée s'éloigne très rapidement de la courbe de référence et dépasse les limites fixées. C'est ce qui déclenche l'arrêt de la lisse. [3]

Malheureusement, cette technique a été testée dans un hangar, dans des conditions météorologiques favorables ne variant pas beaucoup. Il faut savoir que la capacité est sensible à son environnement et que donc la courbe de référence peut ne pas toujours être la même que celle observée lors de l'installation de la barrière sur ce site [1].

Ce travail nous a permis d'avoir une première vision du problème auquel nous devons faire face. Cela nous a donné une idée du travail que nous devons réaliser si nous choisissons de continuer sur cette technologie.

3.2. Capteurs industriels

Après la lecture du rapport technique, nous avons décidé de commencer nos recherches, sur les capteurs de détection sans contact, dans l'industrie. Cela nous a permis de voir s'il existait déjà des solutions pour l'industrie.

Dans cette section, nous présenterons plusieurs capteurs existants actuellement dans nos industries et en vente libre sur le marché des capteurs. Nous allons relever leurs points forts et leurs points faibles ainsi que leur compatibilité à notre cahier des charges.

Il existe cinq technologies de détection de présence qui équipent la majorité de nos industries, ceux-ci sont : l'optique (photoélectrique), l'inductif, l'ultrasonique, le capacitif et le magnétique [5]. Nous allons analyser le fonctionnement des quatre premiers dans les sous-sections qui suivent. Nous avons fait le choix de ne pas présenter le capteur magnétique, en effet, celui-ci ne permet que la détection d'aimants ou d'éléments magnétisés, ce qui sort du cadre de notre travail.

Capteur photoélectrique

Le capteur photoélectrique est l'un des capteurs les plus présents lorsqu'on parle de capteurs de détection sans contact [5]. Il en existe de plusieurs types mais tous fonctionnent de la même manière. Cette technologie est basée sur un principe d'optique, qui permet de détecter l'atténuation ou l'interruption du faisceau lumineux lorsqu'un élément traverse ce faisceau. Cette technologie utilise souvent l'infrarouge comme source de lumière, qui est émise via un émetteur et reçue par un récepteur [6]. Trois configurations sont souvent utilisées en fonction de ce qu'on veut réaliser. Voici un tableau (Figure 1) qui résume les différentes configurations et qui montre les principaux avantages et inconvénients de chacune.

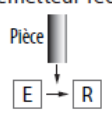
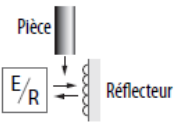
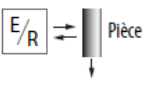
Configuration	Principe	Principaux avantages	Limitations et contraintes
Barrière simple (émetteur-récepteur) 	L'émetteur et le récepteur sont dans deux boîtiers différents montés en vis-à-vis. L'objet passe entre les deux. Il est détecté dès qu'il vient couper le faisceau optique	Portée élevée (plusieurs dizaines de mètres) Détection de pièces très réfléchissantes Bonne répétabilité Utilisation dans des conditions difficiles (pluie, brouillard, fumée, poussières...)	Nécessite de monter deux boîtiers et de les aligner Risque de réflexions parasites Détection délicate dans le cas des objets à faible atténuation (le faisceau ne traverse qu'une seule fois la pièce)
Barrière reflex 	L'émetteur et le récepteur sont dans le même boîtier. La lumière émise est renvoyée par un réflecteur. L'objet est détecté lorsqu'il coupe le faisceau optique	Facilité de montage (émetteur et récepteur dans le même boîtier) Atténuation plus facile à mesurer que dans le cas de la barrière simple (le faisceau traverse deux fois l'objet)	Nécessite d'utiliser un réflecteur Ne convient pas pour la détection d'objets réfléchissants (sauf si on utilise un filtre de polarisation) Utilisation de cellules « spécial verre » préférable dans le cas de produits transparents Portée moins importante qu'en émetteur-récepteur
Détecteur direct (énergétique ou à triangulation) 	L'émetteur et le récepteur sont dans le même boîtier. La lumière émise est réfléchiée par l'objet, qui est ainsi détecté	Montage simple et économique (un seul boîtier) Ne nécessite pas de réflecteur Pour les détecteurs à triangulation : possibilité de s'affranchir de l'arrière et de l'avant-plan, même si ce dernier est plus réfléchissant que l'objet à détecter	La distance de détection dépend de la couleur et du pouvoir réfléchissant de l'objet (sauf pour les détecteurs à triangulation) Portées plus faibles que celles des détecteurs à barrière Détection difficile lorsque l'objet a un état de surface lisse et brillant

Figure 1 : Les différentes configurations du capteur photoélectrique [5]

Cette technologie peut fonctionner dans des conditions difficiles et a une portée élevée. A.S. a déjà expérimenté ce type de capteur, mais celui-ci ne s'avère pas être très efficace, car elle ne permet la détection que sur une faible zone linéaire et est sensible à l'humidité [1]. Cependant des compagnies comme Allen-Bradley appartenant à Rockwell Automation [7] ont développé des capteurs photoélectriques capables de détecter correctement dans un environnement fortement humide, mais le problème de la faible zone de détection persiste. Dans le cadre de notre application, la zone de détection devrait s'étendre un peu plus afin de détecter un usager se trouvant juste avant, juste après ou sous la lisse.

Capteur inductif

Ce type de capteurs est aussi très utilisé pour faire de la détection sans contact [6]. Ils sont beaucoup moins chers que les capteurs photoélectriques et sont très robustes [5]. Ce dispositif est souvent associé à un circuit oscillant qui génère un champ magnétique. Lorsqu'un élément métallique se rapproche de la bobine, celui-ci est parcouru par un courant induit qui va s'opposer au champ magnétique généré par la bobine. Il n'y aura donc qu'une perturbation du champ magnétique et donc une variation inductive. Ce qui signifie qu'une variation de fréquence permettra de détecter la présence d'un corps métallique. Cette technique est utilisée actuellement chez A.S. Une bobine est enterrée sous la route et permet de détecter les véhicules passants sous la lisse, cela permet d'éviter que la lisse ne se referme sur les véhicules [1]. Mais comme nous pouvons le remarquer, cette technique n'est pas utilisable dans le cadre de notre travail étant donné que les objets que nous ciblons ne sont pas métalliques et donc il n'y a pas de courant induit existant. Ce capteur est mis dans ce rapport à titre informatif pour deux raisons, la première étant que ce capteur de proximité sans contact est le plus vendu au

monde selon Gimelec et la seconde raison est qu'A.S. l'utilise [1]. Ce type de capteur ne sera donc pas traité comme une solution viable et il est d'ores et déjà mis de côté.

Capteur ultrasonique

Les ultrasons sont des capteurs très intéressants, car ils fonctionnent sur le principe d'onde sonore (Figure 2). Nous avons comme dans le cas des capteurs photoélectriques, un émetteur et un récepteur. L'émetteur envoie des ondes sonores à haute fréquence. Lorsque celles-ci rencontrent un obstacle, elles reviennent et sont récupérées par le récepteur qui permet de confirmer la présence d'un élément dans la zone [6]. De plus, comme nous connaissons la vitesse d'une onde sonore dans l'air, on peut facilement trouver la distance à laquelle l'obstacle se trouve par rapport au capteur. Ce capteur est très intéressant, car il détecte tous les matériaux mis à part les absorbants phoniques [5]. Il fonctionne dans toutes les conditions normales et a des résultats assez fiables. Il est très utilisé dans le domaine automobile, par exemple pour l'aide au stationnement. Cependant il présente quelques limitations qui sont dues à la pluie ou au courant d'air [8]. Par exemple lorsqu'une forte pluie apparaît, il se peut que des échos parasites apparaissent et fassent croire au dispositif qu'un obstacle est présent [1]. Néanmoins, il est possible de développer un filtre qui permet de réduire fortement cette fausse détection.

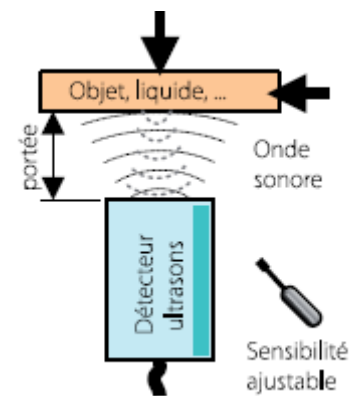


Figure 2 : Capteur ultrason

Capteur capacitif

Ce type de capteur ressemble fortement aux capteurs inductifs présentés précédemment. C'est une capacité composée généralement de deux ou trois électrodes qui sont souvent placées d'une manière coplanaire dont le diélectrique est représenté par l'air. Lorsqu'on alimente le dispositif généralement en courant alternatif, il y a des lignes de champs électriques qui se forment. Lorsqu'un élément, ayant une constante diélectrique fortement différente de celle de l'air, s'approche du dispositif, celui-ci modifie le milieu diélectrique. Cette modification va changer la capacité et le champ électrique de départ. Vu que notre capacité est souvent liée à un circuit oscillant, cela va modifier la fréquence de ce circuit et donc en cas de présence d'un objet, la fréquence détectée sera différente [5].

Nous pouvons voir sur la Figure 3, un modèle montrant les principaux blocs de fonctionnement du capteur capacitif industriel

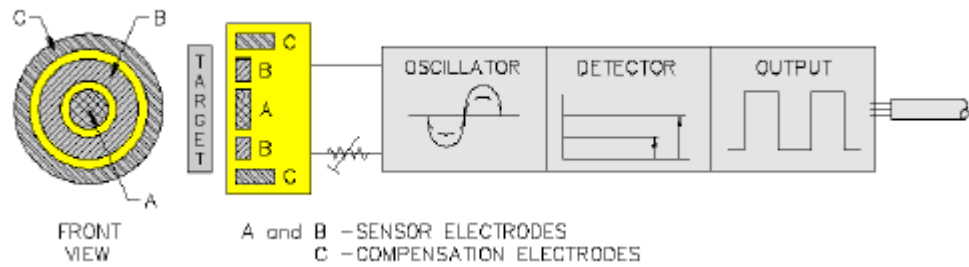


Figure 3 : Blocs principaux d'un capteur capacitif avec électrode de compensation. [5]

Ce type de capteur est très sensible à l'humidité ou aux dépôts surfaciques comme la poussière, la rouille, etc. Afin d'éviter de devoir calibrer le dispositif à maintes reprises, une électrode de compensation est ajoutée comme représenté sur la Figure 3. Cette électrode permet de créer un petit champ qui va fournir des informations uniquement sur l'état de surface. Toutes les électrodes ont le même état de surface en général. Cet état de surface qui modifie la capacité peut alors facilement être négligé ou compensé. Une des techniques utilisées est de faire de la mesure différentielle ; ce qui permet d'annuler cet état de surface et de ne pas en tenir compte. Voici une image sur la Figure 4 qui illustre bien ce champ de compensation.

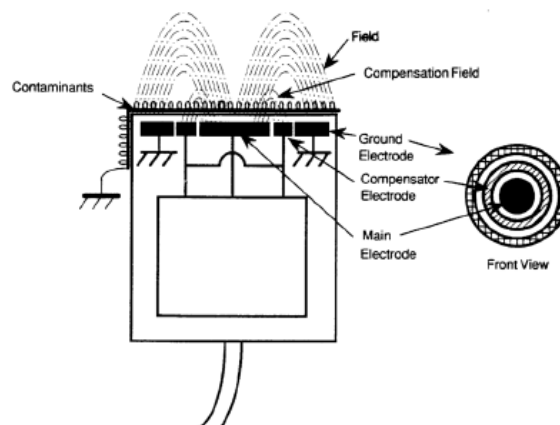


Figure 4 : Champ de compensation [5]

Cette technologie est souvent utilisée dans le cas de prélèvement du niveau d'un liquide ne nécessitant aucun contact direct avec celui-ci. Ces capteurs sont souvent présents dans le secteur pharmaceutique.

Sélection des capteurs industriels

Dans cette section, nous allons appliquer une des méthodologies qu'utilisent les industriels pour choisir leurs capteurs [5] [6]. Cela va nous permettre d'avoir une première idée quant aux capteurs industriels répondant au mieux à notre problématique. Dans la littérature, certains proposent de procéder par élimination dans l'ordre suivant : inductif, capacitif, photoélectrique,

et en dernier l'ultrason [6]. Ainsi on pourra trouver le capteur qui correspond mieux à notre application.

Dans cette section, nous avons décidé d'utiliser une autre méthode plus visuelle qui est présentée dans un article [5]. Ici, on présente un arbre de décision basé sur des critères comme le type de matériaux, la portée de détection et l'environnement dans lequel il va évoluer. En suivant les différentes branches de notre arbre, nous tombons très facilement sur le capteur qui sera le plus adéquat à notre application. [5]

Nous pouvons voir la représentation de cet arbre de décision sur la Figure 5.

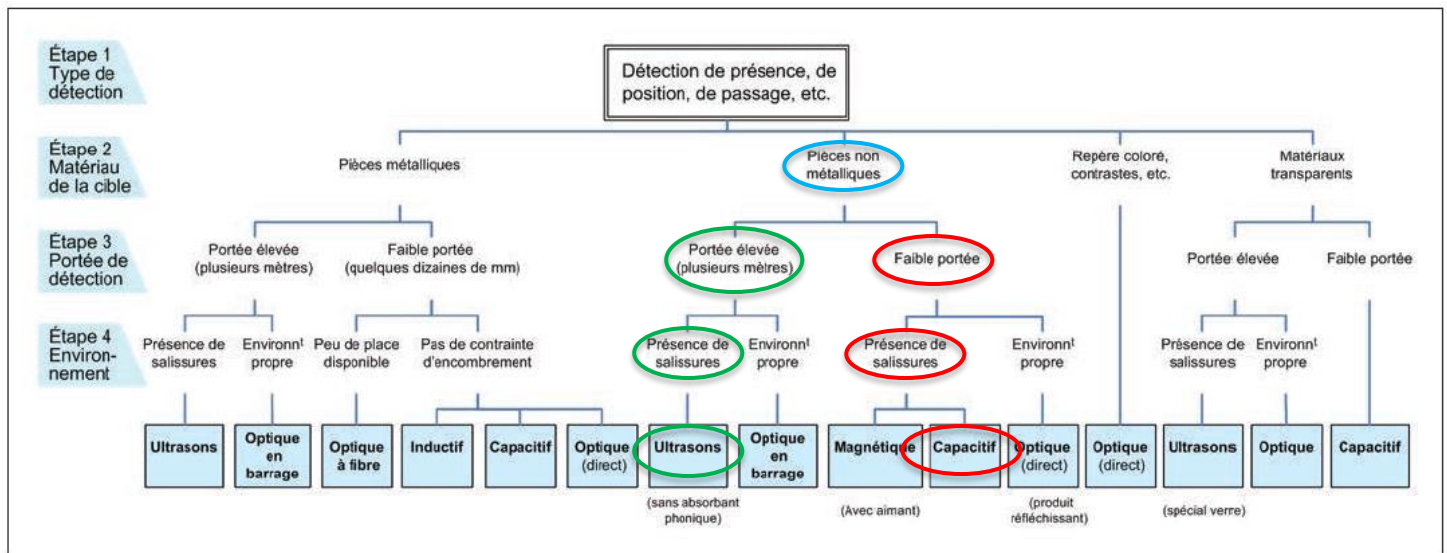


Figure 5 : Arbre de décision [5]

En utilisant cette méthode, on voit que les capteurs capacitifs et les capteurs à ultrasons sont compatibles avec notre problème. Nous avons donc deux solutions qui diffèrent l'une de l'autre par la disposition des capteurs. Par exemple, si les capteurs sont intégrés à la lisse, nous tendrons à ce moment-là vers des capteurs à faible portée (inférieur à 50 cm). En revanche, si les capteurs sont disposés de part et d'autre du passage, nous opterons plutôt pour des capteurs à portée élevée (supérieur à 50 cm).

Cependant, à la fin de ce chapitre, un tableau comparatif comprenant plus de critères reprendra les différents capteurs. Ce tableau final nous permettra de choisir une technologie qui sera développée dans le cadre de ce mémoire.

3.3. Capteurs « exotiques »

Dans cette section, nous allons voir tous les capteurs non standard, en cours de développement ou ayant été développés récemment. Pour réaliser cette partie, nous nous sommes basés sur de nombreux articles scientifiques qui nous ont permis de voir l'intérêt grandissant en

recherche pour les capteurs de détection d'obstacles. Nous avons décidé d'en sélectionner quatre que nous vous présenterons ci-dessous et qui sont, selon nous, les plus intéressantes.

Système de tapis intelligent

Dans cette section, nous découvrons un moyen bien original de faire de la détection humaine. On utilise un tapis intelligent qui est équipé de fibres optiques (Figure 6). Ce tapis que l'on peut voir à l'université de Manchester, permet de détecter les pas des personnes, mais aussi les chutes possibles [9].



Figure 6 : Fibre optique dans le tapis [9]

Ce tapis fonctionne grâce au principe de la déformation de la fibre optique. Lorsqu'une personne est dessus, elle exerce une pression et déforme la fibre optique. Cela perturbe la transmission de la lumière. Comme le tapis est quadrillé par de la fibre optique et que toutes les données sont récoltées et traitées par un ordinateur, la position du pied peut facilement être retrouvée sur le tapis ainsi que la pression exercée par celui-ci [9] comme constaté à la figure 7.

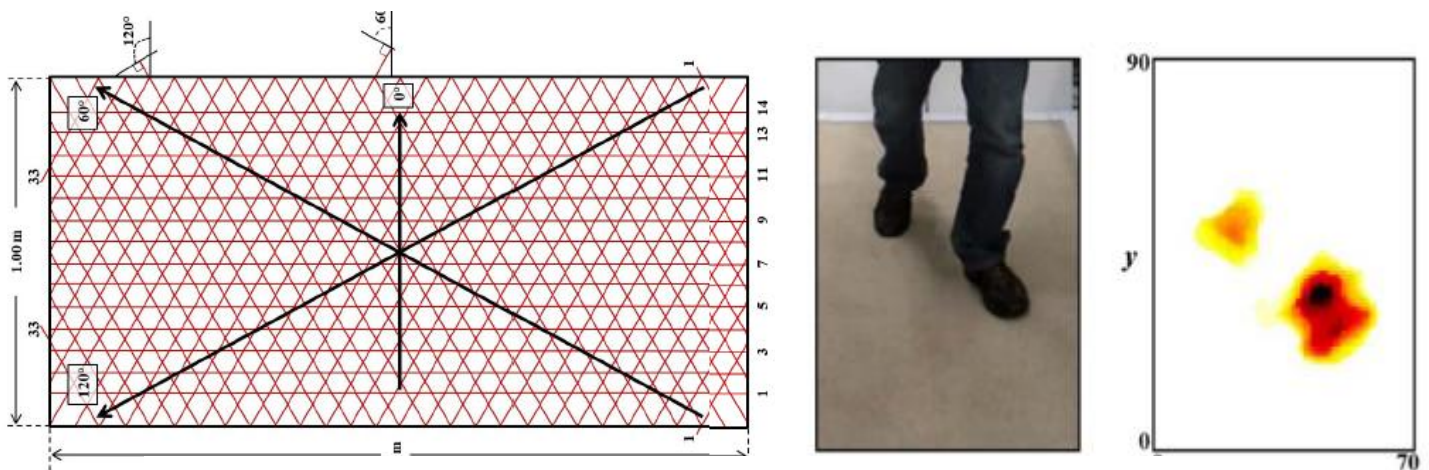


Figure 7 : Quadrillage de la fibre optique et observation de la position et de la pression [9]

Ce tapis a été développé pour des maisons plus intelligentes dans le futur. Avec ce type de tapis, on pourra contrôler les lumières en fonction des pièces occupées, reconnaître la personne présente dans la pièce grâce à ses pas, détecter une intrusion au domicile ou encore contacter les secours si une personne âgée fait un malaise et tombe [9].

Caméra de détection empêchant les collisions

À l'université de Rome "La Sapienza", on s'intéresse de près à la détection grâce aux caméras afin de pouvoir faire coexister les robots et les humains sans incident. Par exemple dans les industries où les bras robotisés sont présents, on observe souvent que ces bras sont soit entourés d'une zone de sécurité, ou sont confinés dans des salles interdites d'accès aux employés. Afin d'avoir un avenir où les robots interagissent avec les humains sans incident, l'université de Rome a travaillé en collaboration avec l'université de Stanford et le laboratoire

d'intelligence artificielle de German Aerospace Center, afin de développer un algorithme basé sur une caméra de type Kinect permettant de suivre les mouvements humains ainsi que ceux du robot [10]. Ils ont réussi à mettre en place un algorithme permettant d'analyser la gestuelle de l'humain et de traduire cette gestuelle pour que le robot effectue la tâche associée à cette gestuelle. Mais cette partie n'est pas en lien avec ce TFE. La partie qui nous intéresse est celle où la caméra détecte que l'humain est dans le champ d'action du robot et qu'elle doit le faire réagir afin d'éviter la collision.

Cette technique est basée sur les images de profondeur fournie par la Kinect et que nous pouvons voir à la figure 8. En processant ces informations, les ingénieurs sont capables de dire où se trouve le corps humain par rapport au robot et éviter la collision en faisant emprunter un nouveau chemin au bras robotisé [10].



Figure 8 : Image de profondeur émise par la caméra [10]

Cette technique est en cours de développement [10] et pourrait peut-être voir le jour dans les années à venir. Mais l'option caméra n'est pas une option négligeable, car comme nous avons pu le lire à plusieurs reprises, les caméras sont de plus en plus utilisées. Par exemple, on utilise des caméras pour détecter les chutes [11] ou encore la présence d'un piéton [12]. De nombreuses techniques sont mises en place pour détecter les personnes par caméra, par exemple l'histogramme de gradient orienté (HOG), ou encore par détection d'émission de chaleur (caméra infrarouge) [12].

Détection de présence par câble coaxial enterré

Cette technique est assez efficace et est gardée comme un trésor de guerre par une poignée d'entreprise spécialisée dans la sécurité d'habitation et de lieux stratégiques [13] [14]. Cette technologie utilise deux câbles coaxiaux parallèles, un émetteur et un récepteur. Ces câbles ont une particularité, ils sont équipés de fentes qui permettent de créer un lobe d'hyperfréquence présent tout le long du câblage

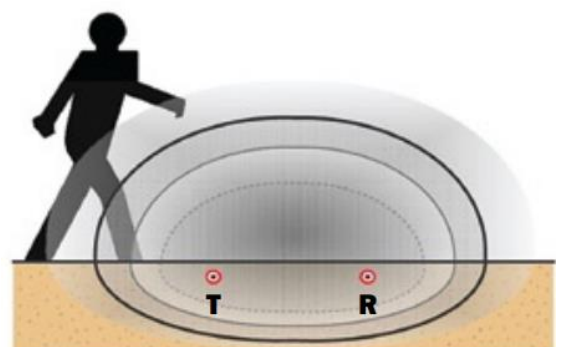


Figure 9 : Lobe d'hyperfréquence [13]

[15] comme nous pouvons le voir à la figure 9. Dès qu'une personne passe à travers ce lobe d'hyperfréquence, cela modifie la fréquence du système qui active une alerte ainsi que la localisation des personnes. Afin de ne pas détecter les animaux sauvages, il est possible de régler la sensibilité de ce capteur en lui fournissant un certain seuil [13].

La ' moustache virtuelle '

Cette technologie "*la moustache virtuelle*" un nom atypique mais pas vide de sens. Il faut savoir que les mammifères utilisent leurs moustaches, qu'on appelle plus scientifiquement des vibrisses [16], pour analyser rapidement leur environnement et détecter les éléments avoisinants leurs têtes. Grâce à cela, ils peuvent éviter des collisions avec des objets qu'ils n'auraient pas vus.

Des ingénieurs de l'université de Stanford en collaboration avec l'Université de Gratz, ont mis en place un capteur capacitif permettant de détecter tout élément se rapprochant du bras robotisé (Robot KUKA). Ces recherches ont le même but que celui vu précédemment avec l'analyse caméra, qui est de permettre une coexistence entre les robots et les humains sans incident [17].

Ce capteur capacitif a quasi le même fonctionnement que celui vu dans la section industrielle. Il y a 3 électrodes sur la partie supérieure, qui permettent la mesure et la compensation.

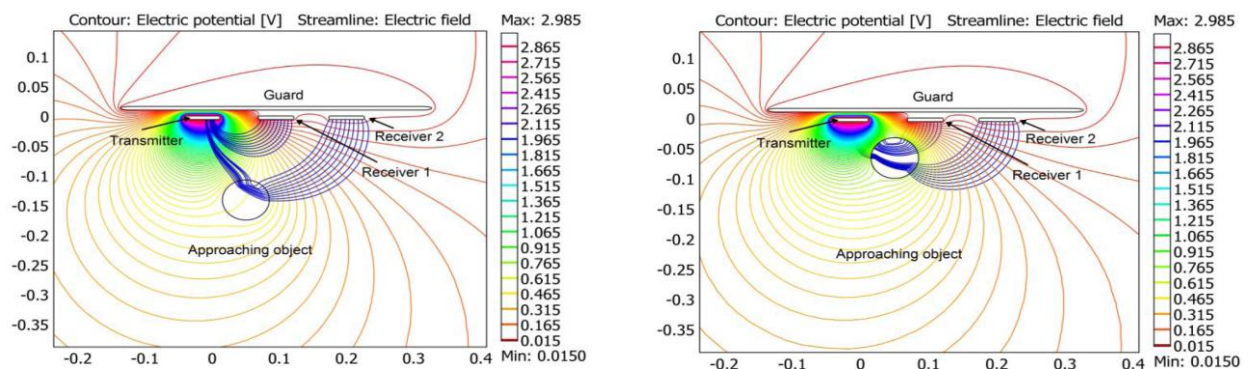


Figure 10 : Champs électriques faiblement et fortement perturbés entre les électrodes [17]

Mais ici on rajoute une autre électrode, qu'on appelle la garde, qui va permettre de donner une direction plus importante au champ électrique comme nous pouvons le voir dans la figure 10.

La technique est établie sur le même principe que le capteur capacitif industriel vu précédemment. Lors de l'approche d'un élément, cela va modifier la constante diélectrique, donc la capacité, ce qui va changer la fréquence du circuit résonnant. Afin de négliger les effets de l'environnement comme l'humidité, la pression ou encore la température, on utilise une électrode de compensation qui va nous permettre in fine d'utiliser une méthode différentielle. La capacité de compensation ne varie pas en cas d'approche d'un élément, car pour influencer cette capacité, il faut être très proche de la surface de l'ordre de quelques centimètres ; ce qui signifie qu'on aura déjà perturbé la capacité de mesure. Cette capacité variera de la même

manière que celle de mesure en cas de changement de température ou du taux d'humidité et donc cela a pour effet d'annuler ces influences, car on compare toujours les capacités dans la méthode différentielle [17].

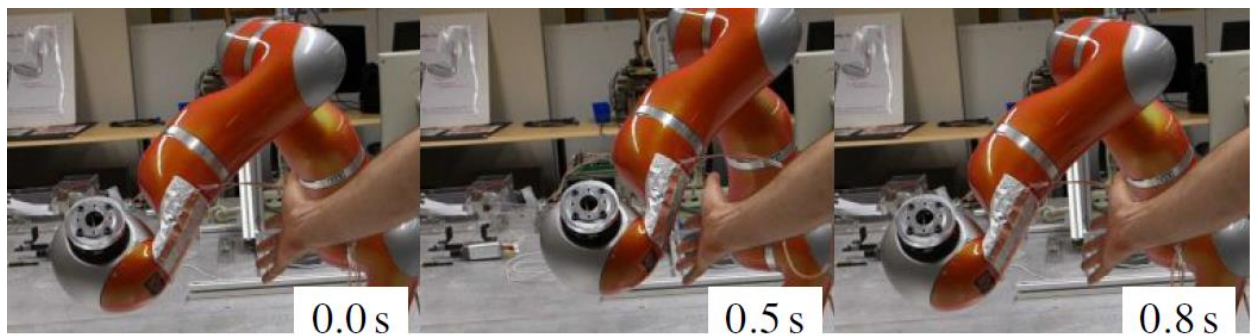


Figure 11 : Le robot évite la collision avec la main (très réactif) [17]

L'expérience présentée sur la figure 11 montre la réactivité de détection de ce genre de capteur qui est quasiment instantané. Dans la vidéo, nous voyons clairement qu'une fois la main présente à moins de 7 cm environ³ le robot se retire immédiatement [17].

3.4. Choix du capteur à étudier

Dans cette section, nous allons évaluer chaque capteur à l'aide d'un tableau de comparaison et choisir celui qui sera le plus adapté à notre projet. À l'issue de cette sélection, nous analyserons plus en profondeur le capteur sélectionné, et nous essayerons de l'adapter au mieux à notre projet.

Choix des critères de sélection

Il existe de nombreux critères sur lesquels nous pouvons analyser nos capteurs, mais nous avons décidé d'en prendre six qui sont objectifs et qui nous permettraient de respecter le cahier des charges que nous avons établi avec A.S.

- Réalisable : En termes de ressource à notre disposition comme le temps, mes compétences ainsi que les compétences des personnes qui m'encadrent.
- Sélectivité : La sensibilité vis-à-vis de l'environnement qui entoure le capteur (pluie, brouillard, humidité, variation de pression, température, poussière ...)
- Réactivité : Vu que nous voulons fermer la barrière rapidement, on a intérêt à ce que le capteur soit très réactif en cas de passage d'utilisateur distrait.
- Entretien : Faut-il déplacer les équipes de A.S. pour des entretiens ? Cela pose problème au niveau des coûts et de l'image qu'on donne de A.S. au client.

³ La distance n'est pas fournie dans l'article mais on peut l'estimer à partir des photos et vidéos,

- Antivol : Est-ce un produit susceptible d'intéresser les voleurs ou d'être dégradé ?
- Prix : un critère qu'il ne faut pas oublier, car le produit doit rester accessible au client. Une solution coûteuse est déjà développée chez A.S. mais les clients ne sont pas intéressés.

Nous allons aussi associer à chaque critère un certain poids pour donner plus d'impact à un critère important. Ce poids sera attribué à l'aide d'étoiles, au plus il y a d'étoiles, au plus il est important. Par exemple, un capteur qui a un poids de 3 étoiles (***) et qui a un « ++ » aura au total 6+ (soit 6+ = 3 fois « ++ »).

Tableau comparatif

Voici le tableau 1 qui compare les différentes solutions possibles en fonction de critères pondérés.

Critères	Réalisable	Sélectivité	Réactivité	Entretien	Antivol	Prix	Total
Poids des critères	***	***	**	*	*	*	
Capt. Capacitif simple	++	-	++	+	++	++	19+ ; 3-
Cap. Photoélectrique	++	--	++	+	+	++	18+ ; 6-
Capt. Ultrason	++	-	+	+	++	++	17+ ; 3-
Camera	-	+	-	+	--	-	4+ ; 10-
Tapis intelligent	--	++	--	0	--	0	6+ ; 12-
Câble coaxial	-	++	++	++	++	0	14+ ; 3-
Moustache virtuelle	++	++	++	+	+	++	24+ ; 0-

Tableau 1 : Tableau comparatif

Comme nous pouvons le remarquer, les capteurs les plus réalisables sont les capteurs industriels, car ils sont déjà produits et vendus sur le marché. Mais il ne faut pas oublier la moustache virtuelle qui elle, est complètement dans mes compétences et dans les compétences des personnes qui m'encadrent. En revanche, le tapis intelligent ainsi que la caméra demanderont beaucoup plus que six mois de travail que je n'ai pas à ma disposition. Il reste le câble coaxial, qui lui est faisable sur ce laps de temps, mais nous devons faire appel à des compétences externes en ce qui concerne cette technologie, ce qui nous rendrait dépendant de l'emploi du temps d'autres personnes externes à l'UCL.

Les capteurs industriels ont été étudiés pour être installés dans des usines. Ils permettent souvent de voir le passage ou la présence d'un produit. Les conditions météorologiques au sein d'une entreprise ne changent pas beaucoup et donc il suffit d'initialiser les valeurs lors de la

dépose du capteur et nous n'aurons plus rien à craindre. Par contre, les capteurs « exotiques » sont un peu plus robustes au changement de condition météorologique.

La plupart des capteurs sont très réactifs, car il faut traiter des données moins complexes comparées aux données reçues par rapport à la caméra ou au tapis.

L'entretien sur des capteurs de détection à distance est rare, car ce sont souvent des éléments hardware qui ne sont pas sollicités mécaniquement, ce qui évite l'usure des composants.

La plupart des capteurs ne susciteront normalement aucun intérêt de la part des voleurs. Mis à part la caméra ou le tapis intelligent qui sont trop visibles et qui pourraient les intriguer.

