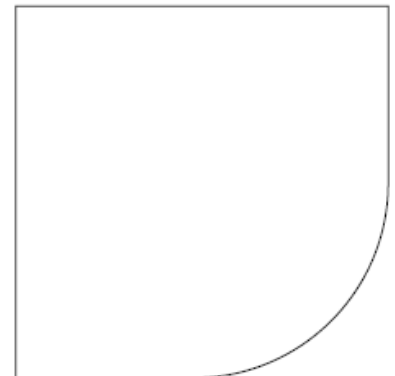


Rapport
d'avancement
CI2019
2017/2018



Projet EchoGuidage

MCHAYMECH Doja, KRAYEM Ibrahim,
HOUEVILLE Alexandre, CORAZZA Alexandre

Table des matières

Table des matières.....	2
Remerciements.....	3
Résumé	4
Abstract	5
Introduction.....	6
I. Suite au projet 3.4.....	7
I.1 Rappel du contexte initial	7
I.2 Point de situation au début de l’UV 4.4.....	7
I.3 Nouvelle gestion du projet.....	8
I.3.1 Planification des tâches.....	8
I.3.2 Organisation des séances	9
I.3.3 Gestion du budget	12
II. Quelques changements dans la partie ingénierie système du simulateur d’anesthésie locorégionale	13
II.1 Architecture fonctionnelle.....	13
II.2 Architecture physique.....	15
III. Description des développements	16
III.1 Localisation de la sonde et de l’aiguille	16
III.1.1 Localisation par traitement d’images	16
III.1.2 Localisation par l’utilisation de transducteurs à ultrasons	20
III.2 Mesure de l’inclinaison de la sonde et de l’aiguille.....	22
III.3 Mesure de la profondeur d’introduction de l’aiguille	23
III.4 Simulation de l’injection du fluide anesthésiant	24
III.5 Centres de traitement des capteurs.....	26
IV. Description des tests.....	27
IV.1 Tests unitaires	27
IV.1.1 Test unitaire des accéléromètres	27
IV.1.2 Test unitaire du capteur de distance IR.....	30
IV.1.3 Test unitaire de la caméra	33
IV.2 Tests d’intégration	33
IV.2.1 Intégration logicielle	34
IV.2.2 Intégration physique	41
IV.3 Tests de validation fonctionnelle	42
V. Conclusion et perspectives	46
Table des figures.....	47
Bibliographie.....	48
Annexe 1 : Mesure de la précision d’un capteur ^[12]	49

Remerciements

Nous tenons à exprimer notre gratitude envers nos professeurs M. Ali Mansour, M. Benoit Clément, M. Luc Jaulin, M. Gilles Le Maillot et M. Yvon Gallou pour leur aide apportée dans les domaines techniques que nous ne maîtrisons pas encore.

Nous souhaitons également adresser nos remerciements à M. Yann Jan, le Dr. Marc Danguy Des Déserts et le Dr. Christophe Vadam pour leur disponibilité et surtout pour nous avoir permis de découvrir le monde médical qui ne nous est pas familier dans notre formation.

Résumé

Aujourd'hui, les moyens de formation pratiques des anesthésistes ne sont pas assez réalistes. Une anesthésie locorégionale (ALR) se déroule en deux étapes. La première est la localisation d'un nerf à l'aide d'une sonde échographique. La seconde est l'introduction de l'aiguille dans la peau suivie de l'injection du produit anesthésiant autour du nerf pour endormir le bloc correspondant.

Le point crucial de l'apprentissage est la coordination des mains et des yeux : en déplaçant la sonde échographique puis en manipulant l'aiguille, l'élève doit uniquement se repérer à l'aide de l'écran. Il existe déjà des simulateurs dans ce domaine, cependant les images affichées à l'écran ne représentent pas ce que l'étudiant rencontrera lors d'une vraie anesthésie. En effet, dans la plupart des cas, l'anesthésiste s'exerce sur un pain de gel et n'observe donc rien de réaliste sur le retour visuel de la sonde.

Le projet EchoGuidage est issu de l'idée de deux médecins de Brest qui souhaitent concevoir un simulateur permettant de préparer efficacement les élèves anesthésistes à une vraie anesthésie locorégionale.

Nous reprenons le projet EchoGuidage 2016/2017 qui a déjà traité de l'acquisition d'une banque d'images échographiques et de la partie dédiée à la localisation de la sonde. Cependant, les calculs lourds pour déterminer cette localisation engendraient un manque de fluidité lors de l'affichage des images. Notre travail a alors consisté à améliorer cette localisation et à représenter l'aiguille et l'injection de produit anesthésiant à l'écran. Nous avons travaillé sur quatre parties indépendantes puis nous les avons reliées entre-elles : la localisation de la sonde et de l'aiguille dans l'espace, leur inclinaison, la profondeur d'introduction de l'aiguille puis l'injection de produit anesthésiant.

Abstract

Nowadays, the practical ways to train anaesthetists are not realistic enough. Actually, local anaesthetic is divided into two steps. The first one consists in finding the interesting nerve thanks to an ultrasound probe. The second one aims at inserting a needle in the patient's skin and injecting an anaesthetizing substance around the nerve so as to anaesthetise all the tissues nearby.

The key point of the traineeship is the coordination between eyes and hands. That is to say: the student has to handle the needle, only by moving the probe and watching the related screen, he should not look at the probe directly. Currently, there are already anaesthetic simulators, however the images displayed by the screen are not realistic enough. Indeed, generally the anaesthetic students can only train by using a real probe on a jelly block with a pipe inside to embody the nerve. The limit of this device is the lack of realism in the images displayed as it does not look like human tissues.

Therefore, the EchoGuidage project born from the need of two physicians of Brest, requests the conception of a realistic simulator in order to enhance the practical training of the anaesthetists as regards local anaesthetic.

We take over the EchoGuidage project already initiated in 2016 so as to improve its features. Over the previous year, the team has already designed a way to acquire an image bank constituted by images of real human tissues. The team has also established a way to locate the false probe on a jelly block. The system was able to display the corresponding image by moving the false probe. However, the complicated calculation to locate the probe, due to the use of cameras was a limit to the flow of the system. Our new objective is to improve the task dedicated to the localisation of the probe and to implement a solution to simulate the insertion of the needle into the skin and the injection of the anaesthetizing substance around the nerve. Thus, we worked according to four tracks related to the needle: the localisation, the inclination, the insertion and the injection.

Introduction

Ce projet est très formateur puisqu'il englobe tous les paramètres variables que nous rencontrerons en tant qu'ingénieur. Nous sommes en contact avec un client qui définit ses exigences. À partir d'une analyse de celles-ci, nous avons pour objectif de les satisfaire par des moyens techniques que nous acquérons grâce à notre formation, à travers des recherches personnelles ou par l'aide de professionnels. Nos solutions devront être réalisables dans la limite d'un budget à respecter. Nos capacités à gérer une équipe sont également suscitées. Enfin, la finalité de nos travaux sera présentée sous la forme d'un démonstrateur avant une date limite.

Notre étude doit mener au développement d'un simulateur de ponction dans le cadre d'une anesthésie locorégionale. Un tel système n'a pas encore été réalisé et le fait de travailler sur un produit inexistant dont le besoin se fait sentir donne une forte valeur à notre projet. Cela démontre que l'ingénierie est un moyen de faciliter le mode de vie des générations présentes et futures et ce dans de multiples domaines.

Nous présentons dans ce rapport, la suite de nos travaux majoritairement axés sur la réalisation du système physique ainsi que sur les différentes vérifications nécessaires à la validation des exigences du système. Après un bref retour sur la première partie du projet, nous nous pencherons sur les différents éléments qui nous ont amené à modifier l'ingénierie système, ensuite nous décrirons les moyens techniques et logiciels utilisés lors de la conception du système. Nous terminerons par une description des différents types de tests effectués sur notre système afin d'en valider le comportement et les attentes.

I. Suite au projet 3.4

I.1 Rappel du contexte initial

Le projet EchoGuidage a pour finalité la réalisation d'un simulateur d'échographie et de ponction pour la formation d'internes anesthésistes. Lors d'une anesthésie locorégionale ^[1], l'anesthésiste repère à l'aide d'une sonde échographique le nerf qui correspond au bloc à endormir. Ensuite, il introduit une aiguille dans le patient pour injecter un produit anesthésiant autour de ce nerf.

Des complications peuvent survenir chez le patient si le produit est injecté dans le nerf. L'objectif principal de notre projet est de donner les moyens aux anesthésistes de s'exercer afin d'éviter une erreur de ce type. Pour cela, le point le plus important à prendre en compte est le suivant : l'extrémité de l'aiguille n'est correctement affichée à l'écran que si elle se trouve dans le plan du faisceau de la sonde échographique. De ce fait, il y a une incertitude sur la connaissance de la position du point d'injection.