Le prix reste cependant acceptable pour tous les capteurs. Nul ne va sans le savoir que depuis plusieurs années, le prix des capteurs ne cesse de chuter. Cela permet de développer énormément le monde vers un futur où tout est mesuré et analysé.

Conclusion

Grâce à cet état de l'art, nous avons pu avoir une vue globale de ce qui se faisait en industrie et en recherche. Grâce à nos recherches, nous avons pu évaluer les différents capteurs et choisir celui qui correspond le mieux. Comme nous pouvons le voir sur le tableau de comparaison (Tab. 1) dans la sous-section « tableau comparatif », le capteur à moustache virtuel est celui qui a obtenu le plus de points. Ce sera le capteur que nous développerons dans le cadre de ce mémoire. Dans la section suivante, nous allons faire des simulations afin de pouvoir adapter au mieux ce capteur à moustache virtuel à notre application. Ça nous permettra d'avoir un premier feeling du comportement de ce type de capteur et nous permettra in fine de faire un choix de la variante à développer.

4. Simulation Comsol

Dans ce chapitre, nous allons aborder les différentes simulations réalisées grâce à la version 5.0 du logiciel Comsol. Ce logiciel est un logiciel d'élément finis assez complet, il permet de simuler différents éléments tels que les contraintes mécaniques, la conduction de chaleur, le champ électrique, etc.

Comme nous avons décidé d'implémenter un capteur capacitif coplanaire, les paramètres les plus importants à observer seront plutôt les lignes du champ électrique ainsi que la variation de la capacité. Comsol nous permettra d'avoir une idée assez proche de la réalité en ce qui concerne la disposition des lignes de champs électriques, la variation de ce champ ainsi que la variation de la capacité lors de l'approche d'un corps fantôme. Le fait de pouvoir simuler les différents modèles imaginés, permet d'éviter de faire des prototypes à l'aveugle et non fonctionnel. Grâce au logiciel, nous pouvons faire des prototypes virtuels qui nous donnent une idée de la représentation réelle. Nous pouvons donc après les simulations, sélectionner le meilleur modèle et le prototyper en réalité.

4.1. Mise en place de la lisse sur Comsol

Afin de pouvoir simuler les différents modèles, nous devons élaborer tout d'abord un modèle de base qui représente la structure fixe. Cela signifie que cette structure de base ne changera pas en fonction des modèles et pourra être utilisée à la base de chaque modèle. Dans notre cas, cela revient à dessiner la lisse ovale dont les plans ont été fournis par A.S.

Nous avons l'habitude de voir des lisses rondes sur nos parkings, mais celles-ci ne sont pas assez rapides. Les plus rapides sont les barrières équipées de lisses ovales qui sont généralement installées sur les autoroutes ou sur les parkings qui veulent avoir des barrières ultrarapides. Ce type de barrière peut se fermer en moins d'une seconde ($\pm 0.7s$). [1]

Afin d'atteindre cette performance, les mécaniciens ont dû concevoir la lisse d'une certaine manière. Nous remarquerons que la lisse est creuse et est faite en aluminium, de plus, il y a des indentations qui ressortent à l'intérieur (voir figure 12). Tout ce design permet à la lisse d'être plus rigide, plus aérodynamique, tout en étant la plus légère possible.

Dans le cadre de nos simulations, nous pouvons négliger ces petites tiges, car elles auront peu d'impact sur le champ électrique contrairement au creux et à l'aluminium qui composent la

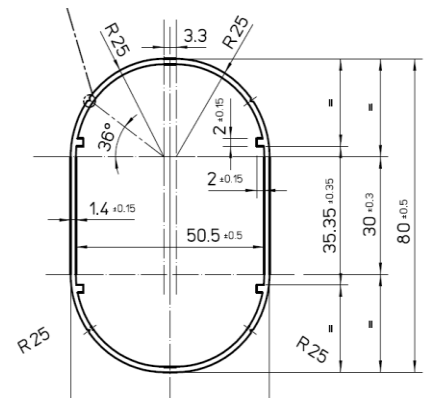


Figure 12 : Plan de coupe de la lisse

lisse. Le modèle 2D simplifié sera composé d'une paroi en aluminium et respectera la forme extérieure ainsi que l'épaisseur d'aluminium. Le modèle présent à la figure 13 sera utilisé à la base de chaque simulation réalisée sur Comsol. Le modèle 3D quant à lui, sera seulement une extrusion de ce modèle 2D.

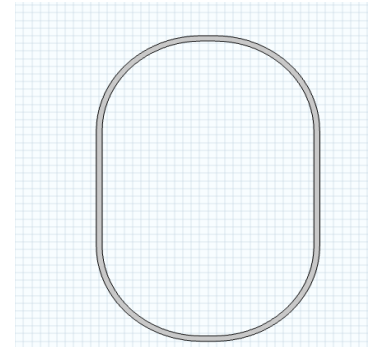


Figure 13 : Modèle Comsol de la lisse

Les modèles que nous allons réaliser dans la suite doivent tenir compte des contraintes de ce type de lisse. Selon les ingénieurs, nous devons faire attention à deux aspects.

- Le poids : le dispositif ne doit pas excéder les 150 g/m, sinon il y'aura une perte de performance lors des mouvements de fermeture et d'ouverture de la lisse.
- Enlèvements de matière : La zone circulaire est une zone sensible, il nous est impossible de retirer de la matière pour y introduire des capteurs ou autres.

Ma solution devra être légère et minimiser les modifications sur lisse. La solution évidente est d'utiliser des fines barrettes d'aluminium qui viendront se fixer sur la surface.

4.2. Représentation de la tête humaine

Afin de représenter la tête lors de nos tests, nous avons décidé de mettre une pastèque comme corps fantôme, c'est ce qui représente le mieux la tête humaine en termes de taille, de portion d'eau (92%) ainsi que de dureté du crâne (c'est ce qui est utilisé pour tester les casques de moto). Pour représenter cette pastèque sur Comsol nous avons décidé d'utiliser un long cylindre plein d'eau dans le modèle 2D et d'une sphère d'eau dans le modèle 3D.

4.3. Modèles 2D

Nous avons commencé par simuler des modèles 2D afin d'avoir une première idée du meilleur modèle. Faire des simulations sur des modèles 2D est beaucoup plus rapide que sur le modèle 3D, cela nous permet d'en tester plusieurs et de choisir le meilleur.

Considérer que notre modèle de lisse peut s'apparenter à un modèle 2D est totalement correct, car notre lisse a une épaisseur faible devant la longueur et donc nous pouvons faire l'hypothèse que la lisse tend vers l'infini et que les effets de bord sont négligeables. Cependant, en ce qui concerne la représentation de la tête, le modèle 2D le considère comme un cylindre et non une sphère. Ce qui nous poussera à faire un modèle 3D plus précis une fois un bon modèle 2D obtenu.

Comme nous voulons respecter les contraintes imposées par le design de ce type de lisse, nous avons décidé d'utiliser des électrodes en aluminium qui seront introduites dans un profil en

polymère ou collées sur la surface grâce à une colle isolante (p. ex. : un tape double-face). Une attention doit être faite afin d'éviter tout contact entre la lisse et les électrodes, car la lisse est mise à la masse pour avoir le rôle de garde. Cela va permettre de concentrer le champ électrique dans une direction précise.

Dans cette section, nous allons aborder dans un premier temps les équations de base et analyser les paramètres importants et par la suite nous allons exposer les trois modèles les plus intéressants, ce qui va nous permettre d'observer le processus de convergence vers la solution actuelle.

Point de départ

Pour réaliser les différents modèles, nous sommes partis sur base du modèle moustache présenté dans la section 3.3. Le dispositif que nous voulons implémenter doit fonctionner de la même manière (figure 14), mais seulement dans notre cas la surface où nos électrodes seront installées est circulaire. La taille des électrodes est aussi beaucoup plus grande que celles présentées dans l'article (environ 15cm vs 2m) et la zone de détection est plus importante.

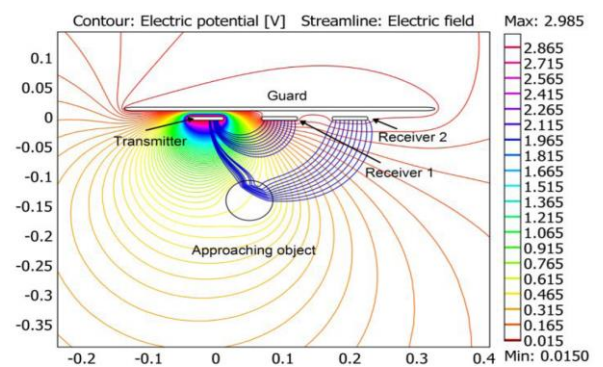


Figure 14 : Modèle moustache virtuel [17]

Analyse théorique

Dans cette section, nous n'allons pas rentrer dans les détails des équations des capacités coplanaires, mais nous allons plutôt voir quels sont les paramètres qui vont fortement influencer notre dispositif.

Pour commencer, prenons une capacité standard comme à la figure 15.

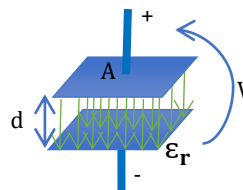


Figure 15 : Capacité standard

Comme nous pouvons le voir sur la figure 15, il y'a plusieurs paramètres qui vont fixer la valeur de notre capacité à commencer par la surface des électrodes (A), la distance séparant nos électrodes (d), la tension (V) et la permittivité diélectrique relative (ϵ_r). La relation qui donne la valeur de la capacité en fonction de ces paramètres est la suivante :

$$C = \frac{\epsilon_0 \epsilon_r A}{d} V^2$$

La technologie qu'on veut développer dans ce mémoire est basée sur la variation de capacité due à un changement du diélectrique. Généralement dans notre cas ce diélectrique sera de l'air. Mais lorsqu'un élément comme une main se trouve dans cette zone inter-électrode alors là, le diélectrique sera composé d'une partie d'air et d'une partie d'eau. Ce qui s'explique par le fait que le corps humain est composé à majorité d'eau et lorsqu'on met une main dans cette zone inter-électrode on y introduit une portion d'eau, ce qui fait changer le milieu diélectrique qui était composé à majorité d'air. Ce qui est représenté à la figure 16.

En quoi est-ce que ce changement de diélectrique change notre valeur capacitive ?

Comme nous avons pu le remarquer dans la formule de la capacité, celle-ci dépend de la permittivité relative et comme nous le savons la permittivité relative de l'air sec est égale à 1 et celle de l'eau vaut 80. Donc lorsqu'on introduit notre main dans la zone inter-électrode on passe d' $\epsilon_r = 1$ à une valeur d' ϵ_r perçus par la capacité se trouvant entre 1 et 80.

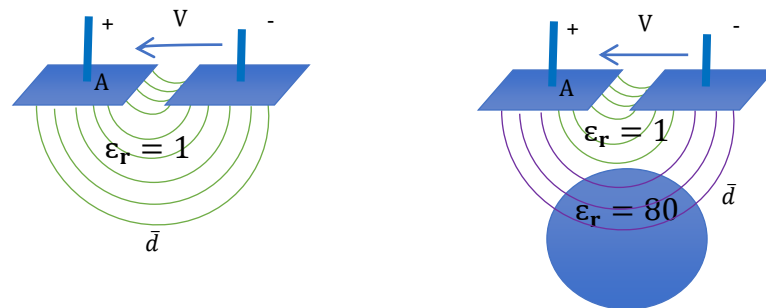


Figure 16 : Variation du diélectrique dans une capacité coplanaire

Ce changement de diélectrique apporte aussi des perturbations sur les lignes de champ électrique c'est ce que nous allons voir dans les simulations qui suivent.

Dans la suite de ce travail, nous évaluerons à plusieurs reprises les différentes capacités du système grâce à la densité d'énergie potentielle électrique du système fourni par Comsol. Etant donné qu'il n'y a aucune autre forme d'énergie mise à part les énergies stockées par nos capacités, nous supposons que la densité d'énergie potentielle électrique totale du système (W_e) correspond à la densité d'énergie emmagasinée par les capacités (W_q). De plus, comme nous travaillons avec des densités d'énergies cela revient à avoir des valeurs de capacité par mètre (C). La relation est donnée par les équations suivantes :

$$W_e = W_q = \frac{1}{2} CV^2$$

Modèle 1 : Système à cinq électrodes

C'est l'un des premiers modèles qui a été simulé, celui-ci est composé de cinq électrodes rectangulaires en aluminium et d'une membrane en élastomère. Dans ce type de design, on voit deux dispositifs fusionnés, un pour la partie de gauche et un pour la partie de droite. Comme nous pouvons le voir à la figure 17, l'électrode '1' est celle qui est alimentée en 3.33V et toutes les autres ainsi que la lisse sont mises à la masse. La capacité créée entre la '1'/'2'

ainsi qu'entre la '1'/'3' sont les capacités mesurant l'environnement. La capacité formée par les électrodes '1'/'4' ainsi que la '1'/'5' sont les capacités de détection. Ensuite, il y a une capacité créée entre la '1' et la lisse qui est une capacité parasite.

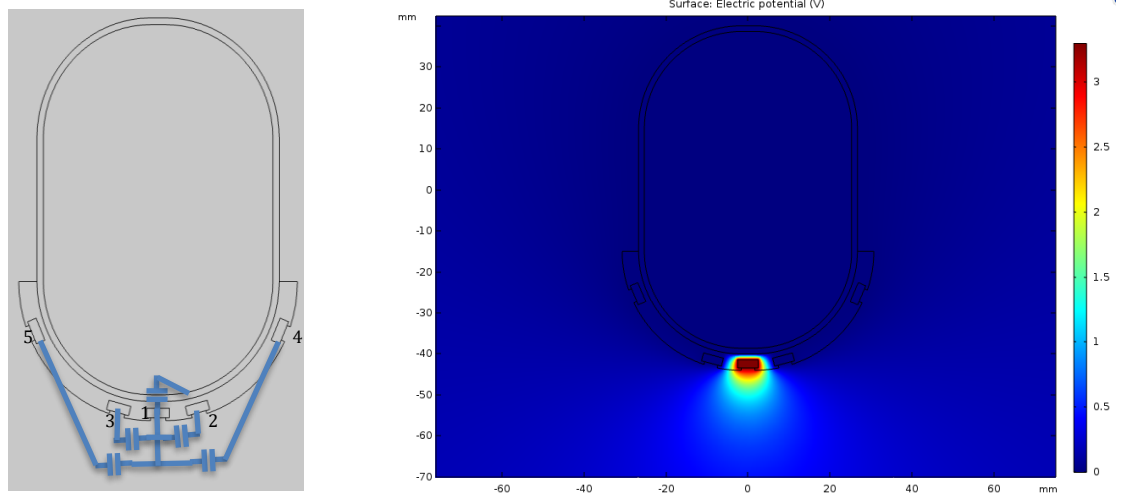


Figure 17 : Disposition et alimentation des électrodes

Une fois le modèle disponible nous pouvons afficher les lignes de champs électriques ainsi que les vecteurs d'intensité du champ électrique normalisé. (voir figure 18)

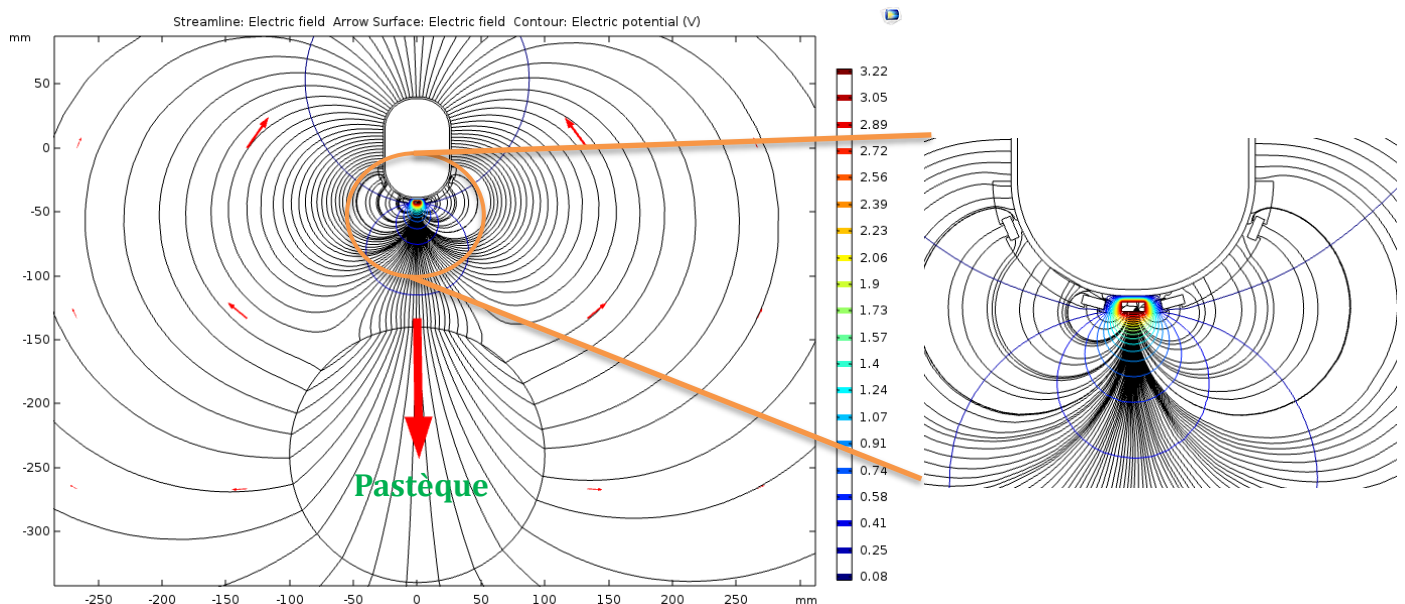


Figure 18 : Lignes de champs électriques

Ce modèle fonctionne à courte distance. Comme nous pouvons le voir sur la figure 18, la pastèque représentant la tête se trouve à 10 cm seulement, mais celle-ci n'influence en rien les lignes de champs qui relient les électrodes 1-4 & 1-5. Vu qu'il n'y a pas de variations des lignes de champs, cela signifie que le milieu diélectrique inter-électrode n'a pas changé ce qui pousse à conclure que la capacité n'a pas changé. De plus notre dispositif est assez lourd, car il contient cinq électrodes et une membrane en polymère. Nous estimons la masse linéique de ce dispositif à 250g/m. Ce qui représente environ 27% de la masse linéique de la lisse.

Même si ce modèle n'a l'air pas viable à cause de son poids et de sa faible distance de détection. Nous avons décidé d'analyser les variations de capacité en fonction du rapprochement de la grosse pastèque. Cela nous donne le tableau 2 :

Distance	Densité d'E. (J/m)	Capa. Tôt. (pF/m)	Var. de capa (pF/m)
Sans obstacle	3540.69 E-11	122.94	Réf
1cm	3584.48 E-11	124.46	1.52
6cm	3541.12 E-11	122.95	0.01
11cm	3540.78 E-11	122.94	0
21cm	3540.71 E-11	122.94	0
31cm	3540.70 E-11	122.94	0

Tableau 2 : Evaluation de la variation capacitive

La colonne distance, correspond à la distance séparant la partie supérieure de la pastèque et le bas de la lisse.

Afin de pouvoir calculer la capacité linéaire totale sur Comsol, nous avons besoin de la densité d'énergie fournie par Comsol (We). Une fois qu'on a calculé la capacité de chaque cas, nous pouvons trouver la variation de la capacité en fonction du cas où il y a aucun obstacle.

Nous pouvons observer que le tableau confirme bien ce que les lignes de champs nous ont montré. La variation de capacité est quasiment inexistante une fois que la tête se trouve à plus de 6cm.

Nous n'irons pas plus loin dans le cadre de ce dispositif, car il viole l'une des contraintes principales imposées par A.S. qui était la masse.

Modèle 2 : Montage AD-HOC des électrodes

En faisant de nombreuses simulations, nous sommes tombés sur un modèle AD HOC (Figure 19). Nous utilisons cette technique dans certaines applications électroniques. Cette technique consiste à mettre l'électrode de détection à l'opposé de l'électrode alimentée. Cela oblige le champ électrique à faire le contour et donc à s'étendre pour pouvoir arriver perpendiculairement à la surface de l'électrode de détection. Cette fois-ci, on utilise que trois électrodes sans support ce qui permet de réduire la masse (+/- 140g/m).

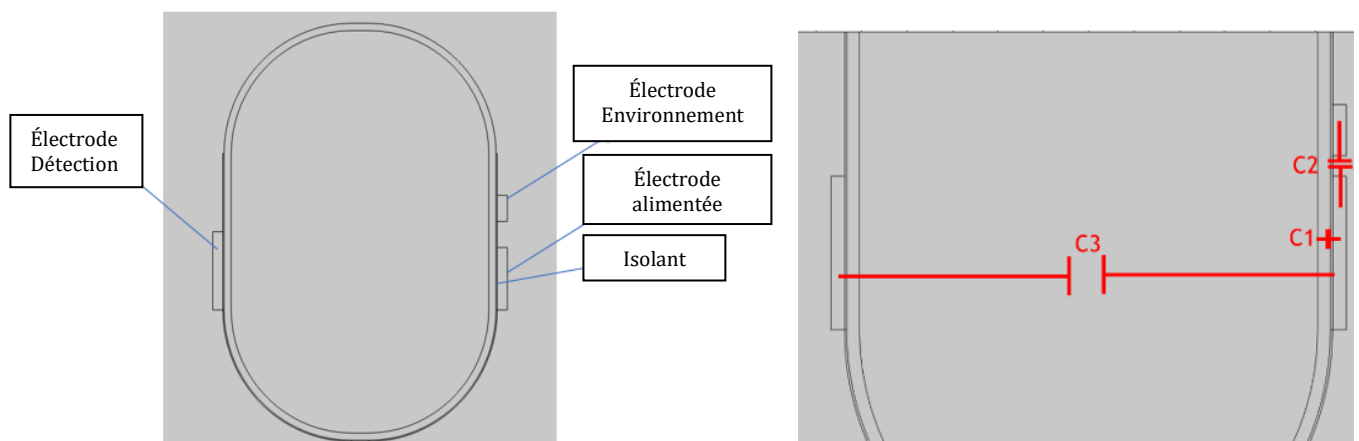


Figure 19 : Dispositions des électrodes et affichage des capacités du modèle AD HOC

Une fois le modèle réalisé, nous allons simuler une fois de plus les lignes afin d'avoir une idée de la répartition des lignes de champs et de la direction où l'intensité du champ est la plus importante. Nous présenterons trois cas différents, le premier sans obstacle, le deuxième avec la représentation de la tête à 16 cm et la troisième est similaire à la précédente, mais cette fois-ci, la tête se trouve à 25 cm.

Qu'est-ce que nous observons ?

Dans le cas où nous n'avons aucune perturbation (figure 20), on voit que les lignes de champ électriques sont bien réparties en dessous de la lisse et que l'intensité du champ électrique normalisé le plus important se trouve aussi en dessous. Ce qui est bon signe, car la zone de détection doit se situer sous la lisse et ceci va dans le bon sens. Cependant, ces vecteurs rouges sont normalisés et nous donnent donc seulement une indication d'où se trouve l'intensité la plus importante sans valeur chiffrée. Nous devons faire une petite analyse de la variation de la capacité pour pouvoir confirmer si ce dispositif peut détecter quelque chose ou pas. Avant de faire cela, introduisons notre pastèque dans le système à différentes distances afin d'observer l'impact qu'elle a sur les lignes de champs. À 16 cm (figure 21) et à 25 cm (figure 22), on voit bien que les lignes de champs varient à l'intersection de la pastèque. Faire descendre la tête encore plus bas risque en réalité de ne pas créer beaucoup de variation. Pour vérifier si ces variations de lignes de champs ont un impact réel sur la mesure de la capacité, nous avons donc décidé de mesurer les capacités principales du système.

$$\begin{aligned} C1 &= 1.51 \text{ nF/m [Parasite]} \\ C2 &= 30.1 \text{ pF/m [Environnement]} \\ C3 &= 11.2 \text{ pF/m [Mesure]} \end{aligned}$$

Maintenant que nous avons les différentes capacités du système, nous pouvons évaluer les variations de la capacité du système.

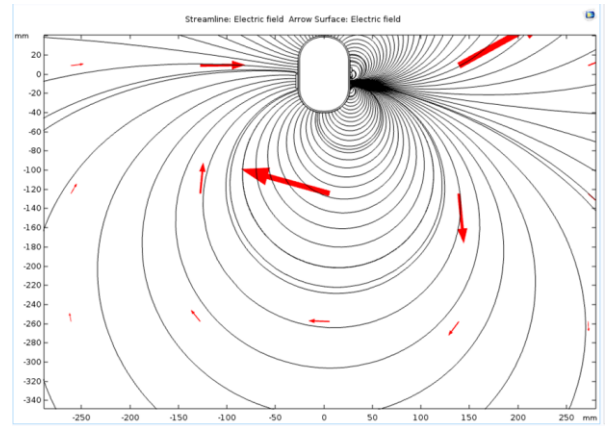


Figure 20 : simulation sans obstacle

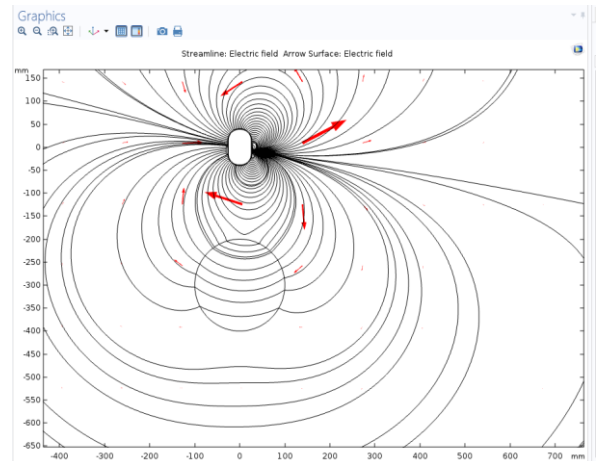


Figure 21 : simulation avec obstacle à 16 cm

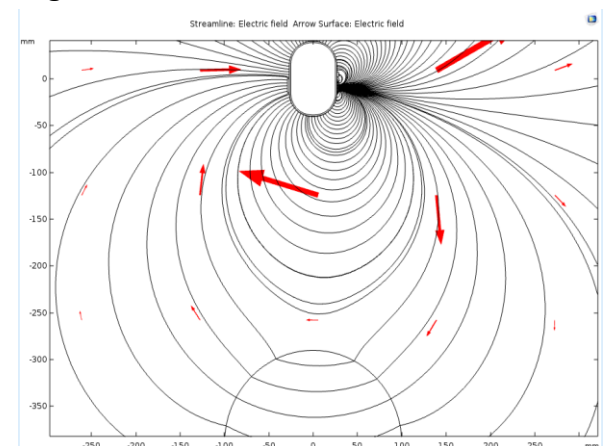


Figure 22 : simulation avec obstacle à 25 cm

Voici tableau 3 qui reprend les variations capacitives du système lorsque la pastèque s'éloigne de la lisse pour différentes mesures.

Distance inter [lisse-pastèque]	Densité d'E (J/m)	Capa. Tot. (pF)	Var. de capa (pF)
Sans obstacle	4512.96 E-10	1567.00	Réf.
1cm	4513.28 E-10	1567.11	0.11
6cm	4513.01 E-10	1567.01	0.01
11cm	4512.97 E-10	1567.00	0.00
21cm	4512.95 E-10	1567.00	0.00
31cm	4512.95 E-10	1567.00	0.00

Tableau 3 : Evaluation de la variation capacitive

On voit que lorsque la pastèque est à plus de 6 cm, les variations des capacités sont très faibles, voire inexistantes. Nous en avons conclu que cette technique avait l'air de premier abord très intéressante, mais les faibles variations de capacités la rendent hors-jeu. Nous devons donc trouver une technique plus intéressante.

Modèle 3 : Solution finale

Après de nombreux essais, nous avons enfin obtenu une solution qui est acceptable. Ce modèle est composé de seulement trois électrodes et il est basé sur le même principe que le précédent. Nous avons simplement disposé les électrodes sur la zone circulaire afin que celles-ci soient plus proches l'une de l'autre tel que représenté à la figure 23. Cela aura pour effet d'augmenter la variation capacitive comme nous allons le voir par la suite.

Ce modèle est le juste milieu entre la portée de détection et le taux de variation de la capacité. C'est le bon compromis, car nous voulons avoir des variations de capacité mesurable avec une distance de détection intéressante.

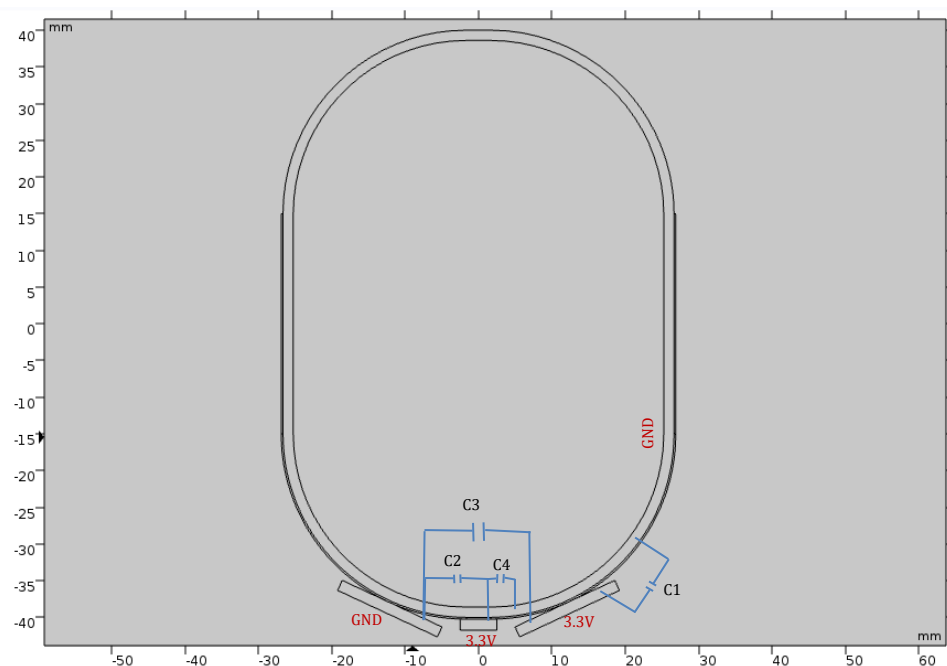


Figure 23 : Disposition des électrodes et affichage des capacités

Lignes de champs électriques

Une fois le modèle mis en place sur Comsol, nous avons lancé notre simulation afin d'observer les différents paramètres qui nous intéressent. Dans un premier temps, nous allons observer la répartition des lignes de champs électriques et leurs modifications lorsque la pastèque entre dans la zone de détection (< 20 cm). Dans un second temps, nous allons évaluer la variation de la capacité, c'est ce qui nous intéresse le plus.

La figure 24 représente assez bien la répartition spatiale du champ ainsi que l'intensité normalisée de celui-ci qui est représenté par des flèches.

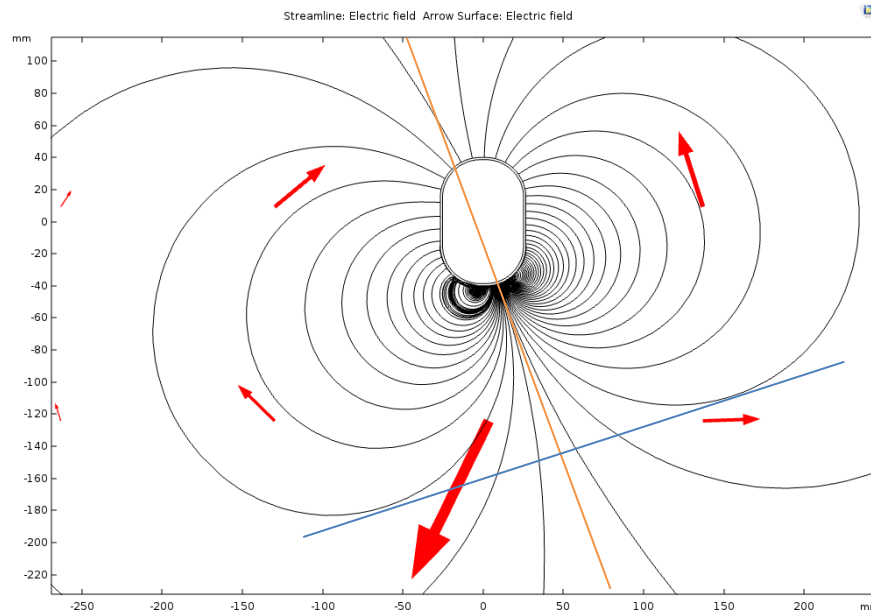


Figure 24 : Lignes de champs sans perturbation

On observe sur la figure 24 que les lignes de champs sont fortement concentrées près des électrodes, car c'est là où l'intensité du champ est la plus importante. De plus, nous voyons la grande flèche rouge située en bas à gauche nous indiquant l'endroit où l'intensité et la direction du champ est le plus fort. Cela s'explique par le fait que nous ne sommes plus dans un système symétrique et équilibré comme dans le premier modèle. Pour rappel, du côté droit et au centre nous avons les électrodes alimentées et du côté gauche nous avons celle qui est mise à la masse et cela crée un déséquilibre de la répartition des charges. De plus, nous pouvons le voir sur la figure 24. La symétrie ne se fait plus selon l'axe y, mais elle se fait selon l'oblique représentée par la droite orange. Cela signifie aussi que le côté gauche sera plus réactif, car les lignes de champs les plus concentrées, donc les plus intenses, sont plus bas que celles du côté droit (voir ligne bleue).

Regardons quelle influence aura notre pastèque sur les lignes de champs. Pour cela nous avons décidé comme précédemment de mettre la pastèque à deux endroits différents sous la lisse comme le montre la figure 25.

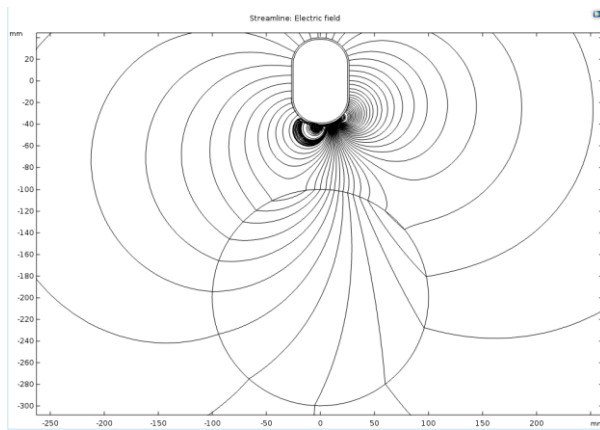


Figure 25 : Présence de la pastèque à 6 cm de la lisse

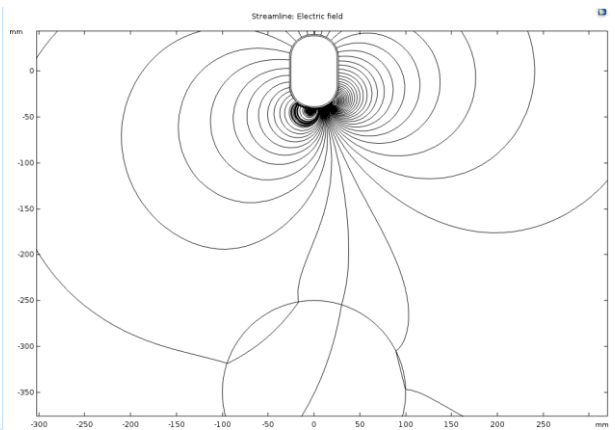


Figure 26 : Présence de la pastèque à 21 cm de la lisse

Sur la Figure 25, nous observons qu'une grande partie des lignes de champs sont impactées par la pastèque, ce qui présage une variation importante de la capacité totale du système. Tandis que lorsque cette même pastèque se trouve à 21 cm, voir figure 26, nous voyons qu'elle a beaucoup moins d'impact sur les lignes de champs ce qui signifie que la variation capacitive sera un peu moins importante.

Faisons un petit zoom, pour voir ce qui se passe à proximité des électrodes. (Figure 27)

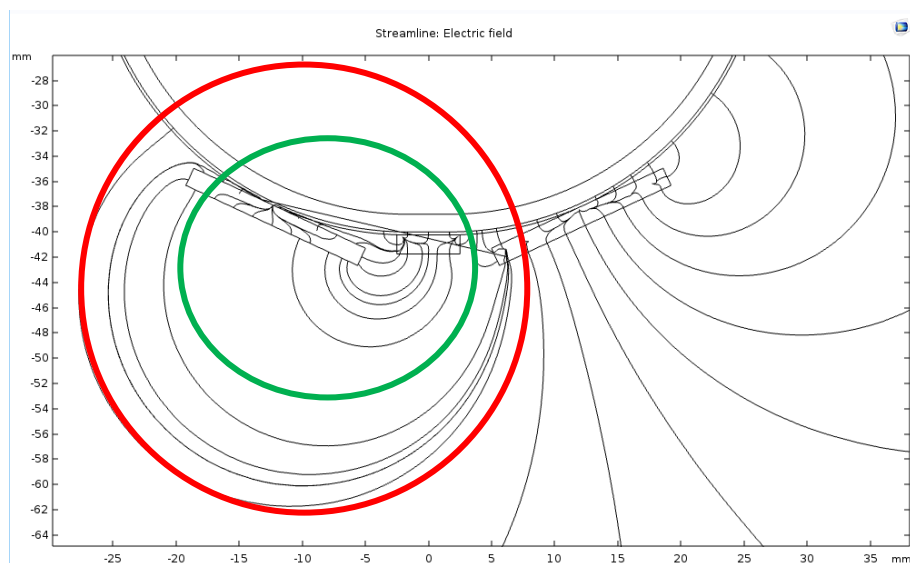


Figure 27 : Lignes de champs électriques proches des électrodes

On remarque dans la figure 27 que les lignes de champs électriques de la capacité C3(zone rouge) enclavent les lignes de champs de la capacité C2 (zone verte). Ce qui implique en cas de présence d'obstacles de rentrer tout d'abord en contact avec les lignes de champs électriques correspondant à la capacité de mesure ce qui stoppera la lisse. Les lignes de champs se trouvant dans cette zone verte ne seront impactées seulement dans le cas où il y a un changement dans l'environnement ou un dépôt sur la surface des électrodes.

Variation capacitive

Dans cette section, nous allons mesurer les variations capacitatives en fonction de la distance séparant notre lisse et l'obstacle. Cela va permettre de confirmer l'intuition que nous avons eue en observant les lignes de champs. Pour mesurer cette variation, nous devons récupérer la densité d'énergie électrique et calculer à partir de cela les capacités comme fait pour les modèles précédents.

Évaluons tout d'abord les différentes capacités non négligeables du système :

$$C1 = 402.9 \text{ pF}$$

$$C2 = 32 \text{ pF}$$

$$C3 = 19.3 \text{ pF}$$

$$C4 = 175.2 \text{ pF}$$

C1 et C4 sont les deux capacités les plus importantes, car la distance séparant leurs électrodes et la lisse qui est à la masse est très faible. C2 est un peu plus important que C3, car elle est plus proche de l'électrode de masse. (Voir figure 23 pour la disposition des électrodes).

Nous pouvons observer les variations capacitatives de ce modèle au tableau 3 :

Distance	Densité d'E (J/m)	Capa. Tot. (pF/m)	Var. de capa (pF/m)
Sans obstacle	1667.83 E-10	579.11	
1cm	1676.92 E-10	582.26	3.15
6cm	1668.19 E-10	579.23	0.12
11cm	1668.01 E-10	579.17	0.06
21cm	1667.84 E-10	579.12	0.01
31cm	1667.84 E-10	579.11	0.00

Tableau 3 : Evaluation de la variation capacitive

Nous observons sur le tableau 3, qu'à 1 cm de la lisse, la variation de la capacité est assez intéressante, de l'ordre de 3pF. Puis dès que nous nous éloignons un peu, nous avons tendance à décroître très rapidement, car les lignes des champs sont moins impactées. C'est ce que nous observions précédemment.

Il est vrai que les variations sont assez faibles par rapport à la capacité totale, mais celui-ci reste le meilleur modèle que nous avons pu mettre en place après plusieurs mois de recherche.

Ce modèle confirme qu'il peut y avoir des variations de capacité à 21cm, mais il ne faut pas oublier que dans ce modèle 2D nous négligeons les effets de bord et nous considérons la tête comme un tube rempli d'eau qui tend vers l'infini. Afin de vérifier si ce modèle est un minimum viable, nous avons réalisé une simulation 3D qui tient en compte ces effets de bord et qui considère la tête humaine comme une sphère de 10cm de rayon remplis d'eau.

En conclusion, les modèles 2D s'avèrent très efficaces, car ils demandent peu de ressources et nous donnent une idée très rapide du modèle imaginé, contrairement au 3D qui lui prend plus

de temps à être réalisé et simulé. Maintenant que nous avons un modèle 2D acceptable, il suffit de le vérifier en 3D, ce qui sera réalisé à la section suivante.

4.4. Modèle 3D

Comme dit précédemment, ce modèle 3D va nous permettre de confirmer notre modèle 2D sélectionné. Ce modèle va tenir compte des effets de bord ainsi que des dimensions réelles, ce qui va nous permettre de mieux nous rapprocher de la réalité. Pour réaliser ce modèle 3D, nous faisons une extrusion du modèle 2D et nous redéfinissons quelques paramètres qui seront décrits en annexe A. Dans cette section, nous allons travailler sur un modèle de lisse d'un mètre, étudier le maillage, le champ électrique ainsi que les variations de capacité.

Maillage

Le fait que nous utilisons une lisse d'un mètre est dû au fait que pour des lisses plus grandes, le simulateur a du mal à calculer car les maillages ne sont pas assez fins. C'est ce qui crée des bugs dans le logiciel. Ces bugs sont apparus au début avec la lisse d'un mètre, mais il a pu être résolu. Nous avons deux types d'objets, d'un côté nous avons la tête et l'air qui entoure la structure qui sont des structures assez importantes et de l'autre côté nous avons la lisse et les barrettes, qui sont très fines par rapport au reste. Pour éviter ces problèmes, nous avons compris qu'on ne pouvait pas avoir un maillage de même taille pour toutes les structures. Nous avons donc réalisé un maillage différencié, ce qui signifie que les grosses structures sont maillées avec un maillage normal et les fines par un maillage extrêmement fin. Une fois ce maillage mis en place, nous pouvons simuler toutes les tailles de lisses jusqu'à un peu plus d'un mètre. Il sera quasiment impossible de simuler les lisses de deux ou trois mètres, car le maillage actuel utilisé sur la lisse est le plus fin proposé par Comsol. Nous pouvons observer cette différence de maillage sur la figure 28.

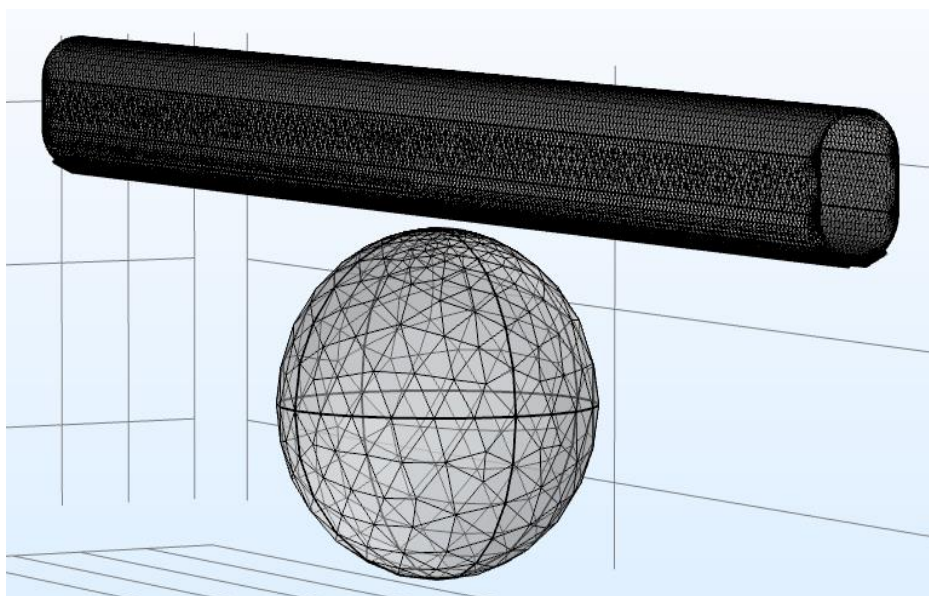


Figure 28 : Maillage du modèle 3D

Lignes de champs électriques

Avant de pouvoir observer les lignes de champs, nous avons dû, dans un premier temps, redéfinir toutes les contraintes physiques comme la tension par exemple. Du modèle 2D, nous ne récupérons que la forme de base.

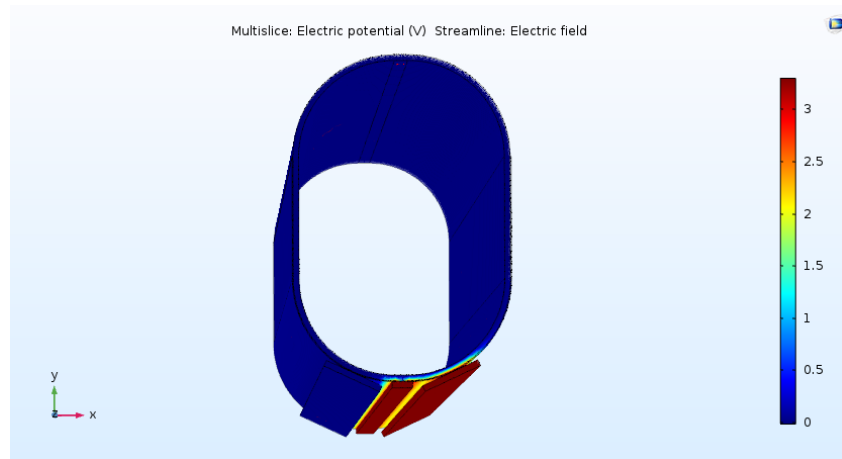


Figure 29 : Modèle 3D & affichage du potentiel électrique

Une fois les électrodes alimentées, nous pouvons afficher les lignes de champs que nous pouvons observer sur la figure 30.

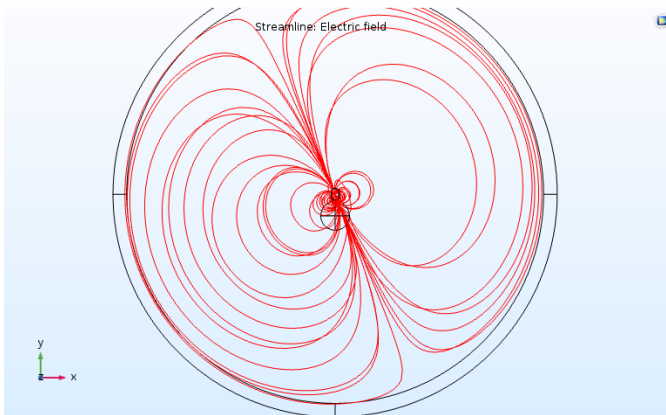


Figure 30 : Lignes de champs électriques du modèle 3D

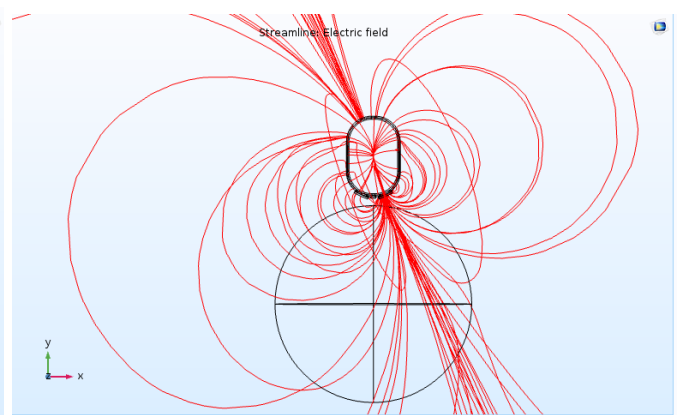


Figure 31 : Zoom sur les Lignes de champs électriques du modèle 3D

Nous observons sur la figure 30 que les lignes de champs sont plus nombreuses et plus éloignées que dans le cas 2D. Cela s'explique par le fait que lors de la simulation 3D, Comsol aura moins tendance à négliger les lignes de champs électriques à faible intensité. Nous pouvons observer cette faible intensité grâce aux écarts importants entre les lignes de champs une fois que nous nous éloignons de la lisse. Si nous faisons abstraction des lignes de champs à faible intensité et que nous faisons un zoom, nous tombons sur la figure 31 qui ressemble fortement au résultat que nous avons obtenu avec le modèle 2D.

Tout comme le modèle 2D, nous avons le lobe de gauche qui est plus concentré et plus orienté vers le bas tandis que le lobe de droite, lui est un peu moins concentré et plus orienté vers le haut. Comme dit dans la section 2D, le fait d'alimenter les deux électrodes (centre & droite) et de mettre celle de gauche à la masse, crée une asymétrie au niveau de la répartition des charges, ce qui influence in fine nos lignes de champs.

La différence notoire, est le fait que nous observons des lignes de champs à l'intérieur de la lisse, ce qui n'était pas le cas auparavant. Cela s'explique par le fait que nous tenons compte maintenant de l'effet de bord tel qu'on le voit dans la figure 32.

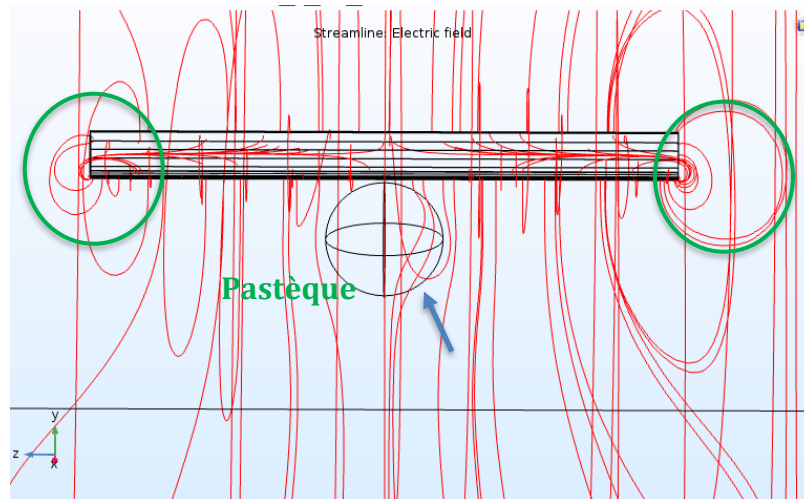


Figure 32 : Effet de bord & perturbation des lignes de champs

Comme le montrent les cercles verts, nous avons les effets de bord qui ont tendance à faire une boucle à la fin. Ces lignes de champs électriques rejoignent en majorité la lisse par l'intérieur, car c'est la masse la plus proche, ce qui implique une présence de ligne de champs à l'intérieur. Lorsque nous étirons la lisse à l'infini comme défini dans le cas 2D, les effets de bord disparaissent et il n'y a donc plus de lignes de champs à l'intérieur.

De plus grâce à la figure 32, nous pouvons dire que les perturbations qu'apporte la tête sont locales. On voit bien que les lignes de champs qui rencontrent la tête sont distordues localement (flèche bleue).

Nous avons pu observer grâce à ces simulations que la tête peut engendrer des perturbations locales. Est-ce que ces perturbations sont détectables ? C'est ce que nous allons observer dans la prochaine sous-section.

Variation capacitive

Dans cette partie, nous allons observer si les variations de capacités sont acceptables ou pas. Faire une simulation 3D prend énormément de temps, c'est pourquoi nous nous concentrerons sur les valeurs importantes. Nous allons dans un premier temps mesurer la capacité totale du système sans aucune perturbation. Par la suite, nous mesurerons cette même capacité, mais en plaçant la tête à 11 et 21 cm. Les résultats de ces mesures se retrouvent dans le tableau 4:

Distance (cm)	Densité d'Energie (J)	Capa. Tot. (pF)	Var. de capa (pF)
Sans obstacle	3026.98 E-11	105.10	/
11	3109.85 E-11	107.98	2.88
21	3108.51 E-11	107.93	2.83

Tableau 4 : Evaluation de la variation capacitive du modèle 3D

Nous constatons que la densité d'énergie ainsi que la capacité totale du système ont diminué d'un facteur cinq par rapport à lisse dans le modèle 2D. Contrairement à ces deux valeurs, la variation de capacité augmente par rapport au modèle 2D. Ces variations inter-modèles sont difficiles à expliquer. Mais ce qui est certain, c'est que le fait de considérer la tête comme une sphère et de tenir compte des effets de bord rend les résultats différents. Dans le cas 3D et 2D nous avons de bonnes variations capacitives mesurables, cela nous permet d'avoir une première idée de ce que nous allons implémenter.

4.5. Piste d'amélioration

Comsol est un logiciel qui offre de nombreuses fonctionnalités. Il est possible de créer des environnements extrêmes et de voir comment notre dispositif réagit. Pour cela il faut ajouter des paramètres comme l'humidité, la température, la pression. Mais comme tout logiciel de simulation, chaque paramètre ajouté rend la simulation encore plus lente et demande plus de ressource de calcul.

Dernièrement, nous avons découvert qu'il est possible d'utiliser un algorithme génétique développer à l'UCL combiner à Matlab afin de pouvoir trouver les positions optimales pour nos barrettes d'aluminium. Cette optimisation du positionnement ne nous aurait pas beaucoup aidés, car monter les barrettes d'aluminium sur la lisse s'avère très complexe. Il faut savoir que les barrettes arrivent un peu tordues à cause du transport et le monter précisément sur une surface circulaire est quasiment impossible dans le cas d'un prototype. De plus comprendre cet algorithme demande un travail important de plusieurs semaines.

Cependant ces outils peuvent être utilisés plus tard pour améliorer la simulation et la rendre plus complète ce qui la rapprochera de la réalité.

Par manque de temps et par envie de passer au prototypage, nous avons décidé d'arrêter les simulations après la simulation 3D et de construire notre prototype afin de réaliser directement les tests dessus.

Afin d'avoir un prototype fonctionnel, nous allons voir dans la section suivante le design et le choix du hardware qui va permettre de mesurer et analyser la variation capacitive.

5. Hardware

Dans cette section, nous allons aborder les différents composants hardware. Dans un premier temps, nous allons parler du capteur en lui-même, puis nous allons voir les circuits d'acquisition et d'alimentation, enfin nous finirons par les systèmes de mesures et d'analyses. Cette section va nous donner une bonne idée du processus suivi pour implémenter ce hardware.

5.1. Capteur

Le capteur implémenté dans notre solution est un capteur capacitif quasi coplanaire avec une garde intégrée qui est représentée par la lisse. Ce capteur est intégré à la lisse et celle-ci participe au bon fonctionnement du capteur. C'est pour cela que nous avons décidé de considérer la lisse comme un capteur et nous l'avons nommé « capalisse ».

Nous avons construit deux capalisses, une d'un mètre et une autre de deux mètres. La première a été réalisée afin de pouvoir développer le hardware de base et la première version du software. Une fois le hardware développé et que les premiers résultats allaient dans le bon sens, nous avons alors développé la deuxième solution. Cette seconde capalisse va nous permettre de peaufiner notre algorithme afin d'observer les variations de fréquence lorsque la capalisse est en mouvement.

5.2. Circuit oscillateur

Ce circuit est très important, car il va permettre de traduire la variation de la capacité par une variation de fréquence. Lorsque quelque chose se rapproche de la capacité, cela change la permittivité relative ce qui à son tour change la capacité et donc la fréquence d'oscillation du circuit.

Il existe plusieurs types de circuits oscillateur, les plus connus sont les LC qui sont généralement utilisées dans les détecteurs de métaux. Ce genre d'oscillateur est composé d'une capacité et d'une inductance qui entrent en résonance. La fréquence d'oscillation est fournie par $f = \frac{1}{2\pi\sqrt{LC}}$, dès que C ou L varient cela change la fréquence. Donc ceci peut être utilisé pour mesurer les variations d'inductances comme les variations de capacités.

Mais nous avons décidé d'opter pour un circuit qui n'utilise pas d'inductance, car nous sommes très vite limités par la taille de cette dernière. Vu que nous devons mesurer la variation de capacité, il existe une solution plus simple qui n'utilise que des capacités, des résistances et un NE555. Cette solution consiste à faire un montage d'un NE555 en mode astable comme nous pouvons le voir à la figure 33.

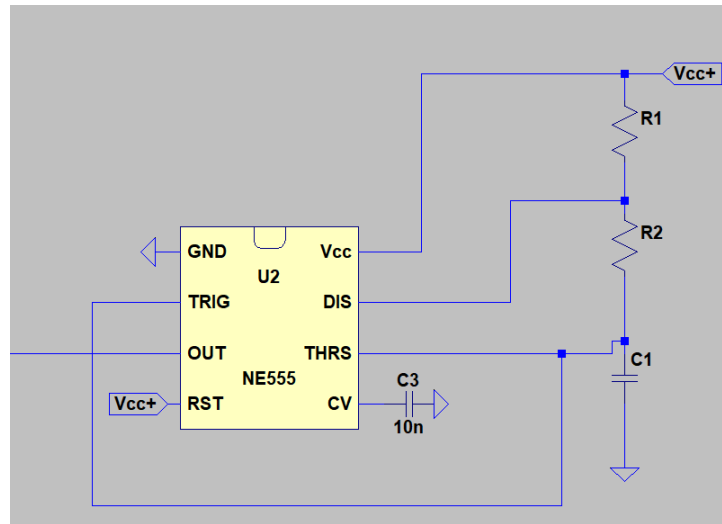


Figure 33 : Oscillateur NE555

Cet oscillateur est très intéressant, car la fréquence ne dépend que de la capacité C1 et des résistances R1 & R2. Ce qui nous permet de monter en fréquence beaucoup plus facilement qu'avec un circuit LC, car ici nous dimensionnerons le circuit avec des résistances et non pas avec des inductances. Comme nous le savons, l'intervalle de valeurs offert par les résistances est plus important que celui offert par les inductances ce qui nous donne un plus grand degré de liberté.

La fréquence de ce type de circuit est donnée par :

$$f = \frac{1.44}{(R_1 + 2R_2)C_1}$$

C_1 est connue, car elle est fixée par la capalisse tandis que R_1 & R_2 sont inconnues. Étant donné que nous avons deux inconnues il nous faut une deuxième équation qui est fournie par l'équation du dutty-cycle qui est la suivante :

$$\alpha = \frac{R_1 + R_2}{R_1 + 2R_2}$$

Le rapport cyclique est représenté par le α . En le fixant à 0.49 nous aurons un temps 'High' quasiment égale au temps 'Low'. À présent, nous fixons la fréquence désirée afin d'obtenir les valeurs des résistances.

Dans un premier temps, nous avons décidé de vérifier si le circuit serait capable de détecter de faible variation de capacité à l'aide du logiciel LTspice. Nous avons choisi la fréquence de 40khz et nous avons pris la capacité mesurée du modèle réduit de la capalisse ($C_{mes} = 56pF$ & $C_{env} = 96pF$).

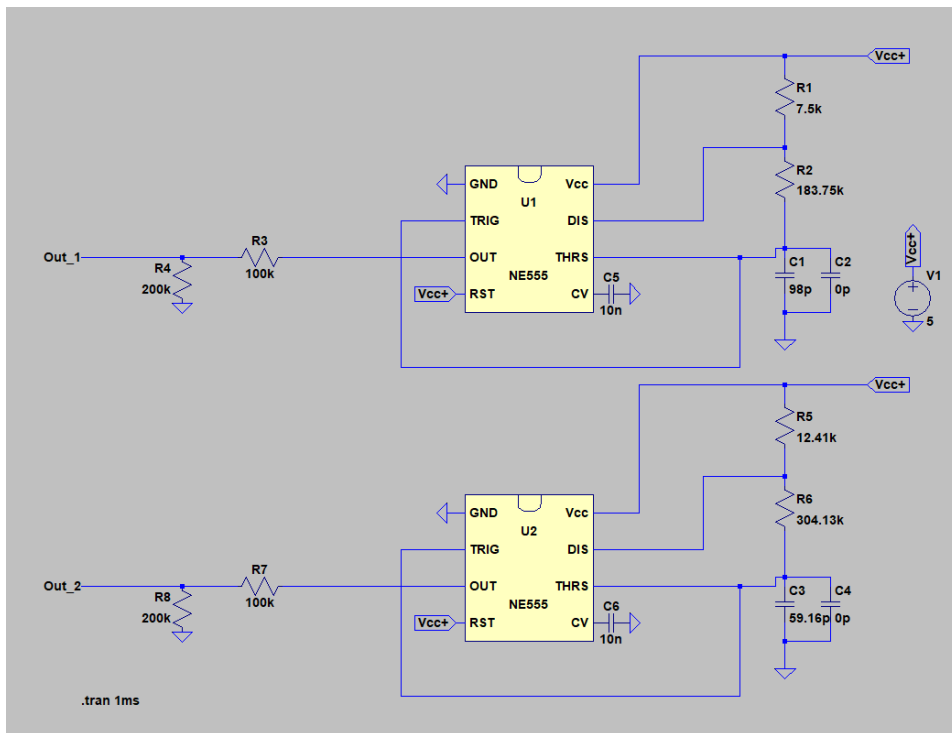


Figure 34 : Circuit implémenté pour la capalisse

Comme nous pouvons le voir sur la figure 34, nous avons deux oscillateurs, le premier va mesurer la capacité d'environnement et le second celle de détection. Ils sont totalement découplés mis à part l'alimentation qui est à la même source et le fait d'avoir la même masse. Les petites capacités en parallèle simuleront les variations des capacités. Elles sont mises à 0 picofarads par défaut.

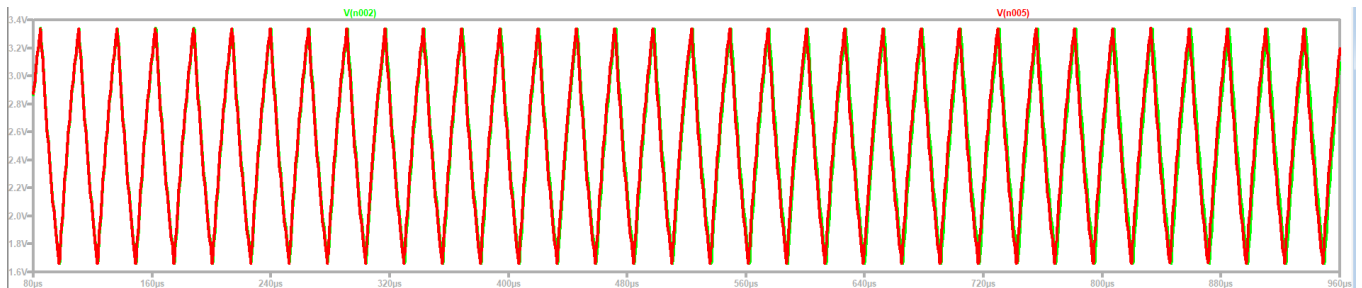


Figure 35 : Signaux en phase aux bornes de la capacité C3 & C1

Nous observons que nos oscillateurs sont en phase. Cela signifie que les capacités sont en phase et par conséquent les sorties sont également en phase. Comme nous pouvons l'observer sur les figures 35 et 36. (le vert=environnement et le rouge=mesure)

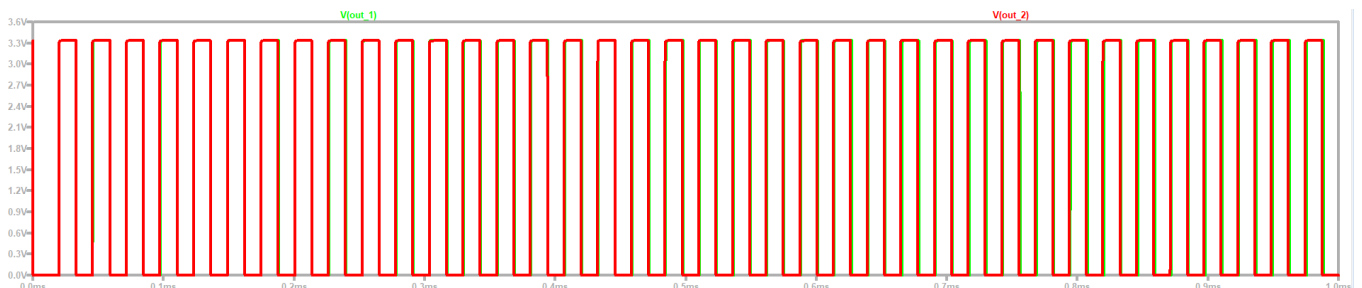


Figure 36 : Signaux en phase à la sortie des oscillateurs

Le fait de les mettre en phase n'est pas important dans le cadre de la capalisse, le but ici étant de mieux voir l'impact qu'aura une variation de capacité.