L'Interface Homme-Machine du simulateur devra alors afficher l'image échographique correspondant à la position et à l'inclinaison de la sonde. Elle affichera également l'aiguille sur l'écran en fonction de sa profondeur d'introduction puis de son inclinaison et de sa position par rapport à celle de la sonde pour simuler le fait que l'aiguille est bien reproduite à l'écran uniquement lorsqu'elle se trouve dans le plan de la sonde. Pour finir, l'anesthésiste devra avoir un visuel sur l'endroit où il injecte le produit anesthésiant.

En ce qui concerne le matériel factice, la sonde et l'aiguille devront se rapprocher de leur aspect réel. La taille de l'aiguille est une exigence importante : elle doit pouvoir être tenue par 2 doigts et ne devra pas être plus grosse qu'un stylo à quatre couleurs.

I.2 Point de situation au début de l'UV 4.4

Ce projet a été divisé en deux parties bien distinctes, cohérentes et chronologiques. La première partie du projet consistait majoritairement à établir toute l'ingénierie système du produit à concevoir ainsi qu'à établir un état de l'art sur le domaine étudié. Lors de cette première partie, nous avons également commencé à commander et à exploiter quelques capteurs nécessaires à la réalisation du système. A la fin de l'UV 3.4, nous avons ainsi décidé d'obtenir l'inclinaison de la sonde et de l'aiguille à l'aide de deux accéléromètres à trois axes. Nous avons également fait le choix d'un capteur de distance infrarouge afin de simuler l'introduction de l'aiguille dans la peau et d'un bouton poussoir pour simuler l'injection de fluide anesthésiant. Ces capteurs étaient encore en cours d'exploitation à la fin de l'UV 3.4. En ce qui concerne le repérage de la position de l'aiguille et de la sonde dans l'espace, nous hésitions entre l'utilisation de transducteurs piézoélectriques et l'utilisation de caméras avec un traitement d'images. Les capteurs piézoélectriques sont certes plus efficaces lorsqu'il s'agit de traiter les informations qu'ils fournissent contrairement aux caméras, néanmoins, ils sont beaucoup plus compliqués à mettre en place ainsi qu'à exploiter.

La deuxième partie du projet était principalement dédiée à la réalisation physique du système et à la validation du système à l'aide de tests. Au début de l'UV 4.4, il restait beaucoup de tâches à effectuer afin d'obtenir un premier système fonctionnel. Nous avons tout d'abord décidé de terminer l'exploitation des accéléromètres et du capteur de distance infrarouge. Ensuite, l'objectif était de mettre en place les capteurs à ultrasons destinés à la localisation de la sonde et de l'aiguille dans l'espace et de revenir à l'utilisation des caméras pour localiser les objets dans l'espace en cas d'échec lors de la mise en place des transducteurs à ultrasons. Enfin, il était également prévu de concevoir une Interface Homme Machine (IHM) ^[2] afin d'intégrer les différents capteurs sur le système global, puis d'afficher les images échographiques du système en laissant à l'utilisateur la possibilité de choisir différentes options avant et pendant le lancement de la simulation.

I.3 Nouvelle gestion du projet

La première partie du projet nous a permis de construire l'équipe actuelle avec laquelle nous avons réalisé ce projet. Nous avons dû apprendre à travailler tous ensemble et à gérer notre propre organisation. Nous avons également été amenés à distribuer des tâches à chacun d'entre nous afin d'améliorer au mieux la productivité et le confort de chacun. La première partie du projet nous a également permis de nous familiariser avec la procédure d'achat de composants électroniques de l'école. De plus, pour faciliter la gestion du programme Python avec l'équipe, nous avons choisi de travailler avec l'outil collaboratif GitHub. Notre simulateur est disponible à l'adresse <https://github.com/Quiveutdumiel/Echoguidage>.

I.3.1 Planification des tâches

Comme lors de la première partie du projet, nous avons décidé de définir et séparer chaque tâche en fonction de ce que préfère faire chaque membre de l'équipe et en fonction des capacités de chacun.

Pour ce faire, nous avons mis en place un diagramme de Gantt ^[3] pour planifier chaque séance.