Introduisons maintenant une faible variation de capacité de 0.5 pF et observons sur la figure 37 et 38 l'impact de cette modification.

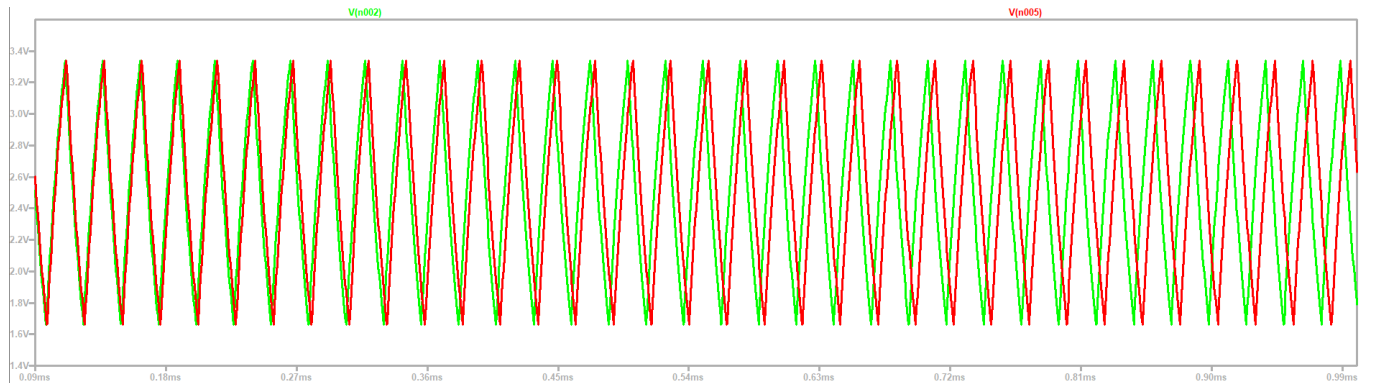


Figure 37 : Signaux déphasés aux bornes de la capacité C3 & C1

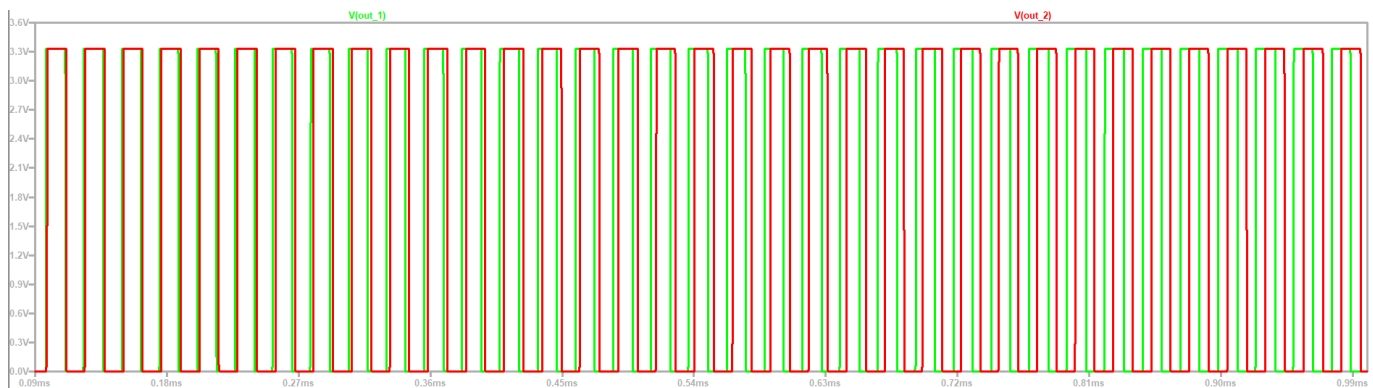


Figure 38 : Signaux déphasés à la sortie des oscillateurs

Nous voyons clairement qu'avec une faible variation de capacité de 0.5pF nous perdons environ 250Hz ce qui est intéressant à développer dans le cadre de notre application.

Pour notre circuit final, nous allons monter un peu plus haut en fréquence afin de rendre notre circuit encore plus sensible. Il faudra pour cela trouver le juste milieu en ce qui concerne la sensibilité. Le rendre trop sensible veut dire qu'une petite goutte ou quelque chose de très petit induira une variation de fréquence importante et dans le cas où le circuit serait peu sensible, cela voudrait dire que la capalisse n'aurait pas beaucoup de variation de fréquence ce qui rendrait la détection plus lente.

Cette fréquence sera choisie lors des tests, il suffira de changer de résistance et de voir le moment où le dispositif sera optimum.

Cette partie nous a confirmé que l'oscillateur que nous voulions utiliser sera fonctionnel pour notre capalisse. Mais encore faut-il faire un choix de fréquence qui elle sera déterminée par la suite. Afin d'avoir un système stable pour le prototype final, nous avons décidé de concevoir un PCB qui se trouve en annexe B. Sur la figure 39 nous pouvons voir le circuit de notre oscillateur.

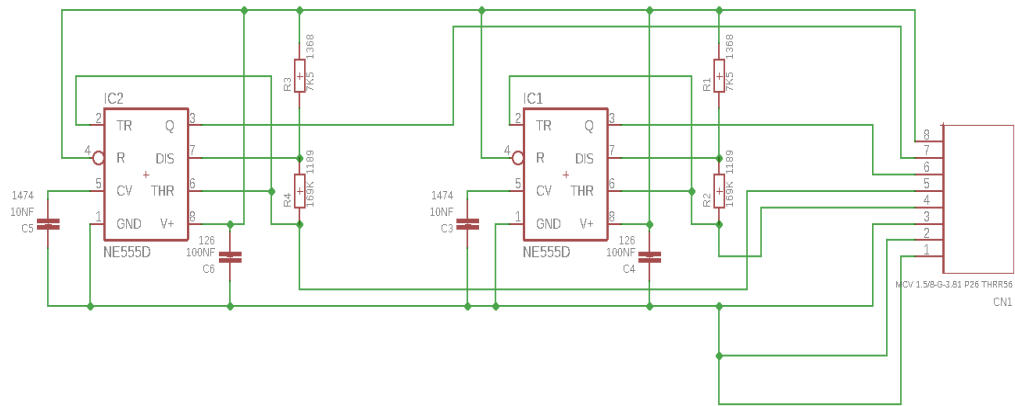


Figure 39 : Circuit Eagle des oscillateurs

5.3. Circuit d'interface & d'alimentation

Ce circuit à deux rôles importants, le premier rôle est de relier le FPGA au circuit oscillateurs. Et le deuxième est d'alimenter tout le dispositif. Cette carte va recevoir du 24V venant de l'alimentation de la barrière et elle transformera cette tension en 5V et en 3.3V afin d'alimenter tous les systèmes. Cette carte contient une 'Logic Level shift bidirectionnel' qui va mettre tous les signaux entrant dans le Deo-nano à 3.3V et tous ceux qui en sortent à 5V. Cela sera utile pour contrôler un relais à partie du Deo-nano ou pour envoyer les signaux venant de l'oscillateur qui sont à 5V dans le FPGA.

Pour ce circuit (figure 40), il y a aussi eu un design de PCB qui a été fait et qui se retrouve dans l'annexe B. Nous tenons à souligner que le bloc d'alimentation 24V-5V a été fourni par les ingénieurs d'A.S (figure 41), qui ont l'habitude de le mettre dans la plupart de leurs circuits.

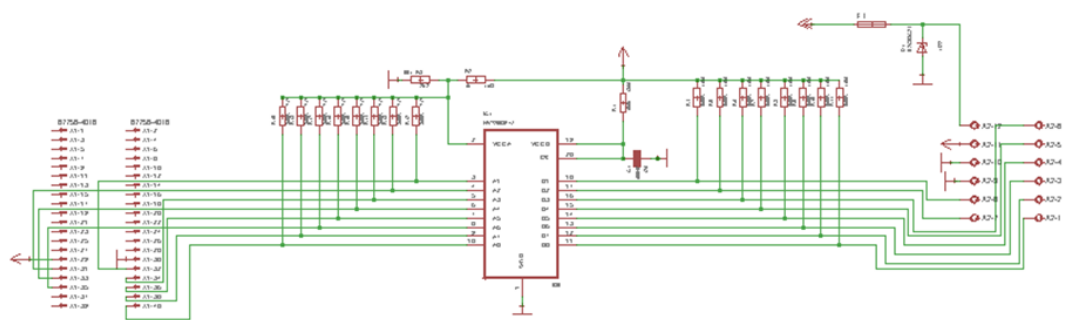


Figure 40 : Circuit Eagle de la carte d'interface

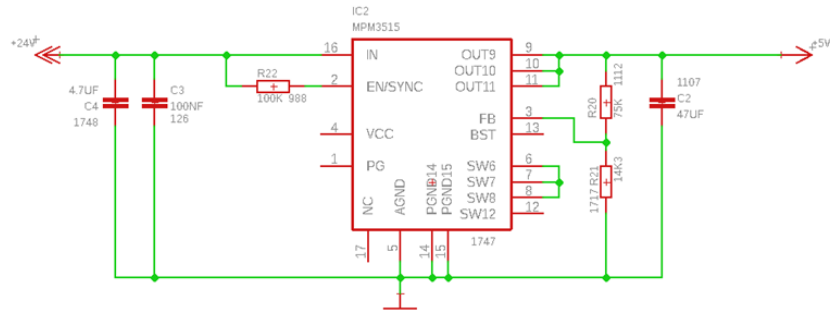


Figure 41 : Buck de chez AS (24v → 5V)

5.4. FPGA : Deo-Nano

Le Deo-Nano est un FPGA que nous avons eu l'occasion d'utiliser dans le cadre de certains cours dispensés à l'EPL. Ce FPGA va nous servir de compteur, mais pas seulement. Il pourra aussi envoyer des bits pour contrôler par exemple un transistor ou un relais et il permettra aussi de communiquer avec le Raspberry-pi via SPI. Mais comme vous le comprenez son activité principale est de compter et d'envoyer ces informations au Raspberry PI. Le Deo-nano nous permet vraiment de n'avoir aucune limite de fréquence, car il peut monter jusqu'à 50 MHz. De plus, nous pouvons le programmer tel que désiré, par exemple dans notre cas nous comptons les flancs montants et descendants ce qui augmente la résolution d'un facteur 2 par rapport à certains compteurs.

Il y a certainement d'autres possibilités, comme utiliser des microcontrôleurs utilisés chez A.S. mais cela demanderait un temps de pris en main assez important. Les responsables de chez A.S nous ont suggéré d'utiliser les outils avec lesquels nous étions les plus à l'aise, car finalement si le prototype fonctionne tout le dispositif serait redessiner pour qu'ils puissent être industrialisé.

Donc le choix du FPGA était un choix d'outils que nous connaissions et qui pouvait faire le travail demandé. De plus, celui-ci partage un écran avec le Raspberry-pi sur lequel nous affichons des informations utiles en temps réel (figure 42). Un autre intérêt non négligeable, c'est qu'avec le Deo-nano une fois le soft terminé nous pouvons générer un circuit en porte logique qui pourra être implémenté sur un PCB plus tard.

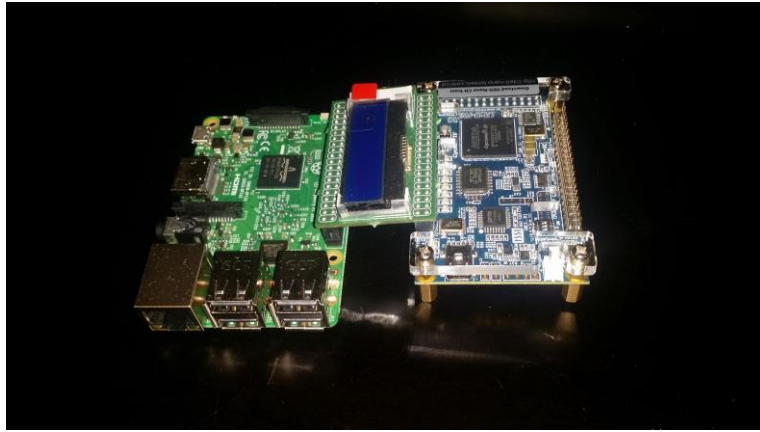


Figure 42 : Raspberry-Pi + écran + FPGA(Deo-Nano)

5.5. Raspberry Pi 3

Dans le cadre de ce travail, nous avons utilisé un Raspberry Pi 3 qui se trouve à gauche dans la figure 42. Le Raspberry est un nano-ordinateur équipé d'un processeur ARM tournant à 1 GHz. Celui-ci est très intéressant pour de petits projets où on a besoin d'une petite puissance de calcul.

Il est possible de programmer le Raspberry-Pi avec différents langages, mais notre décision s'est portée sur le python étant plus adapté au Raspberry-Pi.

Notre Raspberry-Pi contiendra notre software qui fera l'analyse des variations des fréquences et qui décidera d'arrêter ou non la lisse. Il représentera donc le cerveau de notre dispositif.

Dans la section suivante nous allons voir les différents algorithmes que nous avons mis sur le Raspberry-Pi afin de pouvoir détecter les usagers vulnérables lors de la fermeture de la lisse.

6. Software

Dans cette section nous allons discuter principalement de la dernière version de l'algorithme qui tourne sur le RPI. Avant cette version, plusieurs codes ont été développés, ceux-ci étaient améliorés au fur et à mesure jusqu'à atteindre ce dernier algorithme.

Nous diviserons l'algorithme en différentes parties, la première est celle qui fait le rapport des deux signaux afin d'annuler les variations dues à l'environnement, la deuxième partie sera consacrée au filtre mis en place pour mieux lisser le signal et la 3^e partie sera consacrée à la détection.

6.1. Annulation des bruits environnementaux

Comme dit plus haut, le Raspberry Pi va recevoir le nombre de flancs montants et descendants de chaque oscillateur qui ont été comptés préalablement par le FPGA et envoyés via SPI. Nous avons deux oscillateurs ; un que nous appellerons oscillateur de mesure qui est l'oscillateur mesurant les perturbations sous la lisse donc notre tête, et l'autre que nous appellerons oscillateur d'environnement mesurant les effets de l'environnement.

On observe sur la figure 43 que les variations du nombre de tics provenant des oscillateurs varient de la même manière, nous aurions même tendance à dire que ce sont les mêmes, mais ce n'est pas le cas, car derrière ces signaux se cachent deux détections quasiment invisibles ici.

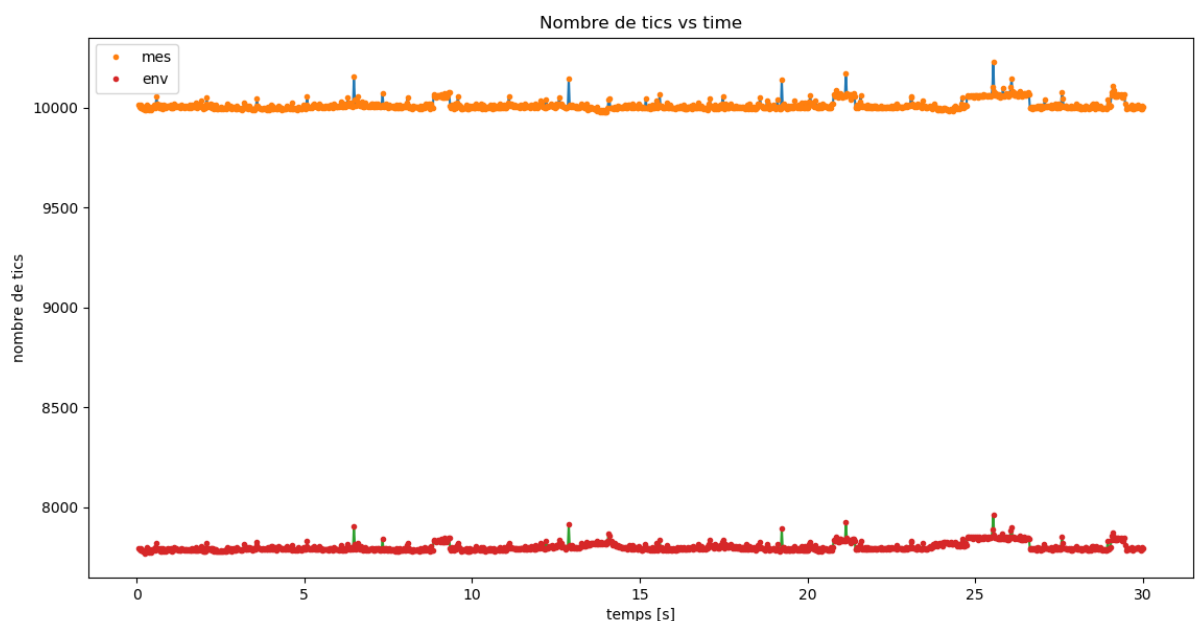


Figure 43 : valeurs du compteur en fonction du temps

Pour annuler ces effets non désirés de l'environnement, nous devons faire le rapport de ces deux signaux, car nous savons que ces variations agissent sur les capacités de la même manière. Mais lorsqu'une tête se rapproche de la lisse celle-ci ne va intervenir que sur une seule capacité et ne fera varier que la capacité de l'oscillateur de mesure. Cela changera donc le rapport des

signaux. Comme nous pouvons le voir sur la figure 44, les perturbations faites avec la tête changent fortement le rapport de fréquence (ou le rapport du nombre de tics). Ici, nous arrivons totalement à distinguer les variations induites par la tête du reste.



Figure 44 : Résultat du rapport des deux compteurs

À présent, nous arrivons à voir concrètement qu’il y a une détection. Cela dit, un problème persiste. L’algorithme n’est pas supposé avoir une vue globale du graphique. Il doit être capable de détecter cette variation dès le début et dans ce cas-ci, le signal est encore un peu bruité et il descend en zigzaguant. Pour pouvoir avoir une pente plus lisse nous devons réaliser un filtre afin de lisser le signal. C’est ce que nous allons voir dans la prochaine section.

6.2. Filtre software

Dans cette partie, nous allons voir les différents filtres que nous avons implémentés ainsi que notre méthode de sélection finale.

Il faut tout d’abord savoir que la récupération du compteur par le Raspberry Pi se fait chaque 25ms ce qui correspond à l’écart de temps entre chaque point. Nous prenons la valeur du compteur en ‘t’ et nous lui soustrayons celle en ‘t-1’ ce qui nous donne alors le nombre de tics pour ce laps de temps.

Pour filtrer ces valeurs, il y a plusieurs manières de faire comme utiliser un filtre de Kalman, faire la moyenne, faire la médiane... Nous avons décidé d’implémenter les deux dernières et de comparer. Quant au filtre de Kalman, c’est l’un des meilleurs filtres, mais le temps d’intégration est assez important et ne permet pas d’être très réactif dans le cadre de notre application.

Nous avons commencé par implémenter le filtre avec la moyenne. Celui-ci s'est avéré assez intéressant, beaucoup de gens l'utilisent comme filtre dans le domaine du traitement de signal. Il y a plusieurs moyens d'utiliser ce type de filtre, nous pouvons faire une moyenne glissante ou encore une moyenne de groupe. Ces deux types de filtres ont été implémentés et sont mis à la figure 45 et 46.

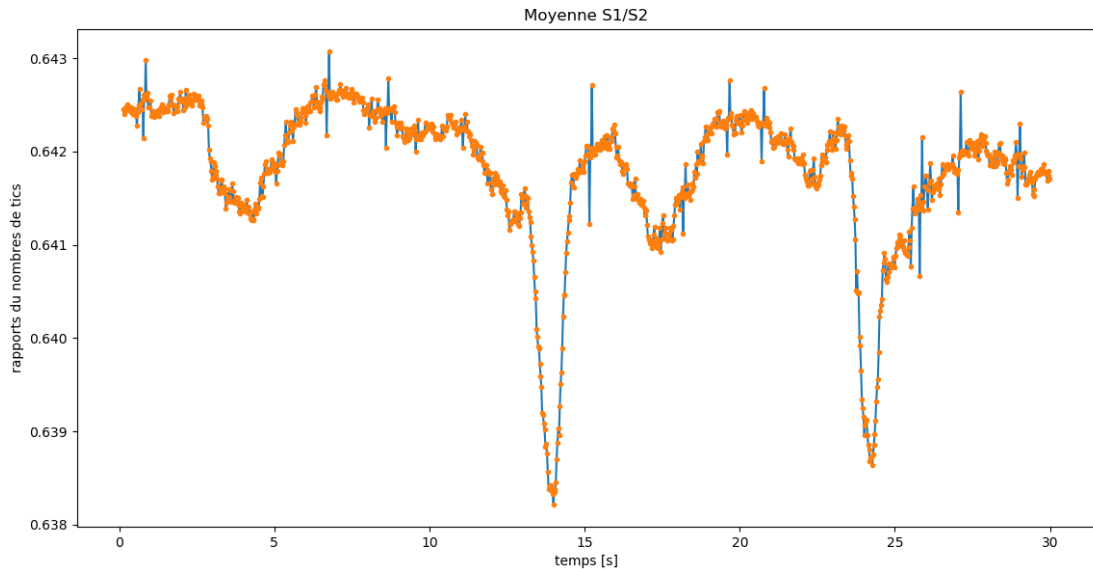


Figure 45 : Moyenne glissante du signal

La moyenne glissante est une moyenne où nous faisons rentrer un nouvel élément et où nous supprimons le plus ancien. Dans notre cas, cela ne nous intéresse pas trop, car lorsqu'il y a une variation d'un groupe de point, nous tenons à ce qu'elle soit la plus prononcée possible et pas progressif comme cela serait le cas ici. L'avantage de ce genre de moyenne est d'avoir des points chaque 25ms et donc d'être beaucoup plus réactif. L'inconvénient est que le filtre ne soit pas très puissant. Nous voyons qu'il reste encore un peu de bruit ainsi que quelques outliers persistants qui sont atténués.

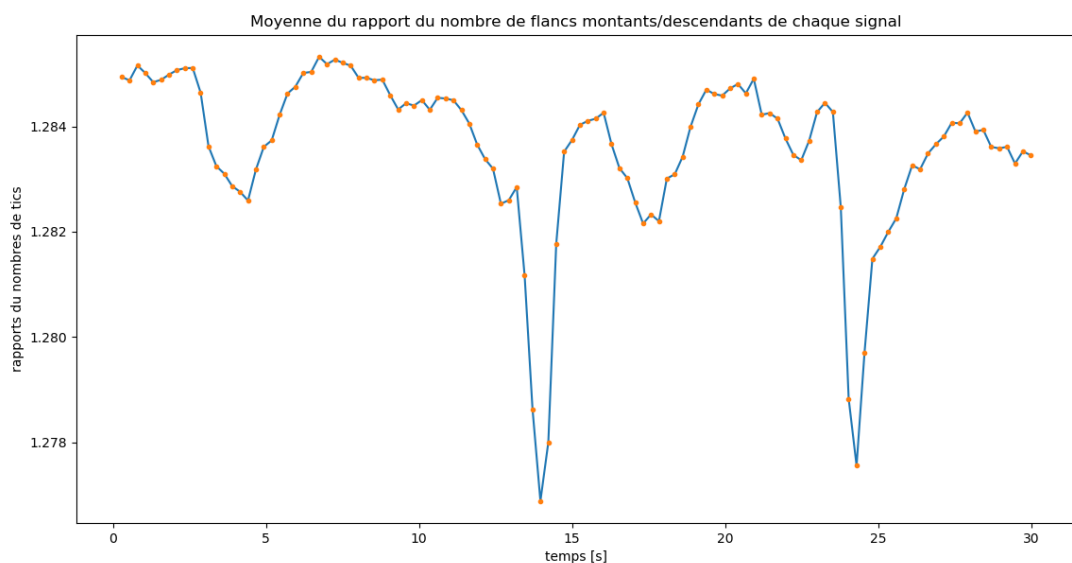


Figure 46 : Moyenne de groupe du signal

Ensuite, nous avons implémenté la moyenne de groupe (Figure 46), qui elle, a l'air assez stable. Nous observons bien que lorsqu'il y a un groupe de point qui chute, l'écart qui se crée devient important. Mais nous n'étions toujours pas convaincus, car l'influence des outliers se fait sentir dans une moyenne même si cela ne se voit pas trop dans ce cas-ci.

Prenons un vecteur de 4 points par exemple [6 6 6 0] ou encore [6 5 6 0], nous comprenons que le outlier soit le 0. Sachez que la moyenne donnera 4.5 et 4.25 tandis que la médiane donnera 6 et 5.5. La médiane arrive à faire abstraction des outliers et affiche une valeur assez correcte de la tendance des points. Nous concluons que les outliers ont peu d'impact sur la médiane.

Les résultats fournis par la médiane pour le même signal sont fournies sur la figure 47 :



Figure 47 : médiane du signal

Ici aussi, nous sommes à un intervalle de 100 ms. Nous aurons une réactivité plus faible que si nous utilisons une médiane glissante. Le fait de prendre la médiane rendra les chutes un peu moins importantes, mais c'est le prix à payer si nous voulons éviter d'être influencés par les outliers et d'avoir un filtre fort.

Comparons à la figure 48 la médiane de groupe et la moyenne de groupe. Nous observons qu'il y a de faibles différences entre les deux. On observe surtout que la médiane est un peu plus stable que la moyenne.

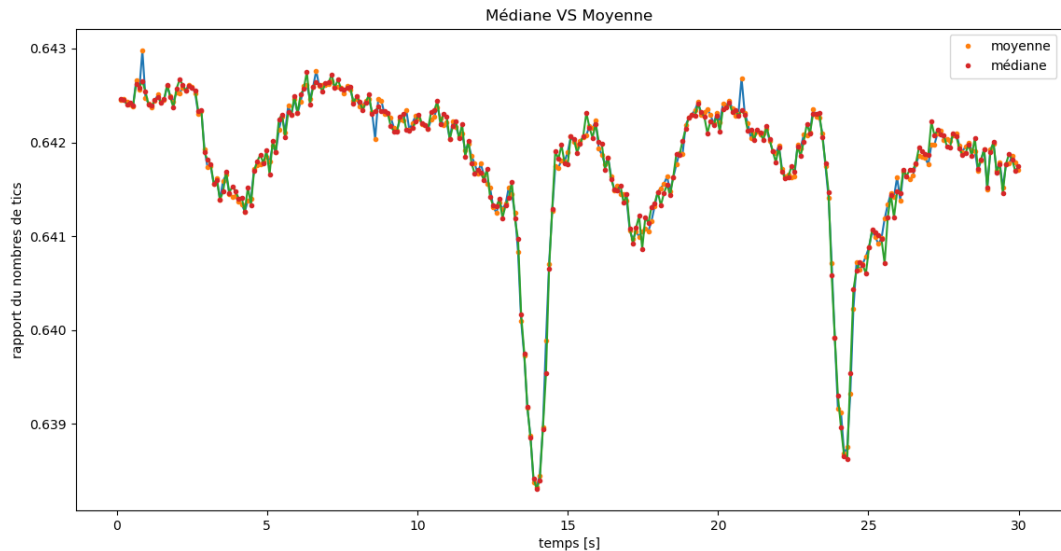


Figure 48 : médiane VS Moyenne

6.3. Détection

C'est l'une des parties les plus complexes, car il faut être capable de faire comprendre à l'algorithme qu'il y a une personne présente sous lisse suffisamment tôt que pour éviter un choc. Plusieurs algorithmes ont été mis en place, il ne me sera pas possible de tous les expliquer, mais nous n'allons aborder ici que les algorithmes les plus intéressants.

Technique de la variance

Cette technique analyse la variance du signal filtré. Lorsqu'il y a une détection comme nous pouvons le voir dans les images précédentes, l'écart entre les points augmente et donc cela augmentait la variance. La variance se fait sur les 4 derniers points du signal filtré. Elle est donc glissante. L'algorithme évalue la variance de ce groupe de point, si ceux-ci dépassent huit fois la médiane des variances comme nous pouvons le voir à la figure 49, il y a alors détection potentielle.

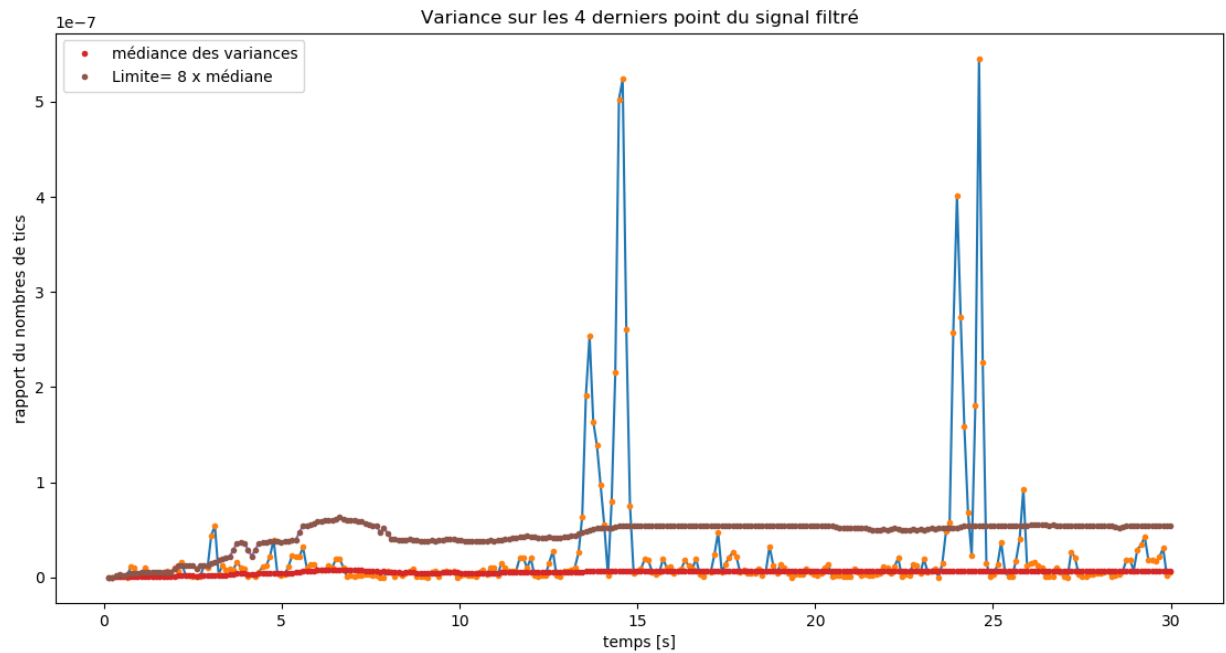


Figure 49 : Variance glissante sur 4 points du signal filtré

Afin d'éviter d'avoir des faux positifs, donc de fausses détections nous avons décidé d'accorder la détection seulement si 2 points consécutifs sont au-dessus de la limite.

Vous remarquerez aussi sur la figure 49 qu'au démarrage, le système est assez instable. Cela est dû au Raspberry-pi. Lorsque le software se lance nous pensons qu'il y a un ralentissement de l'horloge interne pendant quelques secondes. C'est pourquoi nous avons décidé de laisser les 10 premières secondes comme temps d'initialisation et de stabilisation du système. Ceci permet aussi à la médiane d'atteindre une valeur assez stable. Nous pouvons observer les bits détections sur la figure 50.

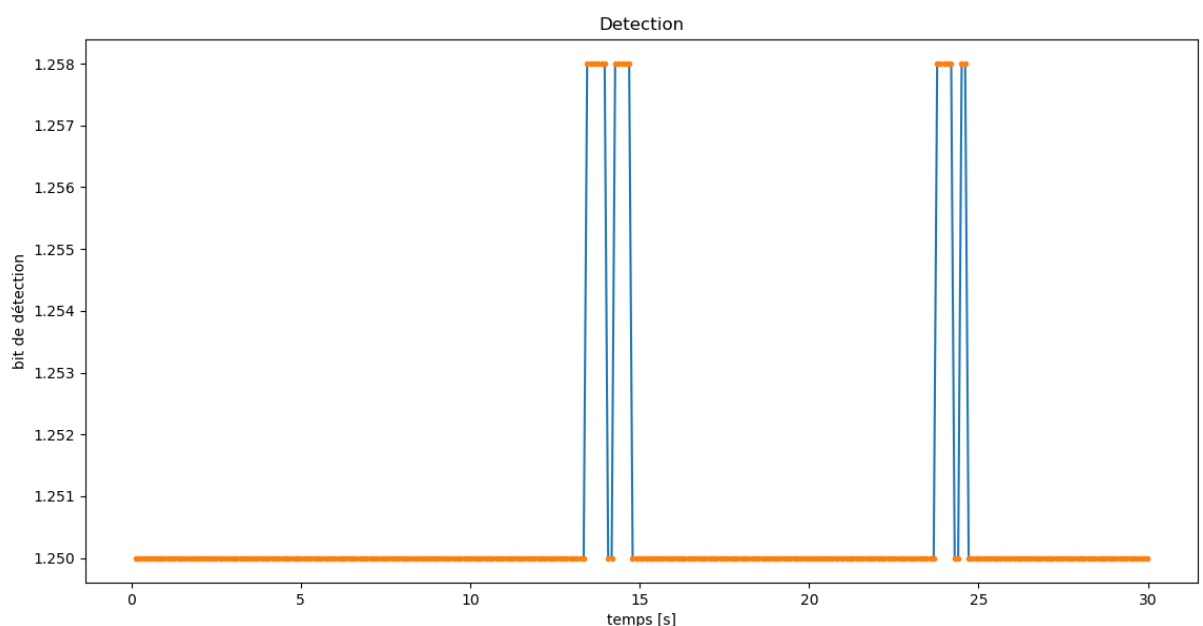


Figure 50 : graphique de détections

Cette technique de détection est adéquate dans le cas où la lisse ne mesure qu'un mètre et que nous sommes dans un cas statique. Une fois que nous sommes sur une lisse de 2m et qu'elle devient mobile, les variations se font moins fortes et donc cette technique s'avère moins efficace. C'est pour cela que nous avons développé un deuxième algorithme de détection qui est expliqué ci-dessous.

Technique finale de détection

Cet algorithme de détection a été développé après des heures d'observations et de tests. Nous sommes arrivés à la conclusion que le moment qui nous intéresse le plus c'est la fermeture, c'est pour cela que nous allons nous concentrer sur celle-ci et négliger tout le reste du signal.

La figure 51 représente le signal filtré lorsque la barrière se ferme et s'ouvre à plusieurs reprises. En dessous de ce signal, nous avons l'état de la barrière qui est représenté par 3 états possibles. Lorsqu'il est 'Low' cela signifie que la barrière est fermée, lorsqu'il est 'High' cela signifie que la barrière est ouverte et lorsqu'il est dans état intermédiaire cela veut dire qu'elle est soit en cours d'ouverture soit de fermeture. La zone sur laquelle nous allons nous concentrer est celle contenue entre les droites bleues.

Dans cette section, nous allons seulement vous expliquer comment fonctionne cette partie de l'algorithme en ce qui concerne les observations cela se fera plus en détail dans la section résultats.

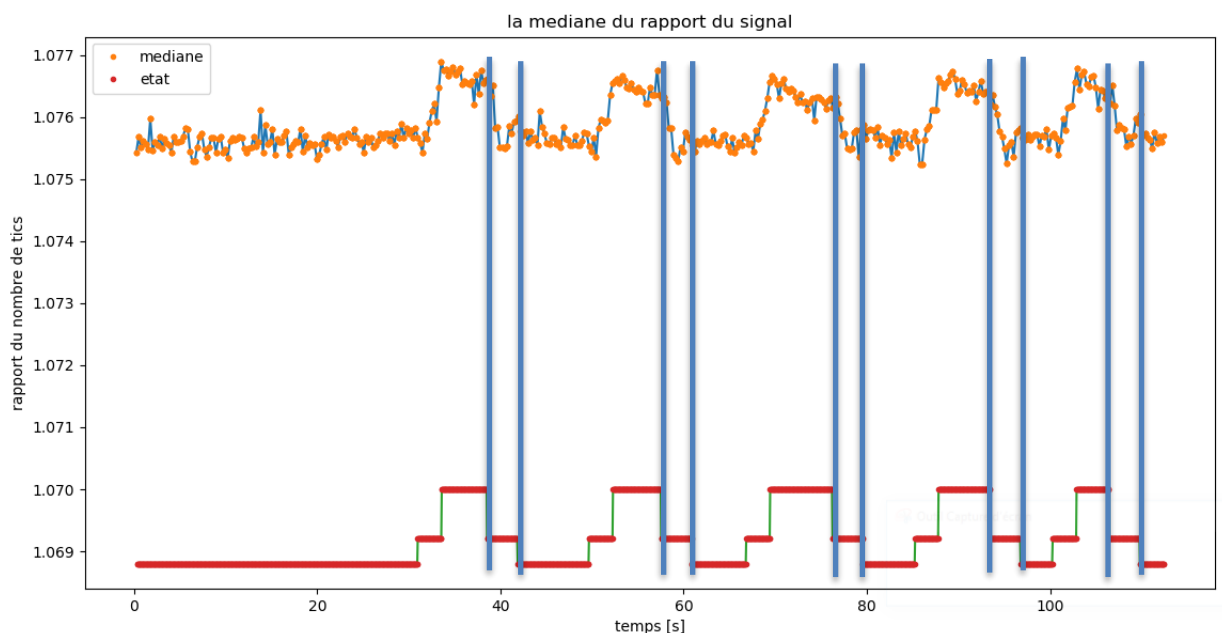


Figure 51 : Signal filtré à l'aide de la médiane & affichage de l'état de la barrière

Le principe est un peu différent de celui observé précédemment. Nous remarquons sur les figures 52 et 53 que lorsqu'il y a détection, le signal filtré a tendance à remonter beaucoup plus fortement que lorsqu'il n'y a pas d'obstacle. Les premiers points sont toujours plus élevés et

les raisons seront explicitées dans la partie résultat. Sachez seulement que le rapport de fréquence est toujours plus élevé aux déclenchements de la fermeture de la barrière.

Sur les figure 52 et 53, on plot à chaque fois le signal filtré lors de la fermeture de la barrière.

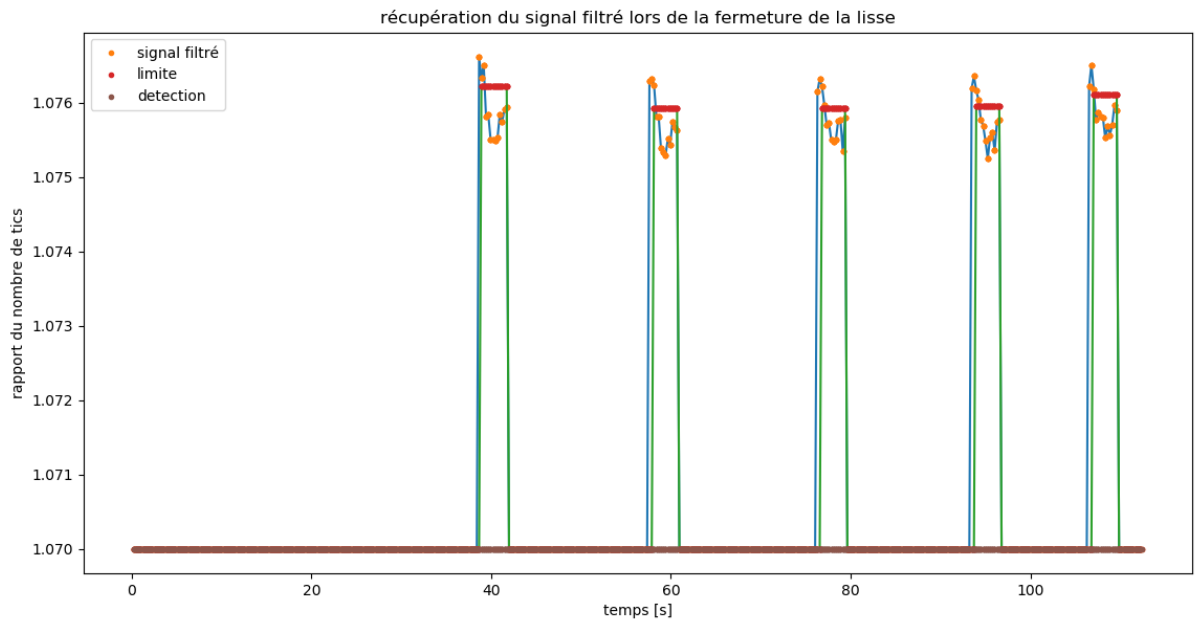


Figure 52 : Signal filtré durant la fermeture sans présence d'obstacle

Comme nous le voyons dans la figure 52, la valeur limite de détection est calculée et elle est indiquée en rouge. Celle-ci est un peu inférieure à la valeur maximum de départ. Il en résulte que tant que nous avons une pente positive au début, nous allons essayer de récupérer la plus grande valeur possible, mais dès qu'il y a un point qui chute alors nous fixerons la limite à 0.9996271 de la valeur maximum. Cette valeur a été obtenue après de multiples essais. Dans le futur elle pourra être calculée automatiquement à partir d'une calibration lors de l'installation du dispositif. Ce que nous envisageons de faire pour cette calibration, c'est de faire 5 fermetures sans obstacle afin que l'algorithme puisse calculer le paramètre. Une fois celui-ci calculer, le capteur sera opérationnel.

Comme nous pouvons le voir en mauve sur la figure 53, si la valeur des signaux filtrés est supérieure à cette limite, cela déclenchera un bit de détection.

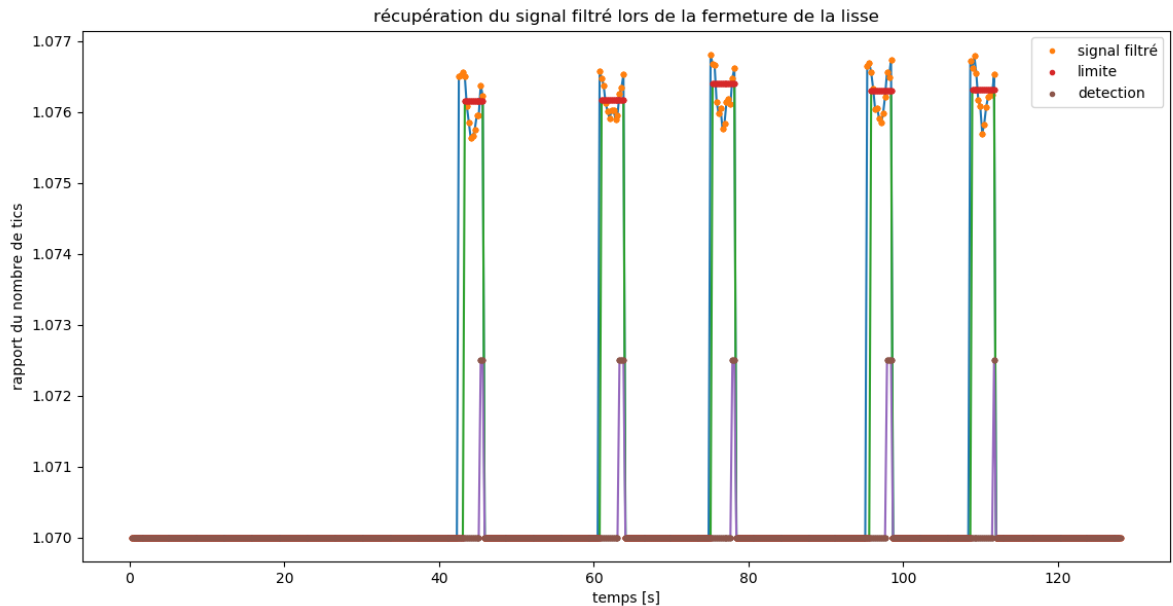


Figure 53 : Signal filtré durant la fermeture avec présence d'un obstacle

6.4. Conclusion

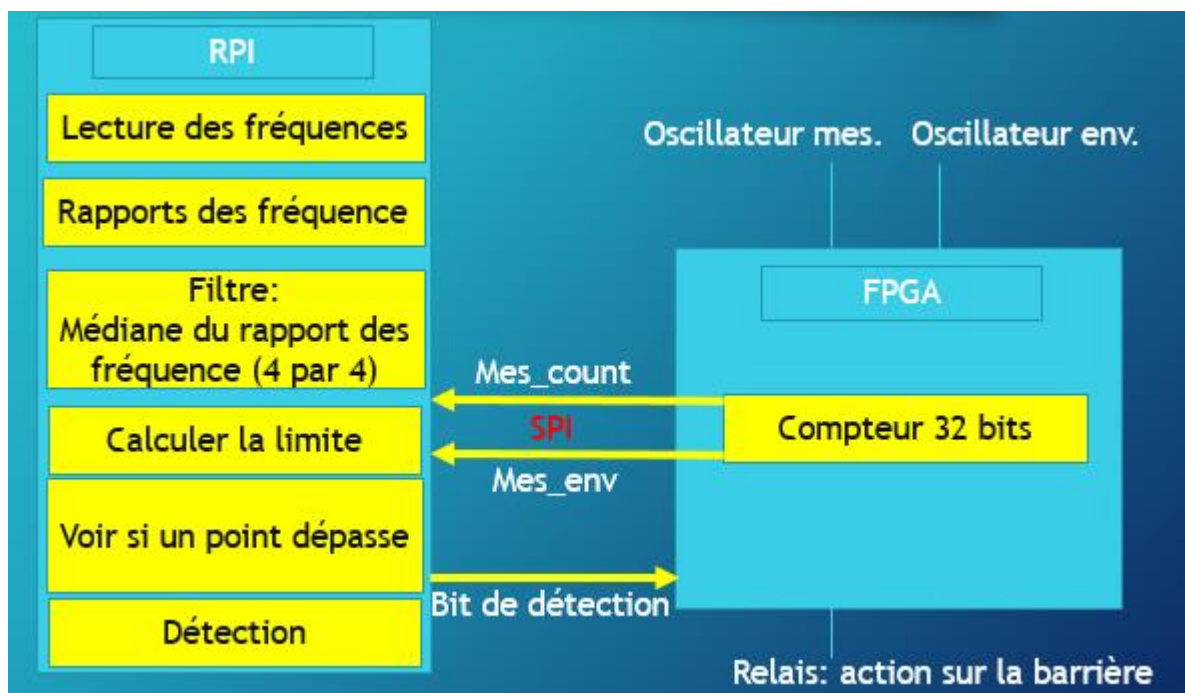


Figure 54 : Architecture du software

Sur la figure 54, nous pouvons voir l'architecture finale qui résume ce qui a été dit précédemment sous forme d'un schéma. Maintenant que nous avons abordé et expliqué les différents algorithmes qui ont été développés, analysons dans la section suivante, les différents résultats obtenus à l'aide de ces algorithmes lors de nos tests.

7. Description des expériences & Résultats

Dans cette section nous allons décrire les différentes expériences que nous avons faites ainsi que les résultats observés. Il y a eu différentes expériences réalisées tout au long de ce travail, ces expériences étaient primordiales pour pouvoir avancer d'une manière constructive. Dans la réalisation de ce travail nous avons travaillé en étape, nous avons tout d'abord confirmé que la technologie pouvait fonctionner à l'aide d'un simple dispositif, puis nous avons réalisé le premier prototype sur une lisse de 1m sur lequel de nombreux tests ont été réalisés et pour finir nous avons créé le prototype final de 2m sur lequel nous avons réalisé des tests et changé ce qui n'allait pas.

7.1. Prototype 0 : Capacité maison

Ce prototype fut créé avec des moyens très primaires afin de pouvoir valider le montage de l'oscillateur Ne555 et le dispositif de détection avec une capacité coplanaire équipé d'une garde qu'on peut voir à la figure 55.

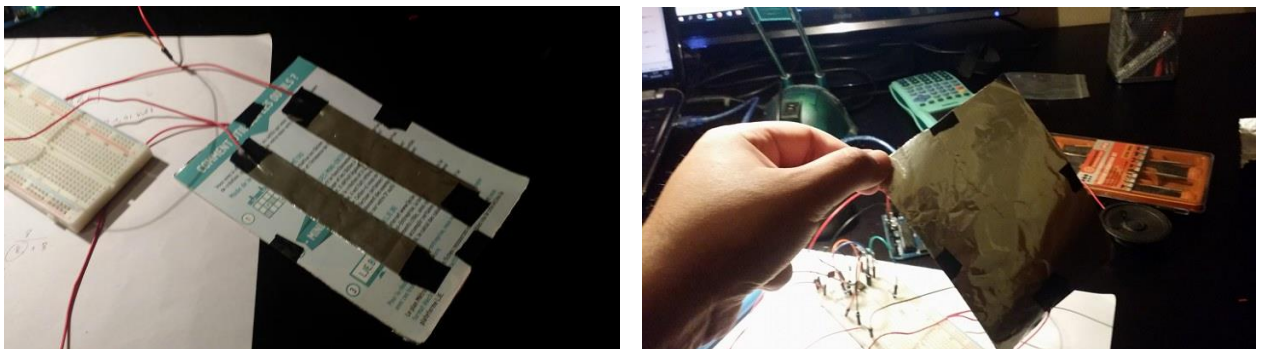


Figure 55 : Prototype 0

Nous avons réalisé cette capacité coplanaire à partir de papier aluminium et de papier journal. Nous avons reproduit une partie de ce qui devait être installé sur la lisse. Pour vérifier que notre dispositif est un minimum fonctionnel et que l'idée de base est réalisable, nous avons réalisé un oscillateur NE555 qui oscillait à 12khz. Lorsque nous rapprochons une bouteille d'eau ou notre main des électrodes, nous entendons une variation de son, ce qui prouvait une variation de fréquences. Donc nous avons réussi à prouver que le principe de base fonctionnait et que nous pouvions passer à l'étape supérieure qui est de designer une mini-lisse.

7.2. Prototype 1 : Mini-capalisse

Une fois que le prototype 0 a montré que cette technologie avait du potentiel pour détecter des corps à fortes compositions d'eau, nous avons décidé de réaliser le premier prototype

monté sur lisse qu'on voit à la figure 56. Ce prototype était assez complexe à réaliser, car collé des tiges d'aluminium un peu tordu sur une lisse est une tâche complexe. Les électrodes ont été collées approximativement comme dans la simulation Comsol. Ce prototype nous a permis de mettre en place une première version de l'algorithme avec lequel nous avons testé la détection de personne dans un état statique, car notre lisse n'était pas encore motorisée. De nombreux tests statiques ont été effectués et ont permis d'améliorer le dispositif ainsi que l'algorithme. Dans un premier temps, nous allons vous montrer les résultats obtenus à l'aide d'un Arduino, puis nous verrons que ce dernier est limité dans le cadre de notre application et nous finirons par les résultats obtenus à l'aide du Raspberry-pi et du FPGA.



Figure 56 : Prototype 1

7.2.1. Détection à l'aide d'un Arduino

Pour réaliser une détection à l'aide de l'Arduino, nous avons utilisé la librairie FreqCount.h, ce qui va nous permettre de calculer le nombre de flancs montant des signaux venant de l'oscillateur. Le fait que l'Arduino Uno n'offre pas une lecture en parallèle des pins est problématique, car les effets de l'environnement apparaissent en même temps sur les deux électrodes, de plus il n'y a que la pin 5 qui fait le compteur haute fréquence. C'est pour cela que nous avons décidé dans le cadre des tests avec l'Arduino de nous contenter de faire nos analyses que sur l'oscillateur NE555 de mesure qui est lié à la capacité de détection.

Dans cette première expérience sur lisse statique, nous allons vérifier si nous arrivons à observer une variation de fréquence importante lors du rapprochement d'une main. Nous allons rapprocher puis éloigner notre main assez rapidement. Observons ce que cela donne !

Résultats avec l'Arduino

Pour commencer, nous avons fixé la fréquence des oscillateurs autour des 150Khz. Cette fréquence a été choisie arbitrairement.

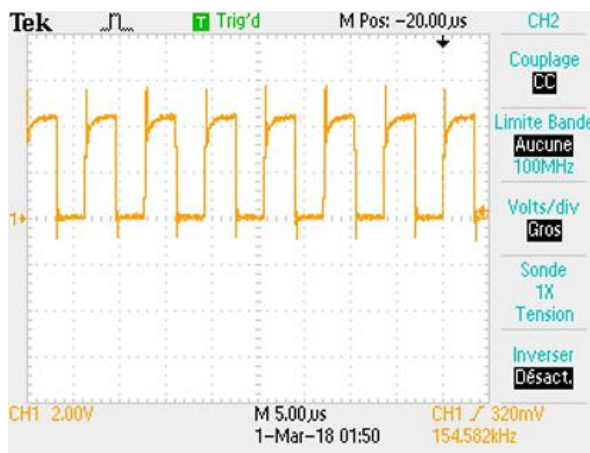


Figure 57 : Signal de sortie de l'oscillateur de mesure non filtrée

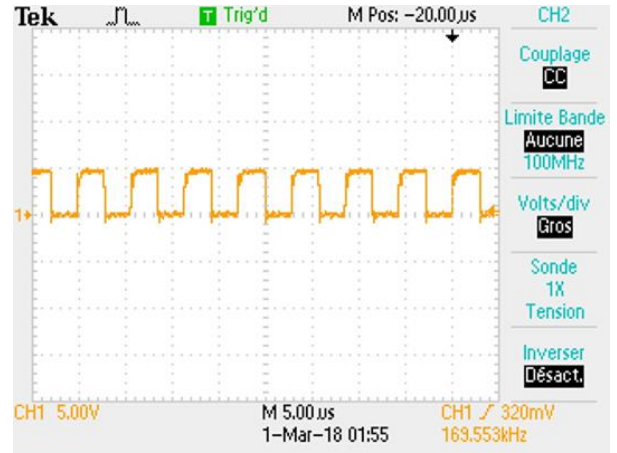


Figure 58 : Signal de sortie de l'oscillateur de mesure filtrés

Comme nous l'observons sur la figure 57, le premier signal que nous obtenions contenait des glitches ce qui n'est pas top pour l'électronique. C'est pour cela que nous avons décidé de mettre des capacités de découplage sur l'alimentation des NE555 afin de faire disparaître ces glitches et de pouvoir l'envoyer dans l'Arduino.

Une fois le code mis sur l'Arduino nous pouvons enfin afficher les résultats en temps réel sur la console du logiciel Arduino. Nous avons donc décidé d'observer l'effet d'une main pour différentes distances. Nous rapprochions la main des électrodes et nous la retirons sur un laps de temps assez faible.

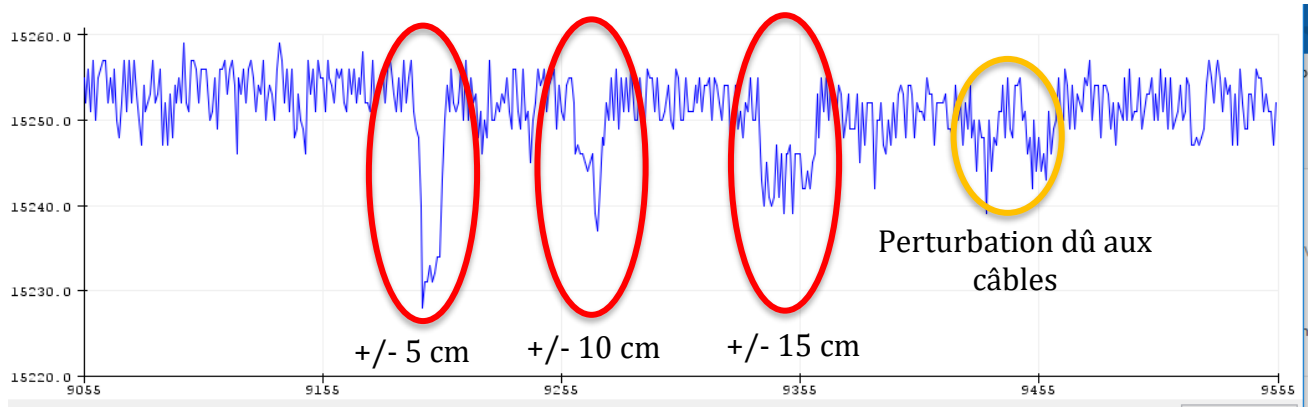


Figure 59 : Mesure de fréquence sur un intervalle de 100ms

On observe clairement sur la figure 59 que le fait de rapprocher une main fait varier la fréquence. Par exemple dans le premier cas, nous avons une variation de 250Hz, ce qui représente une variation de 0.16% par rapport à la fréquence non perturbée. En termes de pourcentage cela a l'air assez faible, mais quant à la variation elle est assez visible pour qu'un bon algorithme puisse la détecter. Cette visibilité s'explique par le fait que les variations de bruit ne sont pas aussi élevées que les variations dues à la main. Nous estimons que dans le cas où la main se trouve à 5cm, le facteur $\frac{Var.Signal}{Var.Bruit}$ est égale à 5. Ce facteur est assez élevé et nous confirme qu'il y a une possibilité de détection non négligeable.

Nous observons aussi sur la figure 59, qu'il y a une variation lorsque nous faisons passer la main au-dessus des câbles qui relient les électrodes à l'oscillateur. Ce sont des capacités parasites, qui émettent aussi un champ électrique tout comme les électrodes. Dès que nous nous rapprochons ou que nous faisons bouger les câbles, nous faisons varier les champs électriques ce qui fait varier à son tour la capacité et donc la fréquence. La solution est d'utiliser des câbles coaxiaux comme fait par la suite et représenté à la figure 60. Ces câbles ont une armature qui est mise à la masse et qui agit comme une cage de faraday, ce qui empêche d'avoir des fuites des lignes de champs.



Figure 60 : lisse avec le câble coaxial

Cette technique avec câble coaxial sera abandonnée plus tard pour des raisons d'encombrement. La solution finale qui a été mise en place consiste à mettre le circuit oscillateur le plus proche possible des électrodes afin de minimiser la longueur des câbles et donc des capacités parasites. Actuellement, le circuit oscillateur est installé dans la lisse à moins de 5 cm des électrodes.

7.2.2. Détection à l'aide d'un FPGA & d'un RPI

Comme l'Arduino n'est pas très adapté à notre application, nous avons choisi de combiner un Raspberry-Pi avec un FPGA. Le FPGA permet de récupérer les signaux en temps réel et simultanément. Ici, nous pourrions donc récupérer les fréquences issues de nos deux oscillateurs (environnement, mesure) et faire le rapport de ces deux signaux ainsi nous supprimerons les effets environnementaux. Nous sommes toujours dans un cas statique.

Résultats avec le FPGA + RPI

Nous avons fait plusieurs tests avec ce dispositif, car c'est celui-ci qui a permis de développer une grosse partie de l'algorithme. Nous allons aborder seulement le test le plus intéressant qui nous a permis de confirmer qu'il était possible de détecter une tête humaine.

Nous avons imaginé un moyen très intéressant de simuler le passage d'une tête sous la lisse. Nous avons fixé la lisse à un mètre du sol et nous avons demandé à une personne sportive de réaliser quelques abdominaux sous la lisse (voir figure 61). Cela s'est fait à l'intérieur dans des conditions climatiques normales d'une chambre. À l'aide d'une latte, nous avons mesuré la plus petite distance qui séparerait la tête et la lisse lorsque la personne faisait ses abdominaux. Cette distance était d'environ 15 cm, c'est ce que nous pouvons voir sur la figure 61.



Figure 61 : Passage sous la lisse à environ 15 cm

Nous avons demandé à la personne de faire deux abdominaux ce qui nous donne les résultats qu'on peut observer sur la figure 62.

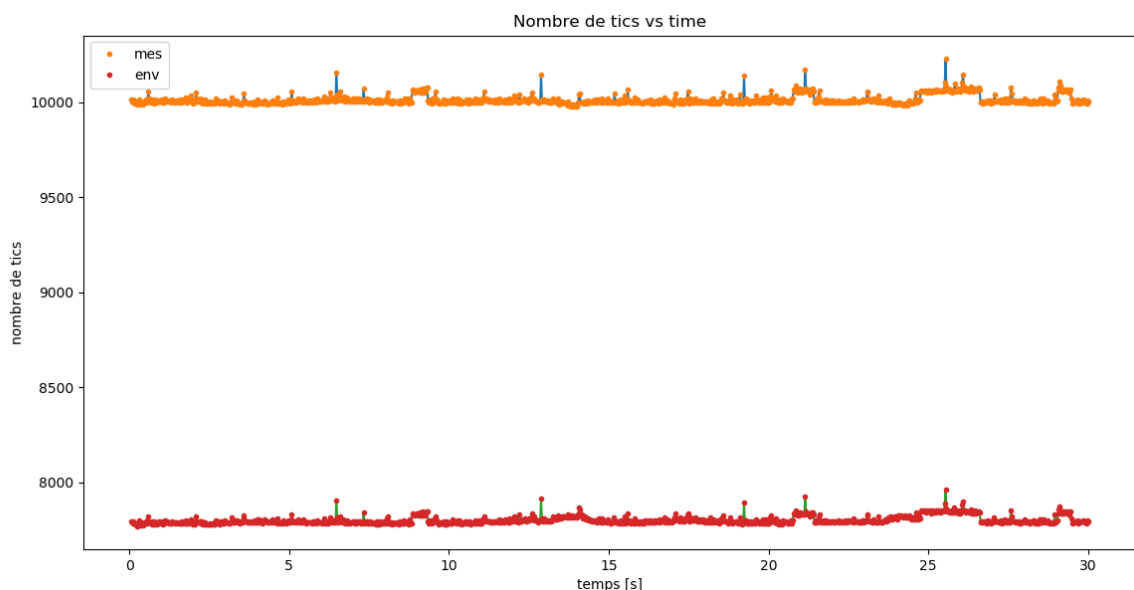


Figure 62 : Nombre de tics

Dans un premier temps, nous allons récupérer le nombre de tics calculer à l'aide du FPGA qu'on peut voir sur la figure 62 en orange et en rouge.

Nous remarquerons que ces deux signaux se ressemblent fort. Il y'a deux explications à cela, la première c'est que si l'horloge interne du Raspberry Pi ralentit alors cela impactera les compteurs. Prenons l'exemple où le Raspberry Pi ralenti de 0.1ms. Dans ce cas, le compteur comptera 0.1ms en plus, et cela impliquera une petite augmentation du nombre de tics calculée. Puis la deuxième explication c'est l'environnement, comme un petit courant d'air ou encore des ondes perturbatrices.

En faisant le rapport des deux signaux ci-dessus, nous arrivons à voir sur la figure 63 qu'on arrive à supprimer la majorité des perturbations. Mais il reste tout de même quelques petits éléments perturbateurs qui s'en iront juste après le filtre qu'on peut voir à la figure 64.

Néanmoins, on peut observer dans les cercles verts à la figure 64 que le signal n'est malheureusement pas très stable, nous n'avons pas trouvé d'explication concrète à ce phénomène qui apparaît de temps en temps. Mais cela ne pose pas de problème dans le cadre de notre détection, car nous arrivons à distinguer l'un de l'autre par le fait que les chutes dues au bruit (entouré en vert) sont caractérisées par une chute dont la pente est faible contrairement à la vraie détection où le pic est très prononcé.