Au début, chaque membre de l'équipe a été réparti sur une tâche individuelle consistant à gérer l'exploitation d'un capteur ou l'affichage des images échographiques (Interface Homme Machine). Ensuite, il a été prévu de travailler tous ensemble lors de la phase d'intégration des différents sous-systèmes sur le système final car les différents sous-systèmes sont tous reliés entre eux par le biais de l'Interface Homme Machine. Les tâches associées aux tests d'intégrations ont également été effectuées à plusieurs par groupe de deux constitués de la personne responsable de son capteur et de celle responsable de l'Interface Homme Machine. Nous avons globalement réussi à suivre le diagramme de Gantt établi ci-dessous.

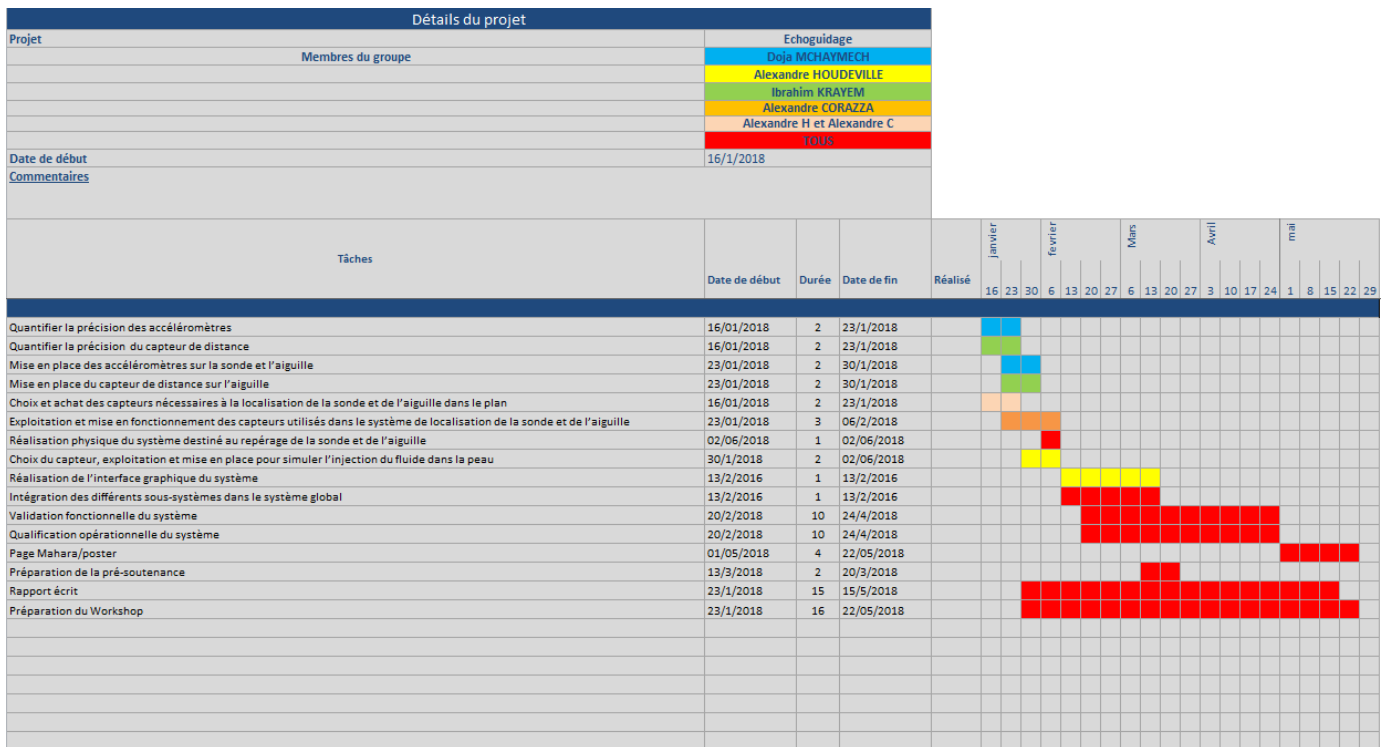


Figure 1- Diagramme de Gantt pour la deuxième phase du projet

I.3.2 Organisation des séances

A cause de divers problèmes rencontrés lors de la mise en accord sur l'heure et la salle de rendez-vous à chaque séance, la productivité et la motivation de l'équipe lors de la première partie du projet n'étaient pas au rendez-vous. Par conséquent, nous avons décidé de mettre en place des solutions afin de parer à ce problème. Tout d'abord, il a été décidé que l'un d'entre nous envoie un mail à tout le monde la veille de chaque séance afin de rappeler l'heure de début de séance (8h30) ainsi que la salle dans laquelle se retrouver. De plus, il a été permis à chaque membre de l'équipe de travailler de son côté l'après-midi lors des journées de projet tant que les tâches attribuées étaient individuelles. Ces mesures ont grandement amélioré la motivation ainsi que la productivité de l'équipe lors de la deuxième partie du projet.