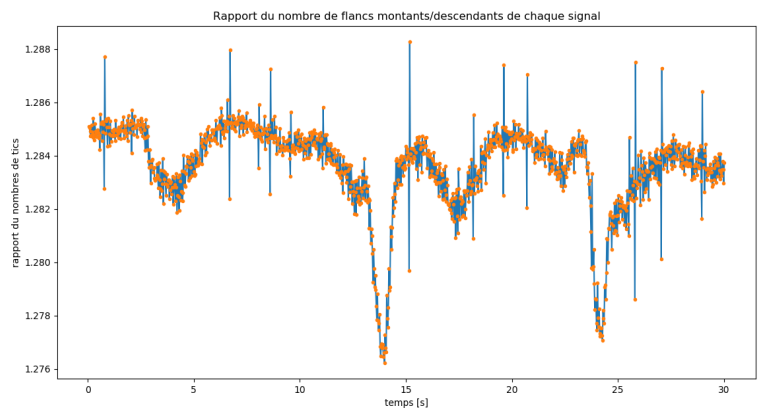


Figure 63 : Rapport des fréquences

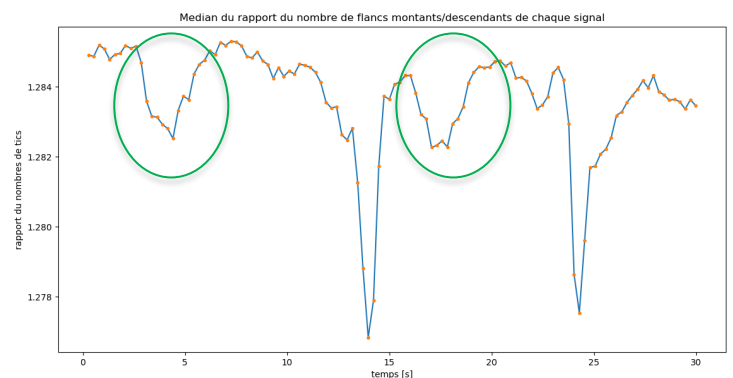


Figure 64 : Filtrage (médiane du rapport des fréquences)

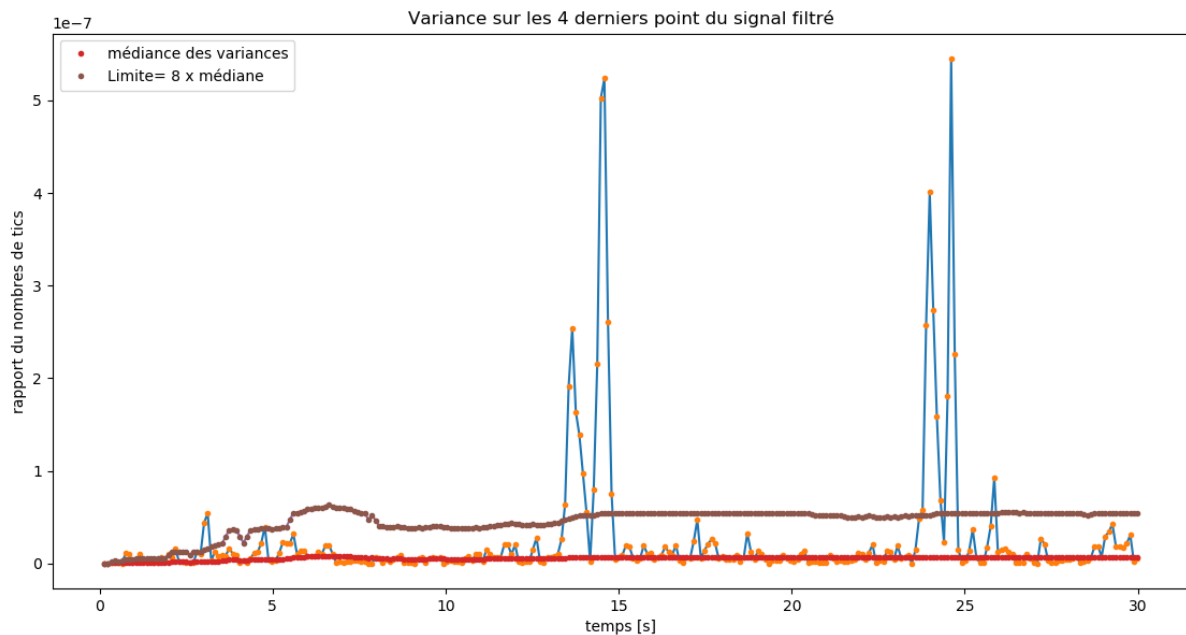


Figure 65 : Variance glissante sur le signal filtré

Nous observons sur la figure 65 que la variance au moment où il y a détection est nettement plus supérieure à 10 fois la valeur médiane de la variance. Nous observons aussi que durant les sept premières secondes il y a des variations importantes qui sont dues au démarrage du software sur le Raspberry Pi. Et nous remarquons aussi que le temps pour que la médiane de la variance se stabilise est d'environ 8 secondes. C'est pour cela que nous avons pris la décision radicale de laisser 10 secondes au dispositif pour pouvoir nous initialiser.

Nous remarquerons aussi que dans ce graphe nous avons un petit effet yoyo qui fait en sorte qu'il y ait une double détection pour un exercice abdominal. Cela s'explique par le fait que lorsqu'on fait des exercices abdominaux en dessous de la lisse on a tendance à aller plus à l'avant et puis à revenir ce qui fait belle est bien deux passage et donc deux détections. C'est ce que nous pouvons voir sur la figure 66.

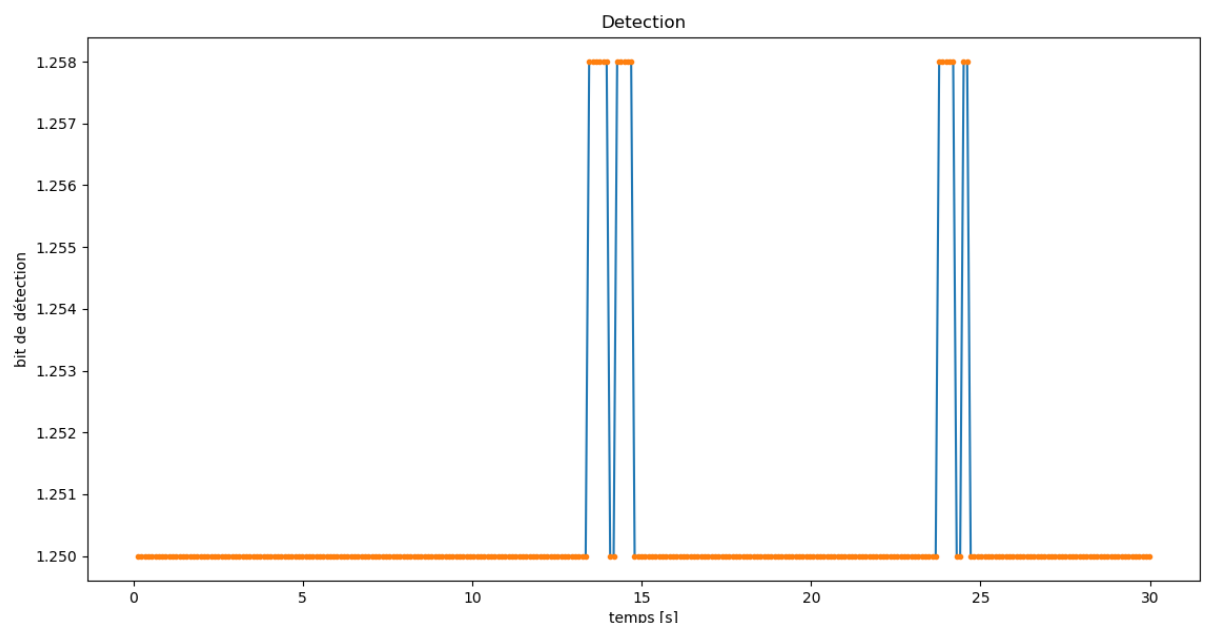


Figure 66 : bits de détection

Jusqu'ici, nous avons vu qu'en mode statique avec une lisse de 1 mètre la détection est faisable. Mais notre lisse finale n'est pas statique et elle ne mesurera jamais un mètre. A présent il est important de tester nos technologies sur une lisse plus grande en mouvement.

7.3. Prototype 2 : capalisse finale

Avec ce prototype, nous allons faire les tests finaux avec une lisse en mouvement et pour représenter la tête nous allons utiliser une pastèque. La pastèque est un fruit qui représente assez bien la tête en termes de taille, de quantité d'eau et de dureté.

En ce qui concerne l'algorithme nous n'allons pas utiliser l'algorithme avec la variance comme montrée ci-dessus, mais nous allons plutôt utiliser l'algorithme final présenté dans la partie software, nous expliquerons pourquoi dans cette partie.

Comme nous voulons mesurer la distance de détection assez précisément, nous allons placer la pastèque sous la lisse, et la fermer puis mesurer la distance qui sépare la lisse et le haut de la pastèque. Une fois cette mesure obtenue, nous pouvons commencer nos tests. La pastèque quant à elle est placée sur une chaise qui peut monter et descendre. Le tout est installé sur un transpalette qui va surélever la chaise. On peut voir à la figure 67 comment se présente le dispositif de test.

L'avantage de mettre cela comme ça permet de ne pas créer de collision entre la pastèque et la lisse et permet de mieux observer qu'elles sont les différences quand il n'y a rien.

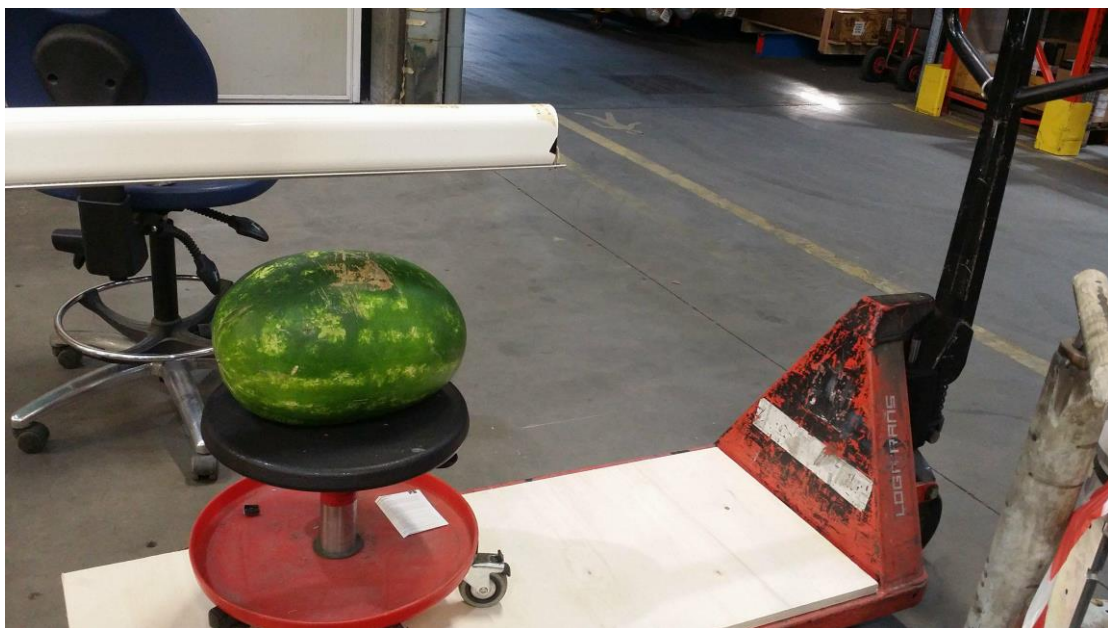


Figure 67 : représentation du dispositif pour faire les tests

Signaux de détection : rien VS pastèque

Dans cette section nous allons voir qu'elle est la différence des signaux entre lorsqu'il n'y a rien et lorsqu'il y'a une pastèque à une certaine distance (1cm dans ce cas-ci)

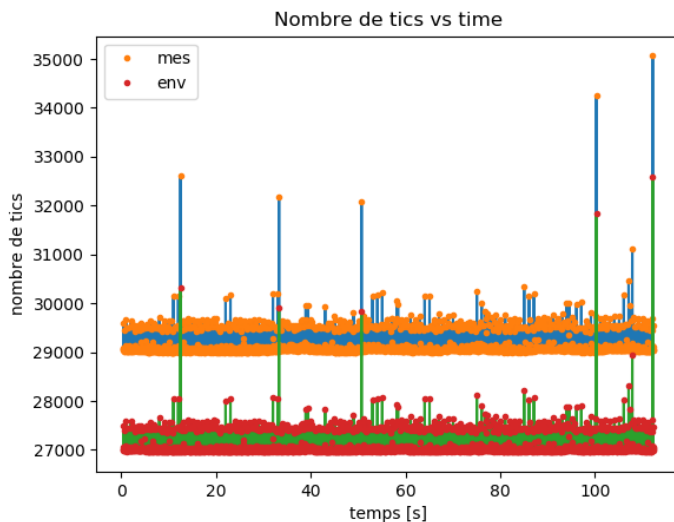


Figure 68 : Nombre de tic en présence de rien

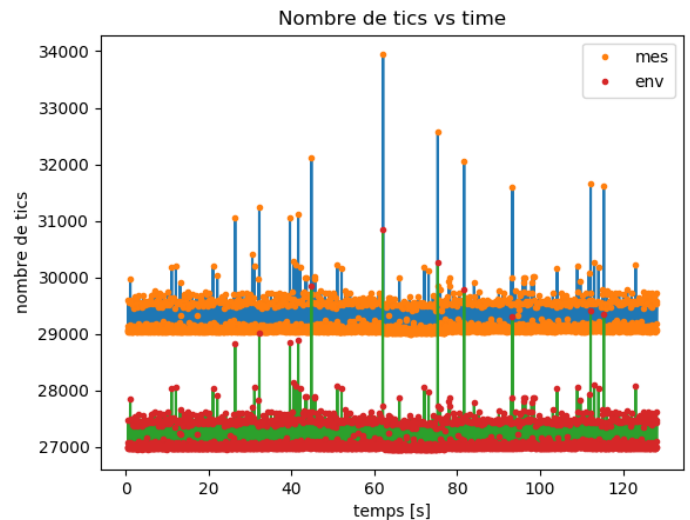


Figure 69 : Nombre de tics en présence d'une pastèque

Nous pouvons observer sur les figures 68 et 69 que les signaux sont très bruités par rapport aux autres sections. Cela s'explique par le fait que maintenant tout tourne sur le Raspberry Pi et il faut savoir qu'à chaque fois que notre algorithme rentre dans un 'if' cela prend un peu plus de temps. Auparavant, nous n'avions pas ce problème, car nous faisons seulement de la récupération de data et le traitement était effectué sur le pc.

Mais ce bruit ne nous pose aucun problème, car le rapport de signal restera le même, un retard de 0.1ms se répercutera sur les deux mesures.

Sur les figures 70 et 71 nous pouvons voir le rapport de fréquence (ou nombre de tics) des 2 signaux (mesure/environnement) ainsi que l'état de la barrière. (Rappel : High = ouvert, Low = fermé, Middle = en cours d'ouverture ou de fermeture)

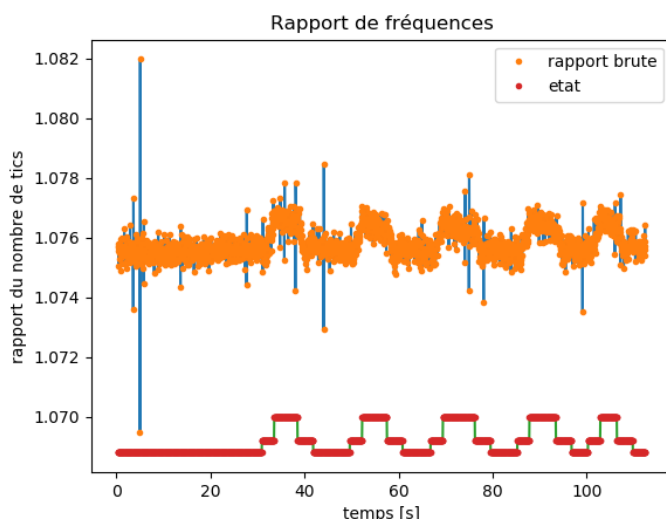


Figure 70 : Rapport brut sans présence de pastèque + état de la barrière

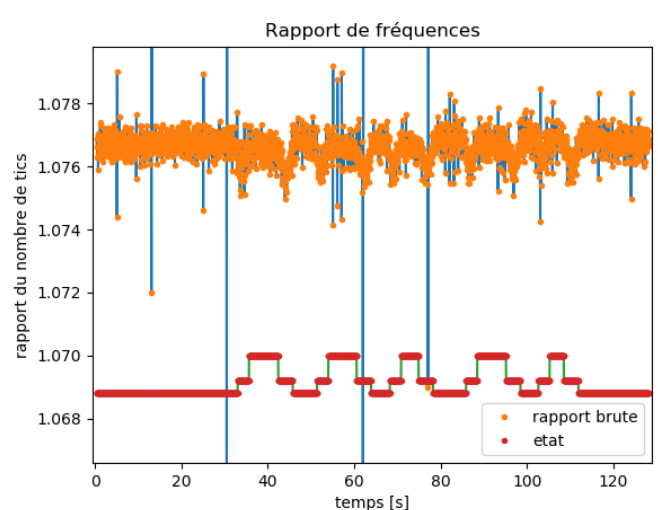


Figure 71 : Rapport brut avec présence de pastèque + état de la barrière

La question qui se pose légitimement, c'est si les variations sont corrélées linéairement ? Car pour l'instant l'hypothèse que nous faisons depuis le début, c'est de dire que s'il y'a des perturbations de l'environnement alors le rapport de fréquence restera le même. Pour vérifier cela, nous avons décidé de faire ploter le nombre de tics compter pour l'oscillateur d'environnement vs le nombre de tics compter pour l'oscillateur de mesure. Nous pouvons voir sur les figures 72 et 73 les résultats obtenus.

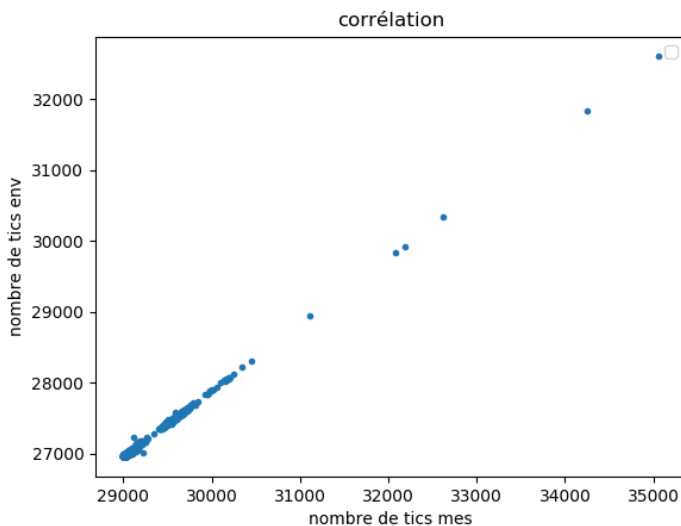


Figure 72 : Corrélation (tics env. vs tics mes.) en l'absence de pastèque

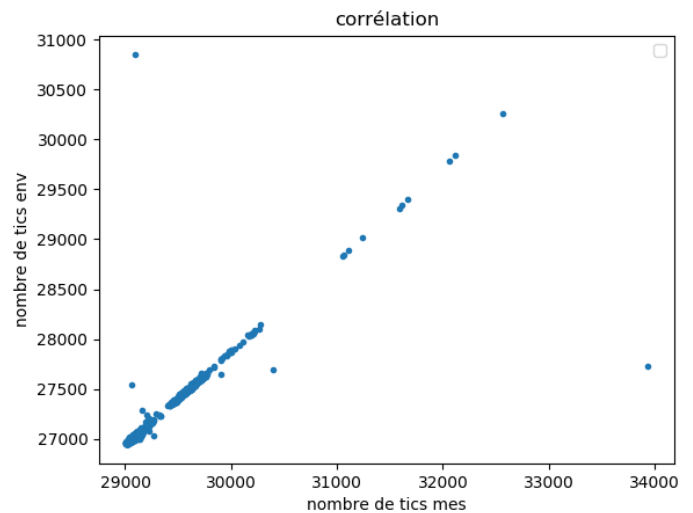


Figure 73 : Corrélation (tics env. vs tics mes.) en présence de pastèque

Nous observons bien une corrélation linéaire ce qui confirme que le rapport des signaux reste constant. Mais nous observons sur la figure 73 qu'il y'a plus de outliers. Ce qui s'explique par le fait qu'il y'a un phénomène de bruit qui apparaît de temps en temps et qui crée un petit saut de fréquence comme nous pouvons le voir dans la figure 71 du rapport brut.

Maintenant à l'aide du filtre nous allons lisser/filtrer notre signal afin d'avoir une vision plus claire de ce qui se passe.

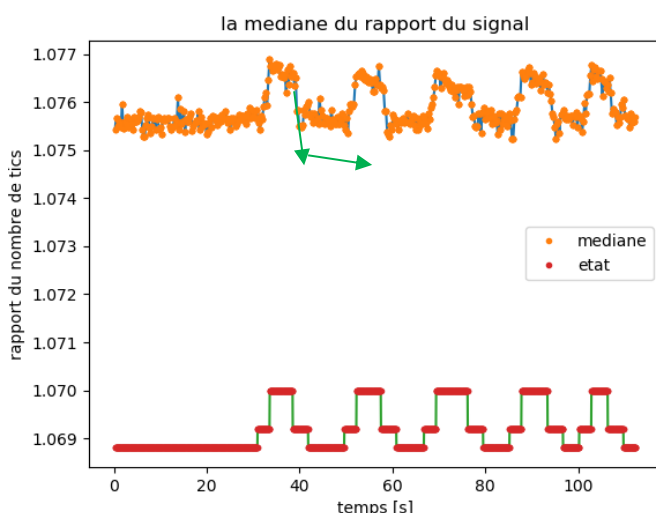


Figure 74 : Rapport filtré + état de la barrière en absence de pastèque

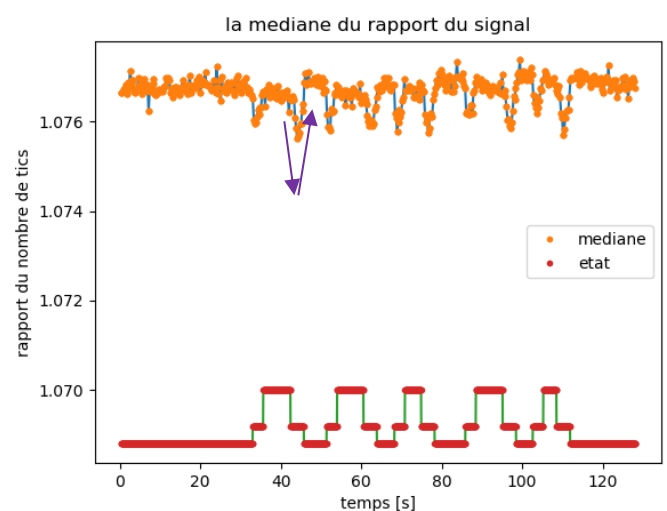


Figure 75 : Rapport filtré + état de la barrière en présence de pastèque

Nous avons du mal à voir quelque chose d'intéressant comme ça, mais si nous regardons les choses de plus près, nous voyons des choses intéressantes. Concentrons-nous sur la fermeture, nous observons que lorsqu'on ferme la barrière dans le cas où nous n'avons rien, comme nous pouvons le voir sur la figure 74 à l'aide des flèches vertes, la fréquence aura tendance à descendre et à rester dans un état plus faible que lorsqu'elle est ouverte. Mais si nous prenons le cas où il y'a la présence de la pastèque, nous observons sur la figure 75 à l'aide des flèches mauves, que la fréquence a tendance à descendre et puis dès qu'elle commence à détecter la pastèque celle-ci remonte très rapidement.

Nous observons cela plus clairement lorsqu'on récupère seulement le signal à l'endroit qui nous intéresse. On voit sur les figures 76 et 77 que lorsqu'il y'a une détection, les points se retrouvent facilement au-dessus de la limite calculer par l'algorithme.

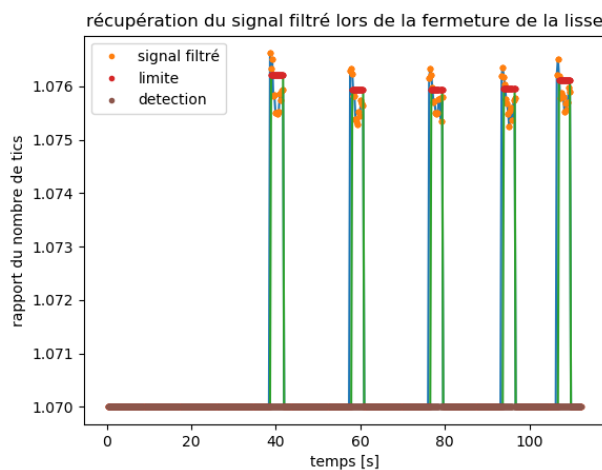


Figure 76 : Rapport des fréquences lors de la fermeture en absence de pastèque

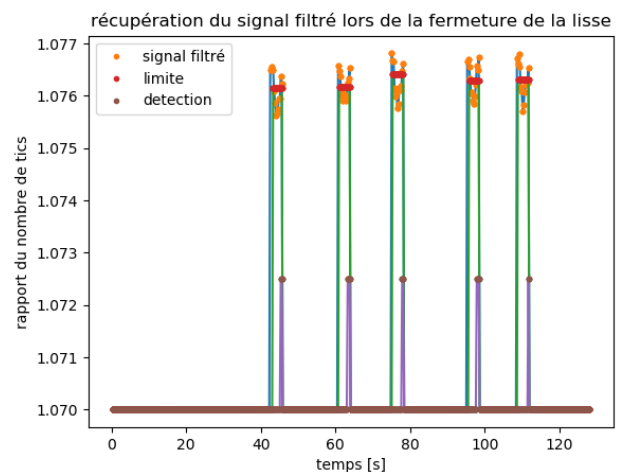


Figure 77 : Rapport des fréquences lors de la fermeture en présence de pastèque

Comme dit dans la section software, nous calculons la limite à partir de la valeur maximum de démarrage. Cette valeur maximum vient du fait que lorsque la lisse est ouverte le rapport de fréquence est toujours plus élevé que lorsqu'elle est fermée. La seule explication que nous avons trouvée c'est que lorsque la lisse se rapproche du sol, cela crée une capacité parasite qui influence nos mesures faiblement. Cela étant dit, cela ne pose pas de problème à notre algorithme ; au contraire, il s'en sert pour calculer cette limite qui va varier en fonction de cette valeur maximum.

Signaux de détection en fonction de la distance

Dans cette section, nous avons fait plusieurs tests pour voir jusqu'à quelle distance nous pouvons détecter. Nous avons pris le pire cas qui est le cas où la personne se trouve à 1m90 du boîtier, car à cette distance la vitesse de la lisse est maximum et donc la détection doit se faire vite. Le temps de descente de la lisse a été programmé à 4 sec.

Les figures suivantes montrent les résultats de détection pour différentes distances : 1,3,8,13 et 18 cm

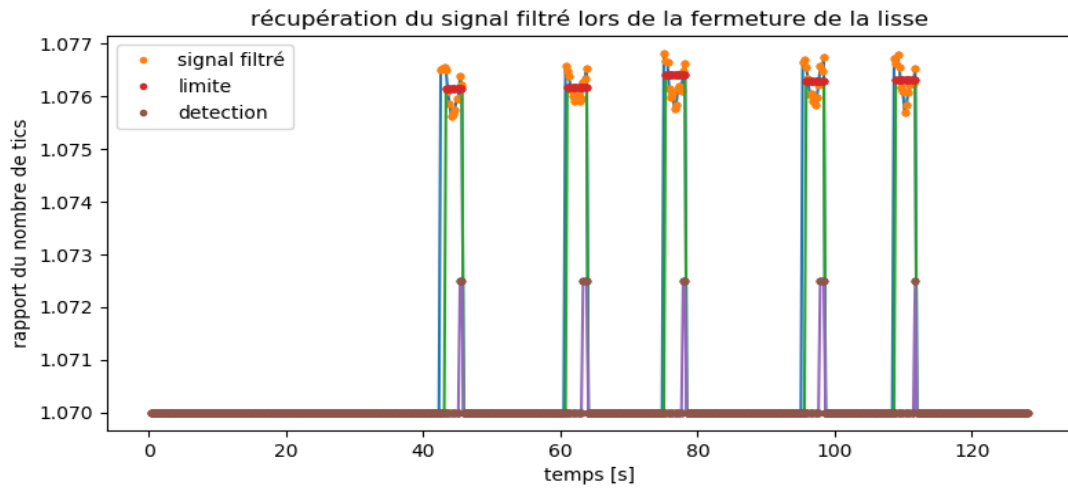


Figure 78 : Détection de la pastèque à 1cm

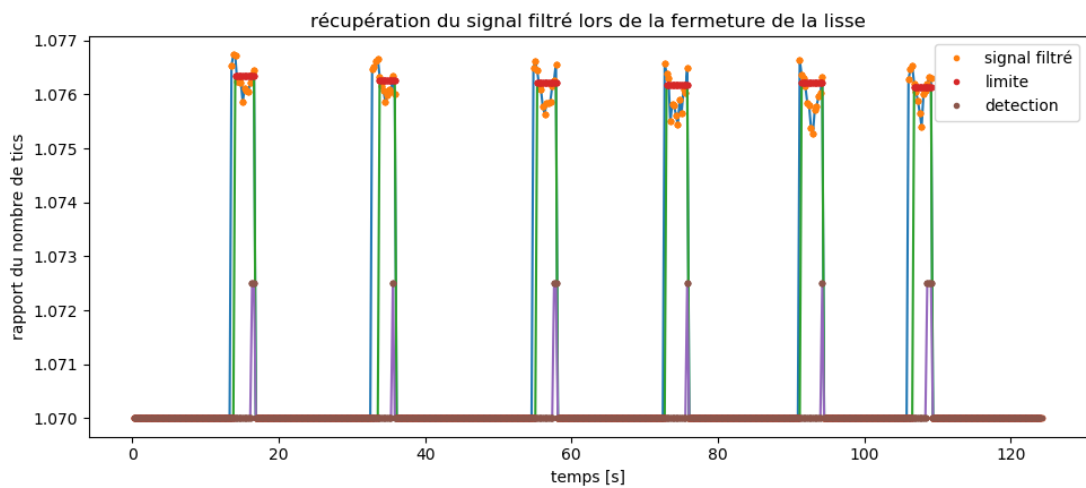


Figure 79 : Détection de la pastèque à 3cm

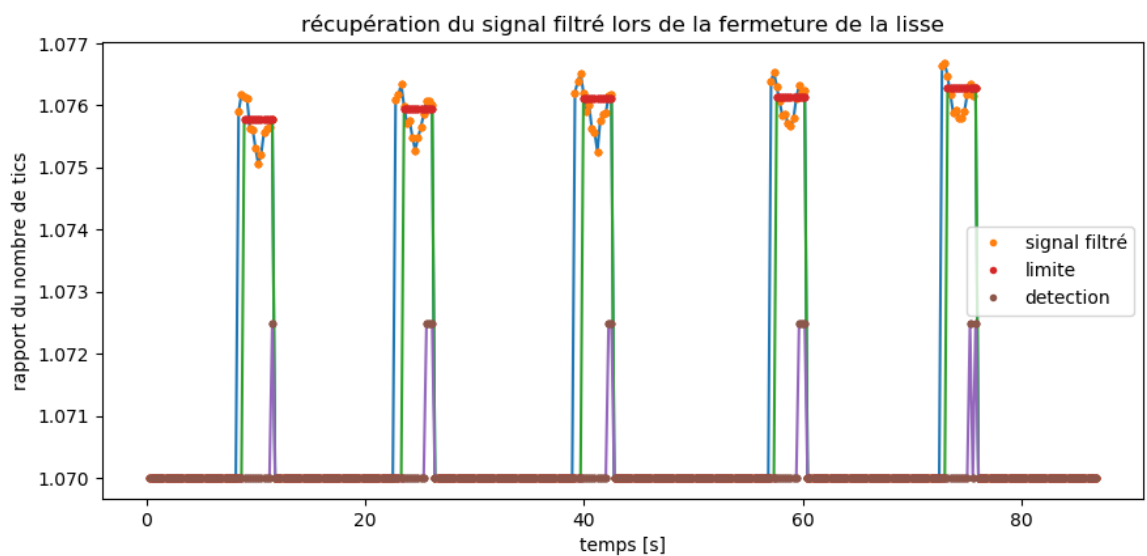


Figure 80 : Détection de la pastèque à 8cm

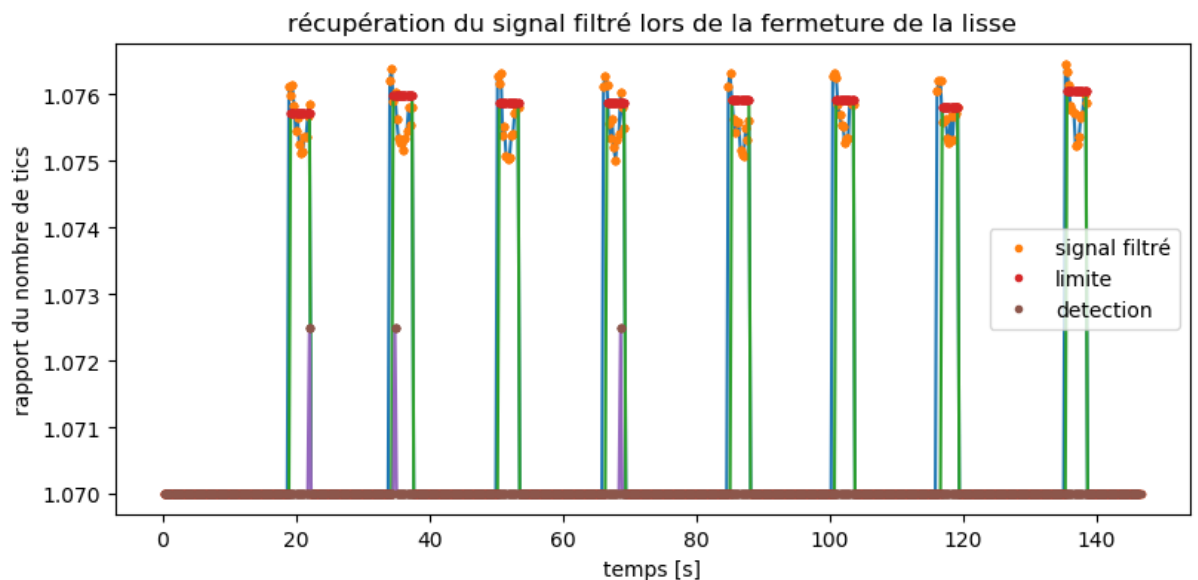


Figure 81 : Détection de la pastèque à 13cm

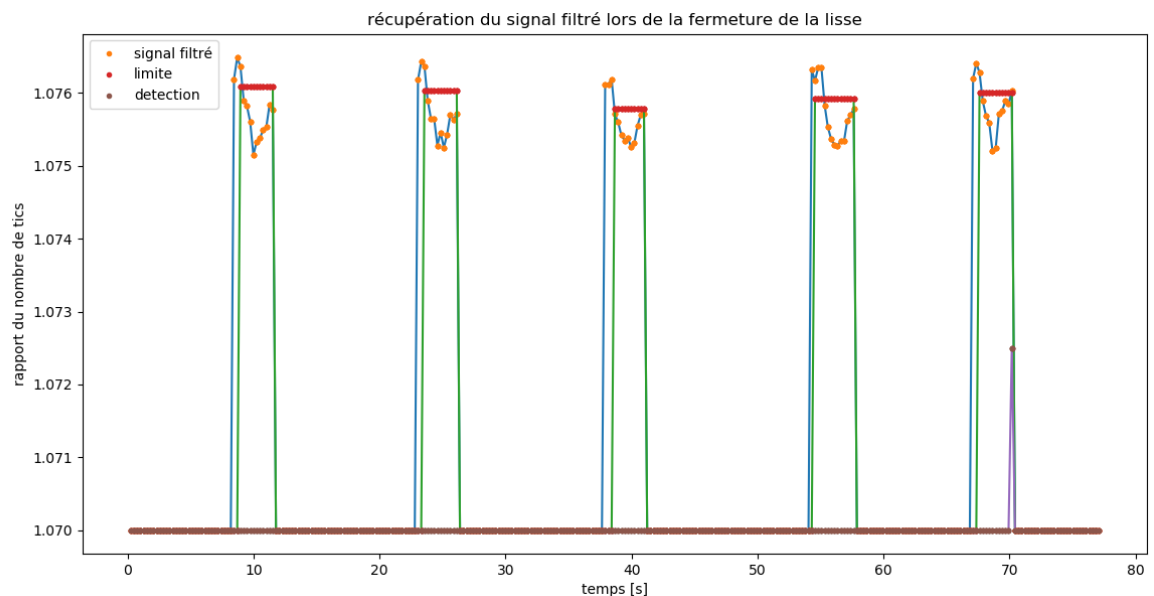


Figure 82 : Détection de la pastèque à 18cm

Comme nous pouvons l'observer sur la figure 81, la détection commence à avoir un peu de mal à 13 cm. Comme nous le constatons, la remonter à la fin du signal est belle est bien présente, mais elle n'est pas assez importante pour passer le seuil. Comme nous pouvons le voir sur les figures 81 et 82, nous sommes à la limite à chaque fois malheureusement. Ce que nous pouvons faire c'est diminuer le seuil, mais à ce moment-là, nous risquons d'augmenter le nombre de faux positifs. Il faut bien évidemment trouver la meilleure valeur de seuil pour que nous puissions avoir un taux de détection acceptable, mais cela reste mon travail pour les jours qu'il me reste. En cas de bonnes valeurs, celles-ci seront partagées avec vous durant la présentation.

Nous allons très probablement diminuer ce seuil afin que la barrière soit plus sensible quitte à avoir quelques faux positifs. Ce que nous voulons dire par là c'est qu'on préfère que la barrière remonte quand il n'y a rien au lieu d'assommer quelqu'un quand une personne s'y trouve.

Pour conclure, cette technique pourra détecter des personnes sous la lisse à condition que la barrière soit lente, si nous voulons augmenter la vitesse nous devons alors pouvoir détecter à l'avance.

Erreur de détection

Cette petite section va discuter des faux positifs qui sont un petit problème qui n'a pas encore été résolu malheureusement. Ils posent un petit problème de détection, ceux-ci font croire aux systèmes qu'il y a une détection alors qu'il n'en est rien.

Voici sur la figure 83, un exemple extrême qui illustre ce problème.

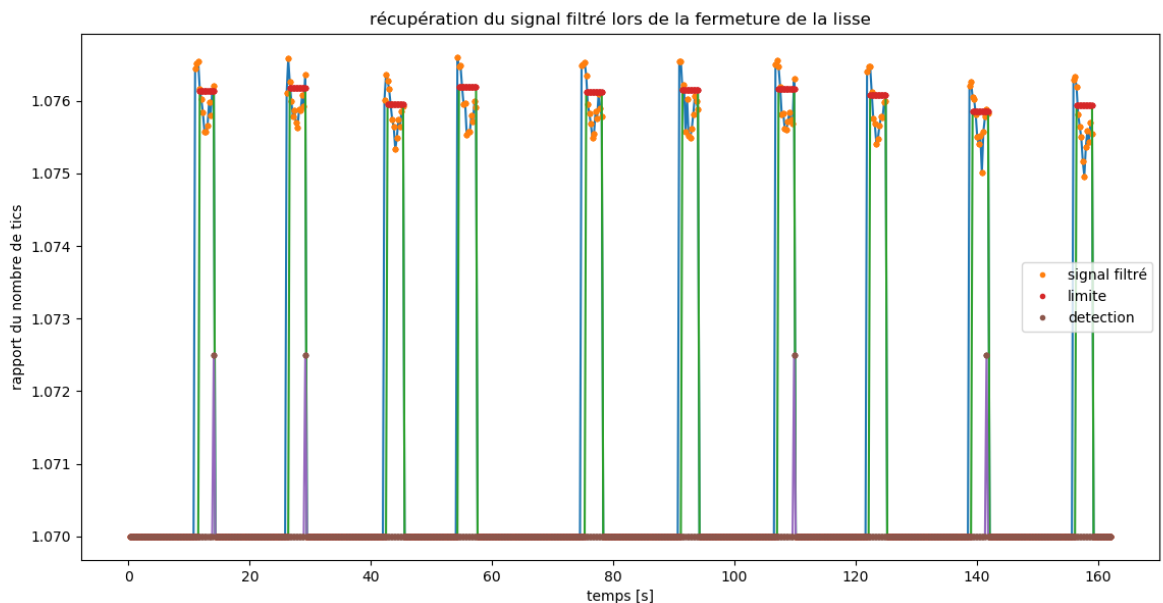


Figure 83 : Faux positif

Nous avons fait une série d'ouverture et de fermeture à répétition sans qu'il n'y ait aucun obstacle sous la lisse. Nous remarquons que sur 10 fermetures 4 disent qu'ils observent quelques choses. C'est beaucoup, mais le problème pourrait être résolu par le fait de négliger le dernier tic dans toutes les fermetures. On remarque que celle-ci intervient à chaque fois à la fin. Normalement s'il doit y avoir une détection d'une personne qui mesure plus de 90cm celui-ci devrait faire réagir le système bien avant la fermeture totale. C'est une piste qui sera étudiée si le temps nous le permet.

Test avec l'épaule

Après les tests avec la pastèque, nous avons décidé de passer au test pour voir si nous pouvions détecter un humain. Nous avons décidé de nous mettre assis sous la lisse, notre épaule se trouvait à 1 mètre 30 du sol et notre chaise se trouvait à 1m 80 de l'axe de rotation de la lisse. Nous avons fermé à plusieurs reprises la barrière sur notre épaule, mais nous ne sommes pas capables de vous fournir une mesure de la distance exacte qui sépare notre épaule et la lisse

mais sachez que celle-ci est comprise approximativement entre 5 et 10 cm. Ce test va nous permettre de voir la réaction de l'algorithme lorsqu'un individu se trouve sur la trajectoire de fermeture.

Comme nous pouvons le voir sur la figure 84, la détection a bien lieu.

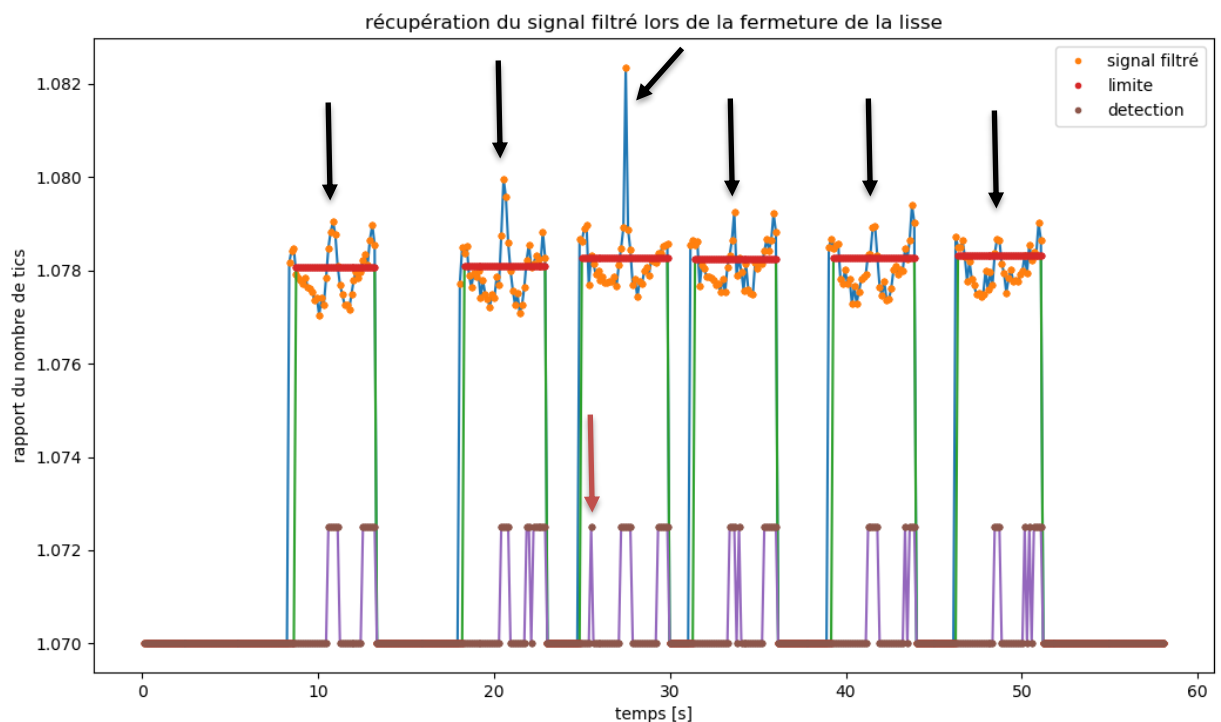


Figure 84 : détection de personnes sous la lisse

Nous pouvons voir plusieurs choses dans la figure 84. La première chose que nous voyons est que ce test confirme bien l'hypothèse de départ : lorsque la lisse se rapproche d'un corps composé à majorité d'eau, le rapport de fréquences dépasse très vite le seuil calculé par le programme. Nous remarquons une seconde chose, qui est que la détection se passe toujours au milieu, cela est totalement normale, car le temps de montée est égal au temps de descente dans notre cas.

Nous observons aussi une chose anormale qui est la détection systématique à la fin de la remontée de la lisse. Cette anomalie peut s'expliquer par le fait que la lisse retourne à la position ouverte et comme nous le savons cette position à un rapport de fréquence plus important et donc il y'a un retour vers l'état de départ. Cela crée une certaine symétrie comme nous pouvons le voir à la figure 84. Cette anomalie n'est pas un réel problème car une fois que la détection est effectuée la lisse retourne à sa position ouverte et permet d'éviter tout incident.

Pour terminer notre analyse, nous voyons que les faux positifs apparaissent encore de temps en temps comme nous pouvons l'observer sous la flèche rouge de la figure 84. On observe qu'il

est à la limite du seuil, ce qu'on devrait faire dans le futur, ce serait d'augmenter faiblement le seuil pour éviter ces faux positifs.

Ce test est confiant et nous montre qu'il y'a une réelle possibilité de détections de personnes sous la lisse.

Humidification des électrodes et détection d'une personne

Après le test de la détection humaine avec des électrodes sèches, nous avons décidé de réitérer l'expérience, mais avec des électrodes humides.

Comme nous n'avons pas encore la possibilité de mettre notre barrière à l'extérieur, nous avons décidé de mettre de l'eau sur nos électrodes à l'aide d'un chiffon saturé d'eau. Nous pouvons voir la présence de gouttelette d'eau sur la figures 85.

Cette expérience va nous permettre de montrer que le dispositif néglige les effets d'humidité grâce à la capacité d'environnement.

Tout comme la sous-section précédente, nous avons réalisé de multiples fermetures sur notre épaule qui se trouvait à 1m30 du sol. Nous ne serons pas capables de vous fournir une mesure exacte de la distance minimale séparant notre épaule de la lisse mais celle-ci est comprise approximativement entre 5 et 10 cm.



Figure 85 : détection de personnes sous la lisse

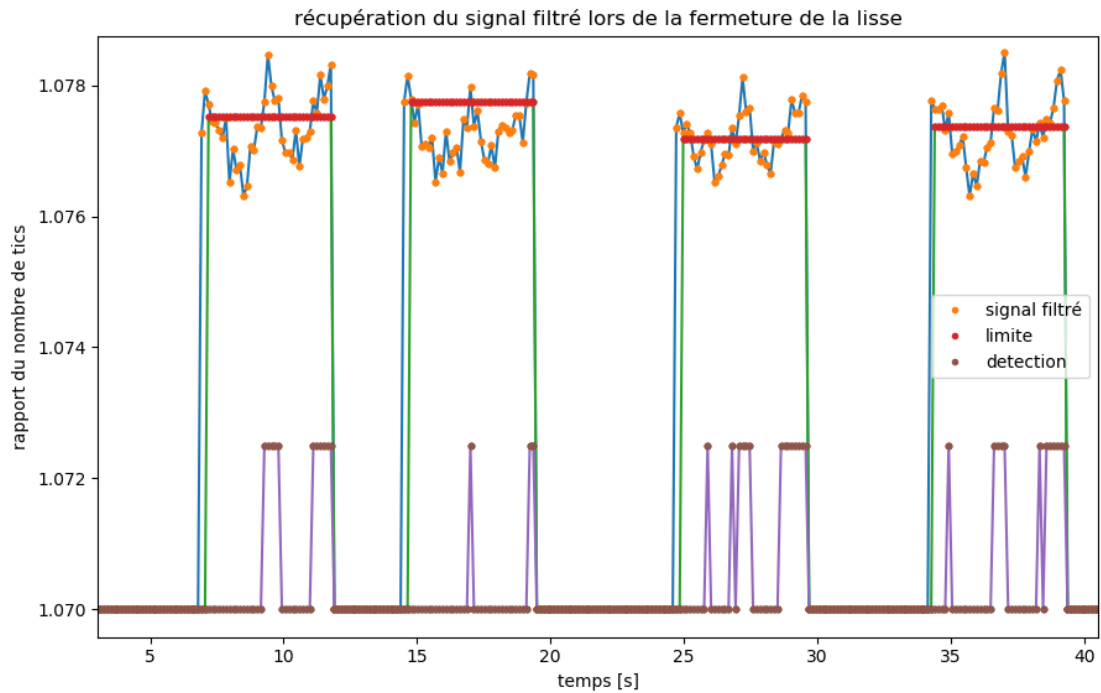


Figure 86 : détection de personnes sous la lisse humide

Nous remarquons sur la figure 86 que la forme des signaux reste quasiment identique au cas où les électrodes sont sèches. Les analyses faites précédemment sont entièrement valides dans notre cas. Il y a encore des présences de faux positif qui viennent fausser la détection. Nous espérons résoudre ce problème de faux positifs d'ici la présentation finale.

Donc cette expérience permet de conclure cette section en montrant que l'humidité n'influence pas la détection et que notre capacité d'environnement joue bien son rôle.

Conclusion

Dans cette section, nous avons montré les différents tests que nous avons réalisés ainsi que les résultats obtenus à l'aide des différents algorithmes. Nous avons réussi à montrer que l'algorithme ainsi que les circuits mis en place fonctionnent. Cependant, nous devons encore améliorer cette technologie en fournissant quelques recommandations dans la section suivante.

8. Recommandation

Nous sommes arrivées à la fin de ce travail et il reste encore des choses à améliorer ou à revoir. C'est pour cela que dans cette section nous allons vous fournir quelques recommandations afin que ce projet puisse avancer dans la bonne direction et pouvoir atteindre les objectifs fixés dans le cahier des charges.

Commençons par le capteur lui-même, nous avons vraiment l'impression que la capalisse d'un mètre est plus intéressante et plus sensible. C'est pour cela que nous vous recommandons de faire des tests avec des petits segments d'électrodes d'un demi-mètre ou d'un mètre au maximum. Mais il faudra veiller à ne pas dépasser le poids maximum et à avoir un encombrement minimum dans la lisse. Cette technique pourra, à notre sens, être plus efficace, car la variation de la capacité sera plus importante. On peut l'expliquer intuitivement par le fait qu'actuellement lorsqu'on rapproche notre tête de la lisse, on ne va modifier qu'un maximum de 10% du milieu diélectrique de la capacité coplanaire. Mais si nous prenons une électrode plus petite, on modifiera peut-être 70% voire même 80% du milieu diélectrique. Ce qui mènera bien évidemment à de plus grosses variations capacitives.

En ce qui concerne le système d'analyse, nous pensons que si celui-ci est installé un jour sur le microcontrôleur qu'utilise Automatic Systems, les signaux seront peut-être encore plus stables, mais cela ne reste qu'une supposition. Ce que nous recommanderons alors c'est d'implémenter tout l'algorithme sur le microcontrôleur et d'observer ce qui se passe.

Finalement si c'était à refaire, nous repartirons sur le même type de technologie, mais avec des électrodes à taille réduite et le software tournerait sur un microcontrôleur. Nous pensons qu'une fois ce dispositif monté, cela fonctionnera correctement même à grande vitesse.

9. Conclusion

La solution apportée à la problématique de départ qui était de savoir « comment protéger les usagers vulnérables lors de la fermeture de la lisse de barrières automatisées par l'intégration d'un capteur intelligent » s'est portée sur un capteur capacitif capable d'ignorer les effets de l'environnement. Nous avons fait ce choix après avoir analysé de manière détaillée les différentes technologies de détection existante dans le monde industriel et de la recherche.

Afin d'implémenter cette solution, nous avons tout d'abord réalisé des simulations par élément fini dans l'objectif de choisir le modèle le plus approprié à designer. Après avoir posé le modèle, nous avons installé le premier prototype, mis en place l'électronique et l'informatique nécessaire pour pouvoir récupérer et analyser les informations provenant du capteur. Par la suite, nous avons réalisé plusieurs tests dans le but de savoir si le capteur arriverait à détecter la pastèque qui représentait notre "corps fantôme".

Par ce projet, nous avons réalisé des avancées significatives. En effet, nous avons pu montrer que lorsque cette technologie était associée à "bon" algorithme, elle était en mesure de détecter des personnes sous la lisse et ce peu importe les conditions environnementales. Cependant, nous pouvons toujours, actuellement, améliorer cette technologie. Nous avons, par ailleurs, proposé plusieurs pistes dans ce rapport comme par exemple placer des barrettes plus courtes afin d'augmenter la variation capacitive et de la même manière la sensibilité. Nous avons aussi proposé de faire tourner le code sur un microcontrôleur dans le but de rendre le signal plus stable et de minimiser le nombre de faux positif.

À l'avenir, ce type de technologie pourrait équiper toutes les barrières d'Automatic System. En effet, une fois amélioré et industrialisé, il sera plus aisé d'équiper et de mettre aux normes toutes les barrières de chez A.S, même celle déjà en fonction, car il suffira uniquement de changer la lisse et la barrière répondra aux nouvelles normes.

Enfin, pour terminer, cette technologie permettra à l'utilisateur vulnérable et à l'outil de "coexister" sans un risque d'incidents graves.

10. Annexes

Ces annexes ont été créées afin de vous partager les différents paramètres utiles à la reconstruction d'un tel dispositif. Nous commencerons par vous fournir les différents paramètres utiles à la construction du modèle Comsol, puis nous allons vous fournir les plans des circuits électroniques et nous finirons par vous mettre les différents codes informatiques utilisés dans le cadre de ce mémoire.

10.1. Annexe A : Réalisation du modèle Comsol

Nous avons réalisé de nombreux modèles Comsol, mais nous décrivons seulement les paramètres du modèle final. Pour réaliser ce projet, nous avons utilisé le modèle physique 'électrostatique' de Comsol.

La lisse

Nous avons tout d'abord réalisé la forme de la lisse à l'aide de multiple forme de base. Il faut savoir que Comsol n'offre pas beaucoup d'outils de dessin comme Solidworks et que donc pour dessiner une forme comme la lisse, il faut passer à travers plusieurs étapes avant d'obtenir cette forme. Vous trouverez les différentes étapes dans le modèle Comsol fourni, il suffit d'aller dans 'Component' puis dans 'Geometry'.

On retrouve sur la figure 77 le plan complet de la lisse fourni par A.S. Vous remarquerez qu'il y'a qu'un plan de coupe de la lisse, cela est totalement normale, car A.S. reçoit les lisses dans une taille standard, puis elle les découpe en fonction de leurs besoins.

Puis nous avons déterminé le matériau de cette structure. Comme nous le constatons sur le plan à la figure 87, la lisse est faite en aluminium et donc le matériau sélectionner sur Comsol est de « l'Aluminium [Solid,Bulk] ». Pour finir, comme la lisse sert de garde, celle-ci est mise à la masse (Ground).

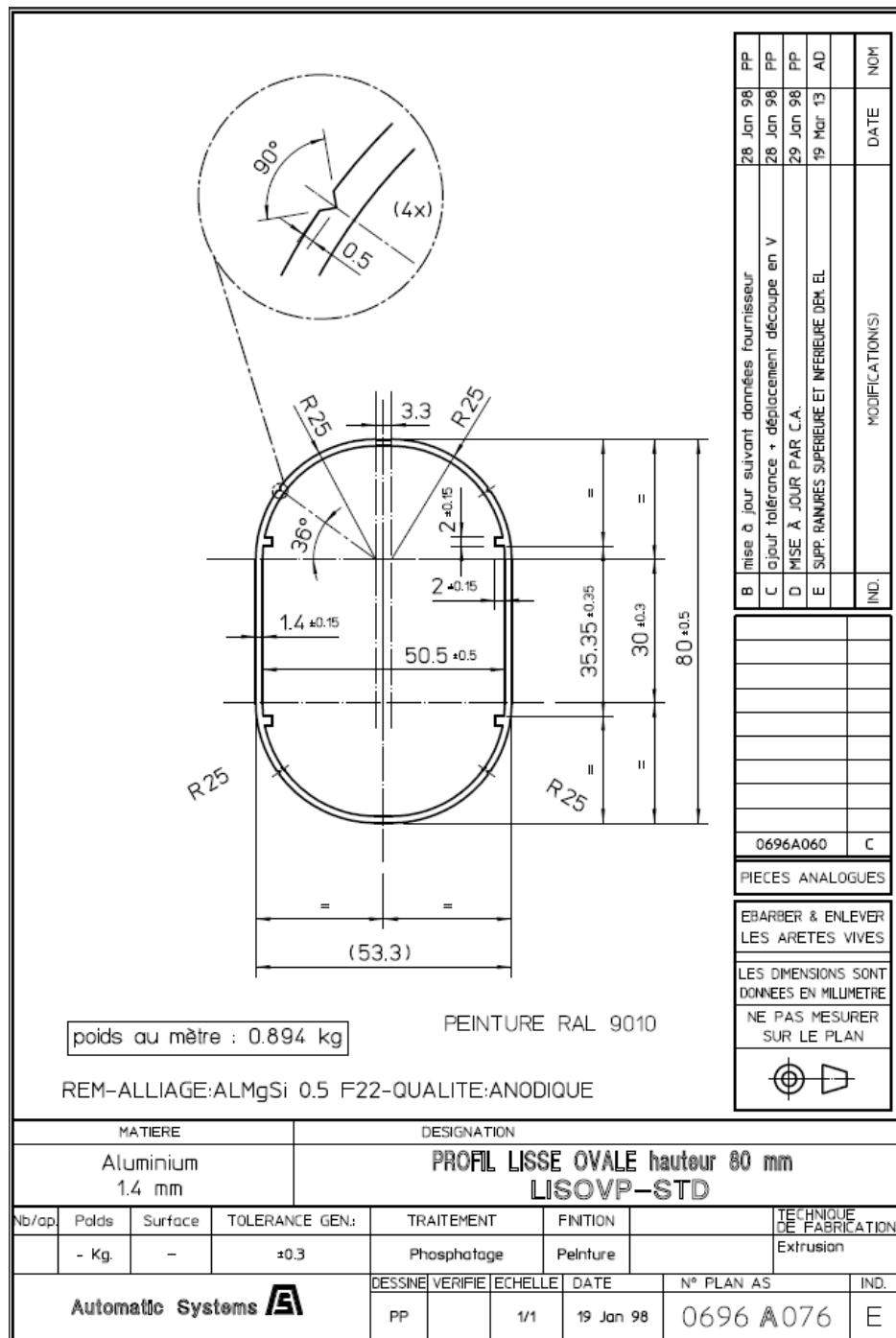


Figure 87: Plan de coupe de la lisse

L'environnement

Une fois la lisse réalisée nous avons ajouté l'environnement/l'air qui est représenté par un cercle d'un mètre de rayon centré en 0 (Figure 88). Cet environnement va permettre de voir comment les lignes de champs électriques vont se distribuer dans l'air.

Sur Comsol nous avons décidé de mettre le matériau de ce cercle en « Air [gas] » et nous avons mis la permittivité relative égale à 1.

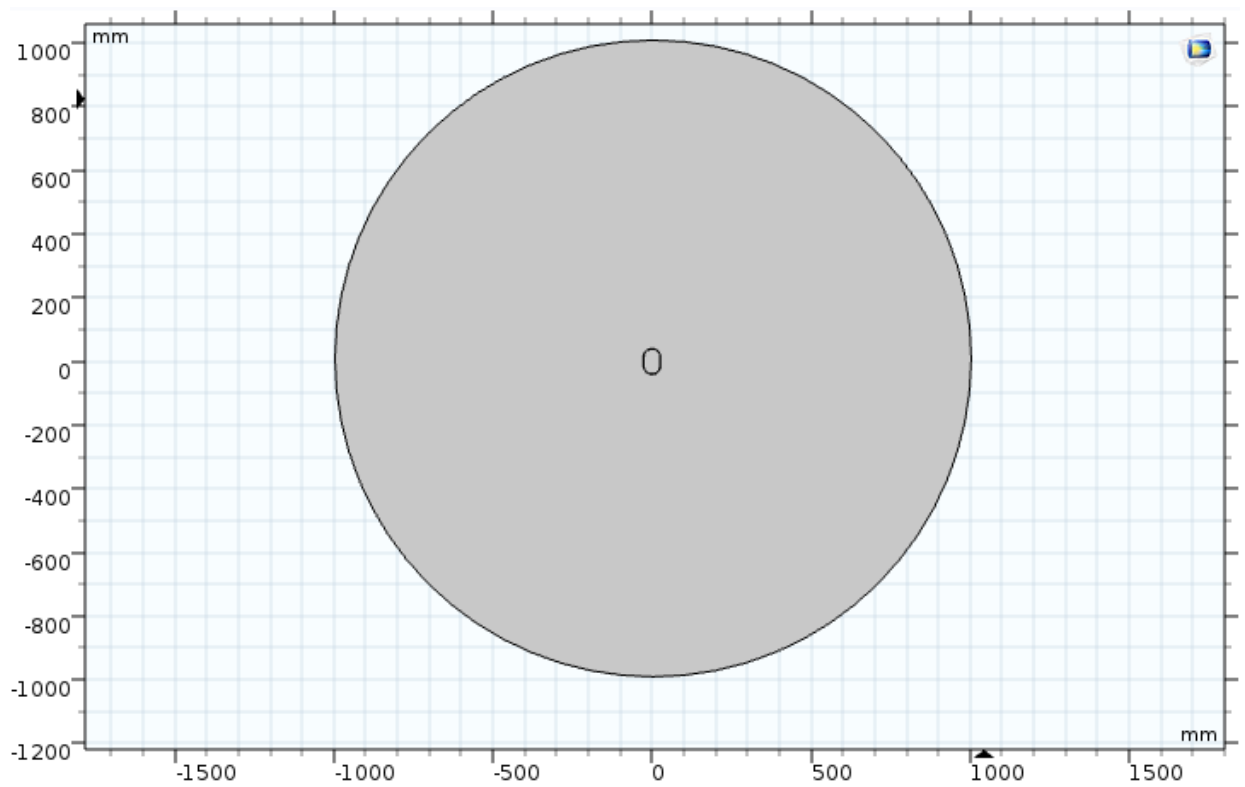


Figure 88 : Cercle d'air

Les électrodes et leur fixation

Une fois le modèle basique crée, nous allons intégrer nos électrodes au dispositif. Pour cela il faut dessiner les différentes électrodes de notre système qui sont simplement des rectangles. Nous avons deux types d'électrodes, les latérales qui sont de dimensions 15mmx1.5mm, et la centrale qui est de dimensions 5mmx1.5mm. Puis il suffit de les positionner sur le bon endroit, ce qui s'avère être une tâche un peu méticuleuse. Les détails des positions ainsi que des tensions se trouvent dans le tableau suivant.

Éléments	Tension [V]	Position x [mm]	Position y [mm]
Référence [centre de la lisse]	/	0	0
Position du centre de l'électrode gauche	0	-5.61	-42.65
Position du centre de l'électrode droite	3.3	5.61	-42.65
Position du centre de l'électrode centrale	3.3	0	-40.99

Tableau 5 : Position des électrodes

En ce qui concerne le matériau utilisé, ces électrodes sont composées aussi d'aluminium tout comme la lisse. Nous avons donc utilisé le même aluminium que celui utilisé pour la lisse dans le modèle Comsol.

Pour fixer ces électrodes en pratique, nous utilisons un papier collant qui est super-collant, double face et isolant électrique. Pour simuler ce papier collant isolant sur Comsol, nous avons

mis une couche de silicone[solide] d'une épaisseur de 0.24mm sur tout le bas de la lisse. Le fait qu'il soit présent tout le long de cette zone permet de changer facilement les positions des électrodes sans commencer à changer les positions des isolants. De plus ça ne change pas grand-chose en termes de résultats, car ce n'est pas un élément déterminant de la simulation.

La pastèque

Comme dit à plusieurs reprises dans le rapport, la pastèque représente dans notre cas la tête humaine et c'est ce qui a été utilisé dans nos tests. Afin de pouvoir mesurer les effets de perturbations d'une pastèque sur le système, nous avons désigné un cercle qui deviendra une sphère sur le modèle 3D. Nous avons fixé son rayon à 100mm ce qui représente plus ou moins la taille de la tête humaine. En ce qui concerne le matériau utilisé, nous avons utilisé de l'eau, car celle-ci représente plus de 90% de la composition de notre tête. Nous avons mis la permittivité relative de l'eau égale à 80 c'est ce qui va permettre de faire varier les lignes de champs.

Conseil

Lorsqu'on fait des simulations Comsol, il est intéressant d'utiliser le tableau des paramètres. Il permet de faire les modifications super rapidement et instantanément sur tous les éléments qui contiennent ce paramètre. Dans notre cas par exemple on l'a beaucoup utilisé pour déplacer les électrodes ou encore le cercle. Vous pouvez trouver les différents paramètres utilisés dans le modèle fourni en allant dans « Global Définition ».

10.2. Annexe B : Circuits électroniques

Comme nous l'avons vu dans la section électronique, nous avons designer des PCB. Dans cette section nous allons voir les Board et les schémas Eagle réalisés.

Carte d'interface

Cette carte et le centre d'alimentation à partir duquel toutes les cartes s'alimentent et échangent leurs informations.