Nous avons également planifié régulièrement plusieurs réunions avec les encadrants du projet ainsi qu'avec les clients du projet : Yann Jan (doctorant anesthésiste), le Dr. Christophe Vadam (anesthésiste) et le Dr. Marc Danguy des Déserts (anesthésiste).

Dates/Informations	Personnes présentes (encadrants et clients)	Objectifs	Appréciation globale	Décisions prises suite à la réunion
Mardi 09 janvier	Ali Mansour	Discussion sur l'utilisation de transducteurs piézoélectriques pour localiser la sonde et l'aiguille	Bonne idée, demande des compétences en modulation que nous possédons mais difficultés et interrogations pour l'utilisation	Commande des transducteurs à ultrasons après dessin du circuit d'exploitation

			du hardware	
Mardi 13 mars	Ali Mansour Yann Jan Marc Danguy des Déserts	Présentation du choix final des capteurs utilisés Présentation de l'interface graphique du simulateur Demande d'informations sur l'affichage de l'aiguille dans le plan de la sonde	Aiguille peu réaliste sur l'interface graphique (détail) L'affichage de l'aiguille n'est toujours pas fonctionnel	Concentration sur l'affichage de l'aiguille
Mardi 10 avril	Ali Mansour Yann Jan Marc Danguy des Déserts Christophe Vadam	Présentation du design final concernant la sonde et l'aiguille Présentation de la partie destinée à l'affichage des images échographiques	Design correcte Problème de liens entre les différentes images affichées : problème repéré au niveau du traitement d'images lors de la détection de la sonde : conflits de couleurs entre l'environnement et la mire colorée placée sur la sonde	Modification de la partie traitement d'image pour repérer la sonde et l'aiguille dans l'espace : changement de couleur et ajout d'une détection de forme et de taille du diamètre de la balle pour filtrer les éléments parasites de l'image Penser à mettre un fond de couleur avec des plaques en face des caméras pour limiter l'influence de l'environnement sur le traitement d'images

Mardi 24 avril	Ali Mansour Yann Jan Marc Danguy des Déserts	Présentation de l'amélioration de la détection de la sonde et de l'aiguille pour repérer leur position dans le plan. Démonstration de l'affichage de l'aiguille sur l'écran.	Problème de récupération des images dans la base de données Problème de luminosité lors du traitement d'image	Intégrer une lampe pour éclairer les mires de couleur (balles)
----------------	---	---	--	--

Figure 2- Tableau récapitulatif des réunions organisées avec les clients et les encadrants du projet

I.3.3 Gestion du budget

Etant donné que nous avons rapidement fait le choix de la plupart des solutions techniques pour chaque sous-système de notre système lors de la première partie du projet, nous avons déjà pu commander un certain nombre de composants électroniques avant de commencer l'UV 4.4. En plus des composants déjà achetés auparavant, nous avons commandé des transducteurs piézoélectriques afin d'expérimenter la solution technique nécessaire à la localisation de la sonde à partir d'ultrasons. Nous avons également racheté un capteur de distance infrarouge car celui que nous avons acheté auparavant ne fonctionnait plus.

Le budget complet du projet :

Composants/Informations	Fournisseur	Prix unitaire euros (TTC)	Quantité	Prix total euros (TTC)
Capteur de distance 10cm 2489 ^[4]	Go Tronic	10,75	2	21,50
L6202 - Driver pont complet DMOS ^[5]	Farnell element14	6,29	1	6,29
MCUSR18A40B12RS - Capteur ultrasonique, Récepteur ^[6]	Farnell element14	4,30	10	43
Module accéléromètre 101020051 ^[7]	Go Tronic	9,00	2	18,00
Ordinateur/clavier/souris/écran	ENSTA	emprunt	1	emprunt
Carte arduino MEGA 2560 ^[8]	ENSTA	emprunt	1	emprunt
Carte Raspberry PI III ^[9]	Reprise de l'année précédente	44,95	1	44,95
Cameras Boulanger HD 720p	Reprises de l'année précédente	22.04	4	88.16
Prix total payé cette année				88,79

Figure 3- Tableau récapitulatif de gestion du budget

II. Quelques changements dans la partie ingénierie système du simulateur d'anesthésie locorégionale

II.1 Architecture fonctionnelle

L'ingénierie système représente une part importante dans la conception d'un système. En effet, elle permet de guider les ingénieurs lors de toutes les phases de fabrication d'un produit par une démarche méthodologique. Cette démarche permet d'organiser la réflexion et le travail des ingénieurs au cours du cycle de vie du produit : de la conception du système à sa fin de vie, tout en satisfaisant le plus possible les besoins du client.