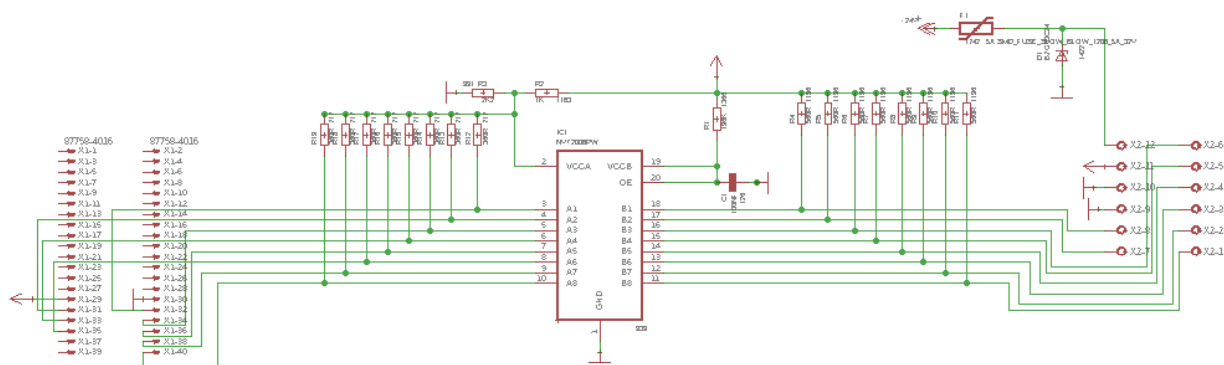


Figure 89 : Circuit de la carte d'interface

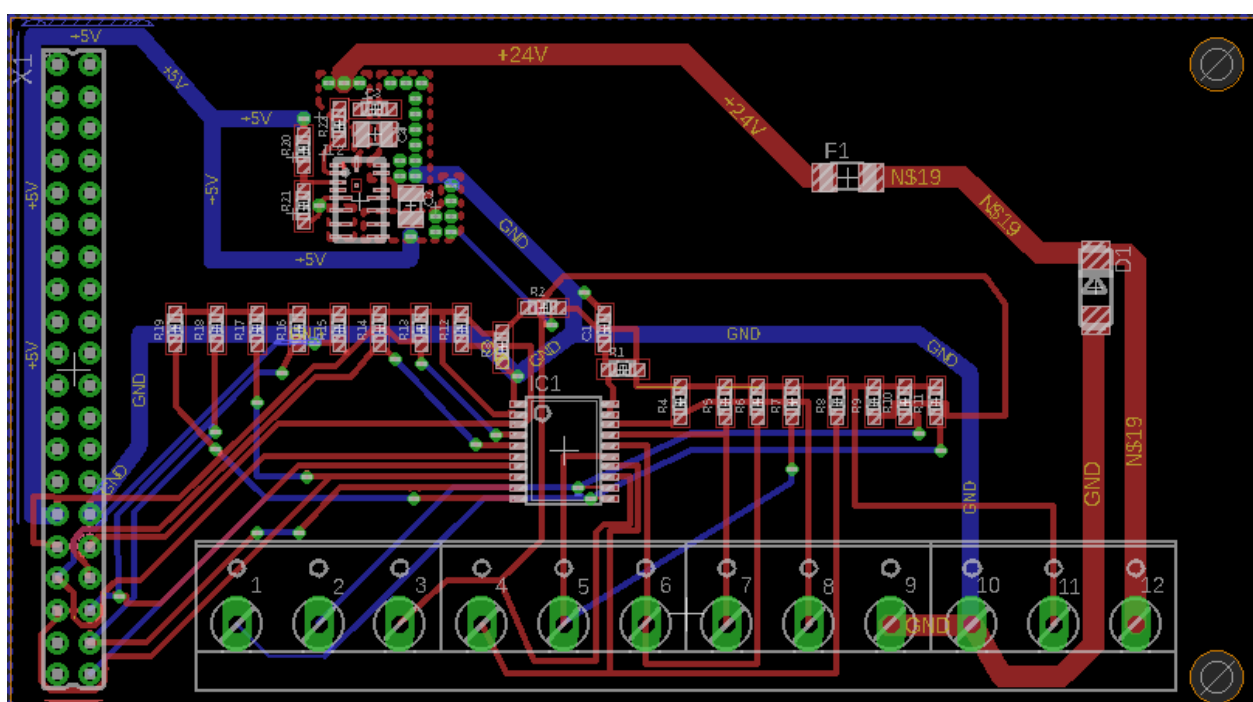


Figure 90 : Board de la carte d'interface

Carte oscillateurs

Cette carte est très petite, car elle devra être capable de rentrer dans la lisse.

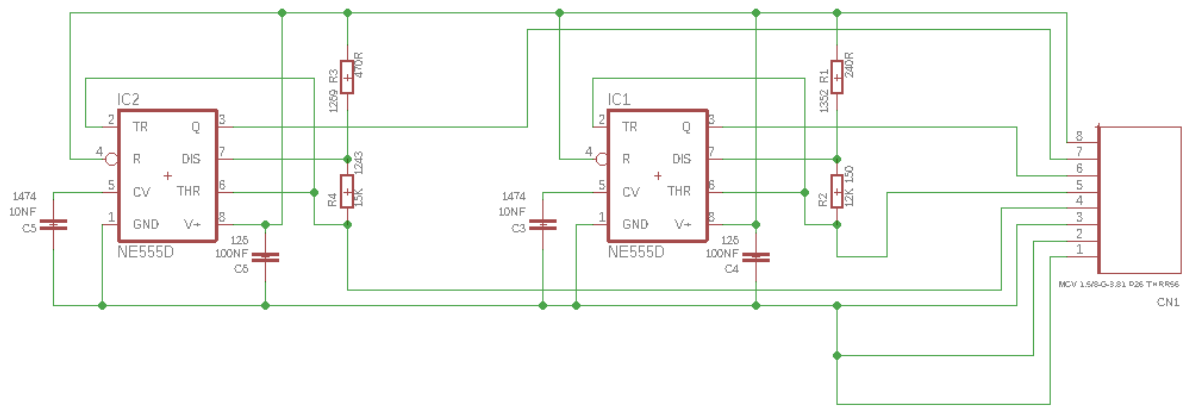


Figure 91 : Circuit oscillateurs

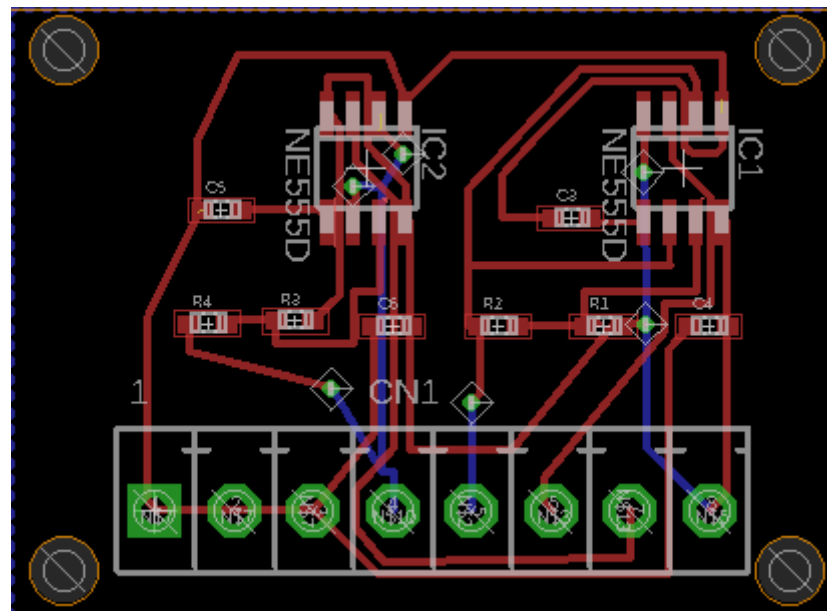


Figure 92 : Board de l'oscillateur

10.3. Annexe C : Les codes informatiques

Code python tournant sur le Raspberry-Pi

```
1 #!/usr/bin/python3
2 from subprocess import check_output
3 import RPi.GPIO as GPIO
4 import time
5 from time import sleep
6 import spidev
7 import os
8 import math
9 import MyLCD
10 import MyMCP2515
11 import numpy as np
12 import csv
```

```

13
14 # Initialisation du système
15 MyLCD = MyLCD.I2C_LCD()
16
17 MyARM_ResetPin = 19 # Pin 4 of connector = BCM19 = GPIO[1]
18
19 MySPI_FPGA = spidev.SpiDev()
20 MySPI_FPGA.open(0,0)
21 MySPI_FPGA.max_speed_hz = 500000
22
23 #Permet de tout reset avant le commencement
24
25 GPIO.setmode(GPIO.BCM)
26 GPIO.setwarnings(False)
27 GPIO.setup(MyARM_ResetPin, GPIO.OUT)
28 GPIO.output(MyARM_ResetPin, GPIO.HIGH)
29 sleep(0.1)
30 GPIO.output(MyARM_ResetPin, GPIO.LOW)
31 sleep(0.1)
32 |
33 #parametre sur lesquels on peut jouer
34 wait_time = 0.05 # Temps d'attente entre chaque boucle
35
36 # Initialisation des parametre du code
37 i = 0 # Les indexs
38 t = 0
39 median = []
40 rapport_vect = []
41 tps_med = []
42 delta_t = 0
43 delta_freq_mes=0
44 delta_freq_env=0
45 rapport_freq = 0
46 counter=0
47 ok_write=False
48 detection = 0
49 lim_act = 0 # elle active la recherche le calcul du seuil(lorsqu'il vaut 2 = barrière en cours de fermeture)
50 limit = 0# valeur du seuil une fois calculer
51
52 #donne l'info sur l'etat de la barriere: 0= transit 1=fermé 2=ouvert mis sur 2 bits
53 etat = 0
54 open_pos = False
55 # creation du fichier d ecriture
56 title = 'youpi.csv'
57 c = csv.writer(open(title,'w'))
58 c.writerow(['Time','f_mes','f_env','bit_etat','f_m/f_e','tps_median','median'])
59
60 dy=0
61
62 print("Let's Go")
63
64 count_mes_1 = MySPI_FPGA.xfer2([0x01, 0x00, 0x00, 0x00, 0x00])
65 count_env_1 = MySPI_FPGA.xfer2([0x03, 0x00, 0x00, 0x00, 0x00])
66 count_etat_init = MySPI_FPGA.xfer2([0x06, 0x00, 0x00, 0x00, 0x00])
67
68 freq_mes1 = count_mes_1[1]*16777216 + count_mes_1[2]*65536 + count_mes_1[3]*256 + count_mes_1[4]
69 freq_env1 = count_env_1[1]*16777216 + count_env_1[2]*65536 + count_env_1[3]*256 + count_env_1[4]
70 etat_init = count_etat_init[4]
71
72 initial_time= time.time()
73

```

```

74 while True:
75     #récupération des valeurs
76     count_mes_2 = MySPI_FPGA.xfer2([0x01, 0x00, 0x00, 0x00, 0x00])
77     count_env_2 = MySPI_FPGA.xfer2([0x03, 0x00, 0x00, 0x00, 0x00])
78     count_etat = MySPI_FPGA.xfer2([0x06, 0x00, 0x00, 0x00, 0x00])
79
80     time_now = time.time()
81     freq_mes2 = count_mes_2[1]*16777216 + count_mes_2[2]*65536 + count_mes_2[3]*256 + count_mes_2[4]
82     freq_env2 = count_env_2[1]*16777216 + count_env_2[2]*65536 + count_env_2[3]*256 + count_env_2[4]
83     etat = count_etat[4]
84
85     #Seulement pour que ca colle mieux au graphe
86     if(etat==1):
87         etat=1.0688
88         open_pos=False
89     elif(etat==2):
90         etat=1.07
91         open_pos=True
92     else:
93         etat=1.0692
94
95     delta_t = round((time_now-initial_time),4)
96
97     if((freq_mes2-freq_mes1)>0 and (freq_env2-freq_env1)>0):
98         delta_freq_mes = (freq_mes2-freq_mes1)
99         delta_freq_env = (freq_env2-freq_env1)
100
101     try:
102         rapport_freq = round((delta_freq_mes)/(delta_freq_env),5)
103     except ZeroDivisionError:
104         rapport_freq = 0
105         MyLCD.writeString("err. Rprt")
106     rapport_vect.append(rapport_freq)
107     freq_env1=freq_env2
108     freq_mes1=freq_mes2
109     #####
110     ##### Code Analyse #####
111     #####
112     if(i>4): # rentre dans cette condition tout les 4 points
113         median.append(np.median(rapport_vect[-4:-1])) #filtre
114         tps_med.append(delta_t) # récupère le temps pour ce point de médiane
115         if(open_pos==True and etat==1.0692 and len(median)>1):#en cours de fermeture
116             dy = median[-1] # dernier élément de la médiane
117             # lim_act plus petit que 0.5 cela signifie qu'on a pas encore de limite calculé
118             if(lim_act<=0.5):
119                 if(median[-2]>median[-1] and lim_act>0):# si il y'a une pente négatif: fixe le seuil
120                     limit=0.9996271*median[-2]# avec cette formule
121                     print(limit)
122                     lim_act=1
123                 else:
124                     lim_act=0.5 # permet de dire que le premier point est passé
125             if(median[-1]<limit and lim_act==1):# commence à checker
126                 lim_act=2
127                 print('Je commence')
128             if(median[-1]>limit and lim_act==2):# detecte une détection
129                 detection = 1.0725
130                 print('detection')
131             else:
132                 detection = 1.07
133
134         else:# valeur par default
135             dy=1.07
136             lim_act=0
137             limit=1.07 # ca va mieux pour les plots
138             detection=1.07 # ca va mieux pour les plots
139     i=i0 # permet de faire une mediane de groupe, commentez la ligne pour avoir une mediane gliss.

```

```

140     ok_write=True
141     if(ok_write): # ecriture dans le fichier
142         c.writerow([str(delta_t),str(delta_freq_mes),str(delta_freq_env),str(etat),str(rapport_freq),str(tps_med[-1]),str(median[-1]),str(dy),str(limit),str(detection)])
143
144     sleep(wait_time)
145     i=i+1
146
147     if(delta_t>t*1):
148         print(delta_t)
149         print(etat)
150         t=t+1
151
152     MyLCD.writeString("Finish")
153     MyLCD.setPosition(2,0)
154     MyLCD.writeString("detect:"+str(counter))

```

Programmation FPGA : Code SystemVerilog

La partie SPI de ce code provient du cours de digital dispensé par le professeur Legat.

Le code antiglitch provient des assistant du projet robot.

Le code principal :

```

1
2 //=====
3 // This code is generated by Terasic System Builder
4 //=====
5
6 module ENIGMA(
7
8     //////////// CLOCK ////////////
9     CLOCK_50,
10
11     //////////// LED ////////////
12     LED,|
13
14     //////////// KEY ////////////
15     KEY,
16
17     //////////// SW ////////////
18     SW,
19
20     //////////// SDRAM ////////////
21     DRAM_ADDR,
22     DRAM_BA,
23     DRAM_CAS_N,
24     DRAM_CKE,
25     DRAM_CLK,
26     DRAM_CS_N,
27
28     DRAM_DQ,
29     DRAM_DQM,
30     DRAM_RAS_N,
31     DRAM_WE_N,
32
33     //////////// EPCS ////////////
34     EPCS_ASDO,
35     EPCS_DATA0,
36     EPCS_DCLK,
37     EPCS_NCSO,
38
39     //////////// Accelerometer and EEPROM ////////////
40     G_SENSOR_CS_N,
41     G_SENSOR_INT,
42     I2C_SCLK,
43     I2C_SDAT,
44
45     //////////// ADC ////////////
46     ADC_CS_N,
47     ADC_SADDR,
48     ADC_SCLK,
49     ADC_SDAT,
50
51     //////////// 2x13 GPIO Header ////////////

```

```

51     GPIO_2,
52     GPIO_2_IN,
53
54     ////////// GPIO_0, GPIO_0 connect to GPIO Default //////////
55     GPIO_0,
56     GPIO_0_IN,
57
58     ////////// GPIO_1, GPIO_1 connect to GPIO Default //////////
59     GPIO_1,
60     GPIO_1_IN
61 );
62
63 //=====
64 //  PARAMETER declarations
65 //=====
66
67 //=====
68 //  PORT declarations
69 //=====
70
71
72 ////////// CLOCK //////////
73 input          CLOCK_50;
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121

```

```

-
////////// LED //////////
output          [7:0]    LED;

////////// KEY //////////
input          [1:0]    KEY;

////////// SW //////////
input          [3:0]    SW;

////////// SDRAM //////////
output          [12:0]   DRAM_ADDR;
output          [1:0]   DRAM_BA;
output          DRAM_CAS_N;
output          DRAM_CKE;
output          DRAM_CLK;
output          DRAM_CS_N;
input           [15:0]   DRAM_DQ;
output          [1:0]   DRAM_DQM;
output          DRAM_RAS_N;
output          DRAM_WE_N;

////////// EPCS //////////
output          EPCS_ASDO;
input           EPCS_DATA0;

output          EPCS_DCLK;
output          EPCS_NCS0;

////////// Accelerometer and EEPROM //////////
output          G_SENSOR_CS_N;
input           G_SENSOR_INT;
output          I2C_SCLK;
input           I2C_SDAT;

////////// ADC //////////
output          ADC_CS_N;
output          ADC_SADDR;
output          ADC_SCLK;
input           ADC_SDAT;

////////// 2x13 GPIO Header //////////
input           [12:0]   GPIO_2;
input           [2:0]    GPIO_2_IN;

////////// GPIO_0, GPIO_0 connect to GPIO Default //////////
input           [33:0]   GPIO_0;
input           [1:0]    GPIO_0_IN;

```

```

122 ////////////// GPIO_1, GPIO_1 connect to GPIO Default //////////////
123 inout      [33:0]    GPIO_1;
124 input      [1:0]     GPIO_1_IN;
125
126 //=====//
127 // declarations des variables //
128 //=====//
129
130 logic [7:0] led_reg;
131 logic clk,reset;
132 logic elec_mes,elec_env;
133 logic [31:0] count_mes, count_env, sous_count,etat_bar;
134
135 //=====//
136 //      Controle de LED      //
137 //=====//
138
139 assign clk = CLOCK_50;
140 assign reset = ~KEY[0];
141
122 ////////////// GPIO_1, GPIO_1 connect to GPIO Default //////////////
123 inout      [33:0]    GPIO_1;
124 input      [1:0]     GPIO_1_IN;
125
126 //=====//
127 // declarations des variables //
128 //=====//
129
130 logic [7:0] led_reg;
131 logic clk,reset;
132 logic elec_mes,elec_env;
133 logic [31:0] count_mes, count_env, sous_count,etat_bar;
134
135 //=====//
136 //      Controle de LED      //
137 //=====//
138
139 assign clk = CLOCK_50;
140 assign reset = ~KEY[0];
141
142 //=====//
143 //  Récupération signaux avec un entigltch, //
144 //  Permet de ne pas compter les effets parasites//
145 //=====//
146 antiGlitch bit_ouverture(
147     .clk (clk),
148     .reset (reset),
149     .in (GPIO_2[9]),
150     .out (ouv_bit)
151 );
152
153 antiGlitch bit_fermeture(
154     .clk (clk),
155     .reset (reset),
156     .in (GPIO_2[11]),
157     .out (ferm_bit)
158 );
159
160 antiGlitch elec_mesure(
161     .clk (clk),
162     .reset (reset),
163     .in (GPIO_2[1]),
164     .out (elec_mes)
165 );

```

```

142 //=====//
143 // Récupération signaux avec un entigltch, //
144 // Permet de ne pas compter les effets parasites//
145 //=====//
146 antiGlitch bit_ouverture(
147     .clk (clk),
148     .reset (reset),
149     .in (GPIO_2[9]),
150     .out (ouv_bit)
151 );
152
153 antiGlitch bit_fermeture(
154     .clk (clk),
155     .reset (reset),
156     .in (GPIO_2[11]),
157     .out (ferm_bit)
158 );
159
160 antiGlitch elec_mesure(
161     .clk (clk),
162     .reset (reset),
163     .in (GPIO_2[1]),
164     .out (elec_mes)
165 );
166
167 antiGlitch elec_environement(
168     .clk (clk),
169     .reset (reset),
170     .in (GPIO_2[3]),
171     .out (elec_env)
172 );
173 //=====//
174 // Encoders contient le compteur qui permet de compter //
175 // Le nombre de tics //
176 //=====//
177 encoders encoders_inst(
178     .clk (clk),
179     .reset (reset),
180     .elec_mes (elec_mes),
181     .elec_env (elec_env),
182     .ouv_bit (ouv_bit),
183     .ferm_bit (ferm_bit),
184     .count_mes (count_mes),
185     .count_env (count_env),
186     .etat_bar (etat_bar)
187 );
188
189 antiGlitch elec_environement(
190     .clk (clk),
191     .reset (reset),
192     .in (GPIO_2[3]),
193     .out (elec_env)
194 );
195 //=====//
196 // Encoders contient le compteur qui permet de compter //
197 // Le nombre de tics //
198 //=====//
199 encoders encoders_inst(
200     .clk (clk),
201     .reset (reset),
202     .elec_mes (elec_mes),
203     .elec_env (elec_env),
204     .ouv_bit (ouv_bit),
205     .ferm_bit (ferm_bit),
206     .count_mes (count_mes),
207     .count_env (count_env),
208     .etat_bar (etat_bar)
209 );

```

```

189 //=====
190 // SPI
191 //=====
192 logic [31:0] WriteDataM,readDataM,DataAdrW,DataAdrR;
193 logic spi_write;
194 logic [31:0] spi_data;
195 logic spi_clk, spi_cs, spi_mosi, spi_miso;
196
197 spi_slave spi_slave_instance(
198     .SPI_CLK (spi_clk),
199     .SPI_CS (spi_cs),
200     .SPI_MOSI (spi_mosi),
201     .SPI_MISO (spi_miso),
202     .Data_WE (spi_write),
203     .Data_Addr_read(DataAdrR),
204     .Data_Addr_write(DataAdrW),
205     .Data_Write (WriteDataM),
206     .Data_Read (spi_data),
207     .Clk (clk)
208 );
209
210 assign spi_clk = GPIO_0[11]; // SCLK = pin 16 = GPIO_11
211 assign spi_cs = GPIO_0[9]; // CE0 = pin 14 = GPIO_9
212 assign spi_mosi = GPIO_0[15]; // MOSI = pin 20 = GPIO_15
213 assign GPIO_0[13] = spi_cs ? 1'bz : spi_miso; // MISO = pin 18 = GPIO_13
214
189 //=====
190 // SPI
191 //=====
192 logic [31:0] WriteDataM,readDataM,DataAdrW,DataAdrR;
193 logic spi_write;
194 logic [31:0] spi_data;
195 logic spi_clk, spi_cs, spi_mosi, spi_miso;
196
197 spi_slave spi_slave_instance(
198     .SPI_CLK (spi_clk),
199     .SPI_CS (spi_cs),
200     .SPI_MOSI (spi_mosi),
201     .SPI_MISO (spi_miso),
202     .Data_WE (spi_write),
203     .Data_Addr_read(DataAdrR),
204     .Data_Addr_write(DataAdrW),
205     .Data_Write (WriteDataM),
206     .Data_Read (spi_data),
207     .Clk (clk)
208 );
209
210 assign spi_clk = GPIO_0[11]; // SCLK = pin 16 = GPIO_11
211 assign spi_cs = GPIO_0[9]; // CE0 = pin 14 = GPIO_9
212 assign spi_mosi = GPIO_0[15]; // MOSI = pin 20 = GPIO_15
213 assign GPIO_0[13] = spi_cs ? 1'bz : spi_miso; // MISO = pin 18 = GPIO_13
214
215 spi_regw spi_regw_instance(
216     .clk (clk),
217     .reset (reset),
218     .count_mes (count_mes),
219     .count_env (count_env),
220     .etat_bar (etat_bar),
221     .spi_cs (spi_cs),
222     .DataAdrW (DataAdrW),
223     .WriteDataM (WriteDataM),
224     .spi_write (spi_write)
225 );
226
227 //=====//
228 // Controle de LED //
229 //=====//
230 // LED logic
231
232 assign LED = led_reg;
233 assign sous_count = count_mes-count_env;
234 always_ff @(posedge clk, posedge reset)
235 begin
236     if(reset) led_reg[7:0] <= 8'd0;
237     else led_reg[7:0] <= count_env[7:0];
238 end
239
240 endmodule

```

```

215 spi_regw spi_regw_instance(
216     .clk      (clk),
217     .reset    (reset),
218     .count_mes (count_mes),
219     .count_env (count_env),
220     .etat_bar  (etat_bar),
221     .spi_cs    (spi_cs),
222     .DataAdrW  (DataAdrW),
223     .WriteDataM (WriteDataM),
224     .spi_write (spi_write)
225 );
226
227 //=====//
228 //      Controle de LED      //
229 //=====//
230 // LED logic
231
232 assign LED = led_reg;
233 assign sous_count = count_mes-count_env;
234 always_ff @(posedge clk, posedge reset)
235 begin
236     if(reset) led_reg[7:0] <= 8'd0;
237     else led_reg[7:0] <= count_env[7:0];
238 end
239
240 endmodule

```

```

215 spi_regw spi_regw_instance(
216     .clk      (clk),
217     .reset    (reset),
218     .count_mes (count_mes),
219     .count_env (count_env),
220     .etat_bar  (etat_bar),
221     .spi_cs    (spi_cs),
222     .DataAdrW  (DataAdrW),
223     .WriteDataM (WriteDataM),
224     .spi_write (spi_write)
225 );
226
227 //=====//
228 //      Controle de LED      //
229 //=====//
230 // LED logic
231
232 assign LED = led_reg;
233 assign sous_count = count_mes-count_env;
234 always_ff @(posedge clk, posedge reset)
235 begin
236     if(reset) led_reg[7:0] <= 8'd0;
237     else led_reg[7:0] <= count_env[7:0];
238 end
239
240 endmodule

```

Le module Antiglitch :

```

1 module antiGlitch(
2     input logic clk,
3     input logic reset,
4     input logic in,
5     output logic out
6 );
7
8     logic [3:0] glitch;
9
10    always_ff @ (posedge clk)
11    if(reset)
12    begin
13        glitch <= 4'b0;
14        out <= 1'b0;
15    end
16    else
17    begin
18        glitch <= {glitch[2:0],in};
19    case(glitch)
20        4'b0: out <= 1'b0;
21        4'b1111: out <= 1'b1;
22        default: out <= out;
23    endcase
24    end
25 endmodule

```

Le module Encoders :

```
1 module encoders(  
2     input logic clk,  
3     input logic reset,  
4     input logic elec_mes,  
5     input logic elec_env,  
6     input logic ouv_bit,  
7     input logic ferm_bit,  
8     output logic [31:0] count_mes,  
9     output logic [31:0] count_env,  
10    output logic [31:0] etat_bar  
11 );  
12  
13 logic [31:0] Count_Pos_mes, Count_Neg_mes; // Compteur pour la mesure  
14 logic [31:0] Count_Pos_env, Count_Neg_env; //Compteur pour l'environnement  
15 logic [31:0] etat_0, etat_1, etat_2, etat_3; // les etat de la barriere, ouvert-ferme  
16  
17 //compteur du nombre de flancs montant et descendant  
18 always_ff @(posedge elec_mes, posedge reset) begin  
19     if(reset) Count_Pos_mes = 32'd0;  
20     else Count_Pos_mes = Count_Pos_mes + 32'd1;  
21 end  
22 always_ff @(negedge elec_mes, posedge reset) begin  
23     if(reset) Count_Neg_mes = 32'd0;  
24     else Count_Neg_mes = Count_Neg_mes + 32'd1;  
25 end  
26  
27 always_ff @(posedge elec_env, posedge reset) begin  
28     if(reset) Count_Pos_env = 32'd0;  
29     else Count_Pos_env = Count_Pos_env + 32'd1;  
30 end  
31 always_ff @(negedge elec_env, posedge reset) begin  
32     if(reset) Count_Neg_env = 32'd0;  
33     else Count_Neg_env = Count_Neg_env + 32'd1;  
34 end  
35 //Etat de la barriere sur 2 bits  
36 // 01 = ouvert  
37 // 10 = ferme  
38 // 00 = en transit  
39 always_ff @(posedge ouv_bit, posedge reset) begin  
40     if(reset) etat_0 = 32'd0;  
41     else etat_0 = etat_0 + 32'd2;  
42 end  
43 always_ff @(negedge ouv_bit, posedge reset) begin  
44     if(reset) etat_1 = 32'd0;  
45     else etat_1 = etat_1 - 32'd2;  
46 end  
47 always_ff @(posedge ferm_bit, posedge reset) begin  
48     if(reset) etat_2 = 32'd0;  
49     else etat_2 = etat_2 + 32'd1;  
50 end  
51 always_ff @(negedge ferm_bit, posedge reset) begin  
52     if(reset) etat_3 = 32'd0;  
53     else etat_3 = etat_3 - 32'd1;  
54 end  
55 assign count_mes = Count_Pos_mes + Count_Neg_mes;  
56 assign count_env = Count_Pos_env + Count_Neg_env;  
57 assign etat_bar = etat_0 + etat_1 + etat_2 + etat_3;  
58 endmodule
```

Le module SPI :

```

1 //-----
2 //
3 // Write SPI Registers Module
4 //
5
6 module spi_regw(
7     input logic clk, reset, spi_cs,
8     input logic [31:0] count_mes, count_env, etat_bar,
9     output logic [31:0] DataAdrW, WriteDataM,
10    output logic spi_write
11 );
12
13 //--- SPI FSM -----
14
15 typedef enum logic [1:0] {S0,S1,S2} statetype;
16 statetype state, nextstate;
17
18 // State Register & Bit counter & SPI Register & MISO
19
20 always_ff @(posedge clk,posedge reset) begin
21
22     if (reset) state <= S0;
23     else state <= nextstate;
24
25 end
26
27 // Next State Logic
28 assign spi_write = ~spi_cs;
29
30 always_comb begin
31
32     // Default value
33
34
35     case (state)
36     S0: begin DataAdrW = 32'd4;
37             WriteDataM <= count_mes;
38             nextstate <= S1; end
39
40     S1: begin DataAdrW = 32'd12;
41             WriteDataM <= count_env;
42             nextstate <= S2; end
43
44     S2: begin DataAdrW = 32'd24;
45             WriteDataM <= etat_bar;
46             nextstate <= S0; end
47
48     default: begin nextstate <= S0;
49                 DataAdrW <= 32'b0;
50                 WriteDataM <= 32'b0; end
51     endcase
52 end
53
54 endmodule

```

11. Bibliographie

- [1] Ingénieurs d'Automatic Systems : Echange orale durant l'année.
- [2] Automatic Systems, *Profil de la société*, [Consulté le 15/10/2017]
<https://www.automatic-systems.com/fr/a-propos/profile-societe/>.
- [3] S. Dassy, *Dossier technique électrique du capteur capacitif développé chez Automatic Systems*, Mai 2002.⁴
- [4] Hristov L., *Capacitive Proximity Detection in Automotive Industry*, Atmel 2010
- [5] PÉREZ V., *Détecteur de proximité : faites le bon choix*, Forum des technologie, janvier-février 2015.
- [6] Bentchikou Moh. Cours_GPA668_E13_02_03_04_05.pdf.
- [7] Rockwell Automation, [Consulté le 19/01/2018]
<http://ab.rockwellautomation.com/fr/Sensors-Switches/Clear-Object-Detection> .
- [8] Bannerengineering, [Consulté le 05/06/2017]
<https://www.bannerengineering.com/fr/fr/company/expert-insights/ultrasonic-sensors-101.html>
- [9] Jose A. Cantoral-Ceballos, Nurgiyatna Nurgiyatna, Paul Wright, John Vaughan, Christine Brown-Wilson, Patricia J. Scully, and Krikor B. Ozanyan, *Intelligent Carpet System, Based on Photonic Guided-Path Tomography, for Gait and Balance Monitoring in Home Environments*, January 2015
- [10] De Luca A., Flacco F., *Integrated control for PHRI: Collision avoidance, detection, reaction and collaboration*.
- [11] Alla R., *Détection de chute à l'aide d'une caméra de profondeur*, Mémoire de l'université de Montréal. Avril 2013.
- [12] Wikipédia, [Consulté le 29/10/2017]
https://fr.wikipedia.org/wiki/D%C3%A9tection_de_personnes
- [13] Southwest Microwave, [Consulté le 27/10/2017]
<http://www.southwestmicrowave.com/pdfs/MicroTrack-Data-Sheet-FR.pdf> .
- [14] Suisse alarme, [Consulté le 27/10/2017]
<https://www.suisse-alarme.ch/detection-dintrusion-par-cable-enterre-fiable-sans-entretien/> .
- [15] Antoine Belconde, *Modélisation de la détection de présence humaine*
- [16] Wikipédia, [Consulté le 30/1/2017] <https://fr.wikipedia.org/wiki/Vibrisse>
- [17] Schlegl T., Kröger T., Gaschler A., Khatib O. and Zangl H., *Virtual Whiskers: Highly Responsive Robot Collision Avoidance*, November 3-7, 2013.

⁴ Je n'ai pas eu accès au mémoire mais seulement au rapport technique.