Au cours de plusieurs réunions organisées avec les clients, nous avons modifié certaines exigences avec leur accord afin de rendre plus simple la phase de réalisation du simulateur tout en respectant au mieux les objectifs principaux qui doivent être remplis par le simulateur. Ainsi, nous avons décidé au début de traiter toutes les données et de gérer l'affichage des images échographiques en utilisant uniquement la carte Raspberry PI. Néanmoins, pour des questions de durée de traitement des données excessivement longue liée à l'utilisation des caméras (l'acquisition de nouvelles données provenant des capteurs prenait une dizaine de secondes), nous avons décidé de traiter toutes les données et de gérer l'affichage, directement sur un ordinateur. La modification de certaines exigences notamment concernant la gestion du problème en une dimension pour la sonde et en deux dimensions pour l'aiguille nous ont également permis de simplifier certains calculs nécessaires à l'affichage des images échographiques et de l'aiguille. Par exemple, nous avons décidé d'afficher une image échographique pour chaque pas de 10 degrés d'inclinaison de la sonde pour une position donnée. Néanmoins, les mauvais raccords entre les images de la banque d'images prises pour différents angles d'inclinaison nous ont amenés à simplifier le problème en ne faisant dépendre les images échographiques de fond, uniquement de la position de la sonde échographique selon un axe.

Les fonctions de base du simulateur n'ont pas changé à cause de ces simplifications, nous avons donc conservé l'architecture fonctionnelle déjà définie lors de l'UV 3.4.

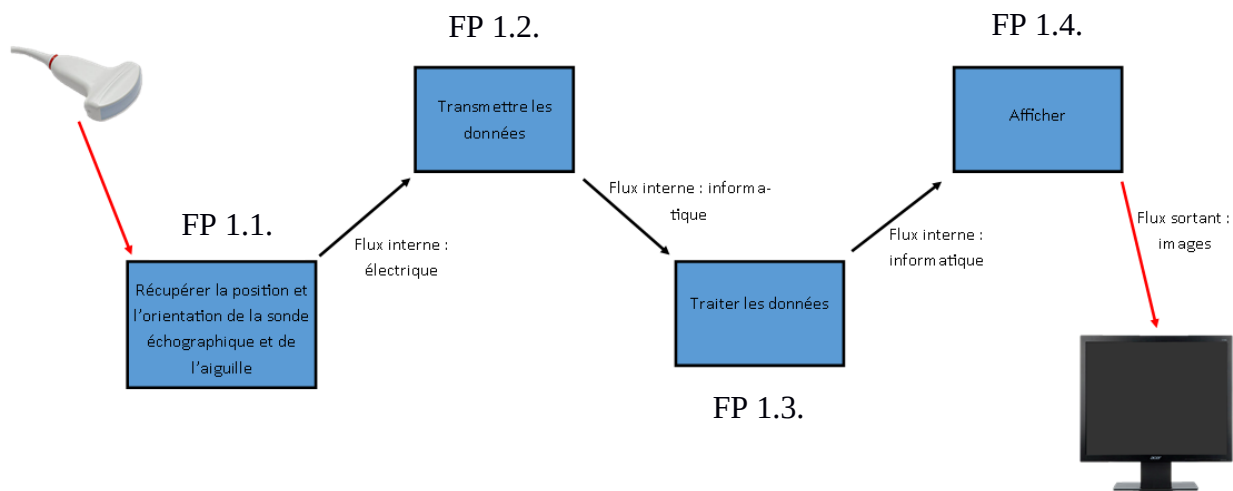


Figure 4- Architecture fonctionnelle du simulateur

Nous avons décrit les fonctions qui aident à comprendre le fonctionnement du système à travers une spécification fonctionnelle.

FP1 : Le système doit permettre à l'utilisateur de pratiquer une simulation d'injection d'anesthésiant locorégionale à l'aide d'un support visuel (écran, projection, ...)

Comportement : Le simulateur est capable de localiser la fausse sonde échographique et l'aiguille.

Donnée : Banque d'images échographiques.

Service : Affiche l'image échographique qui correspond à la position de la sonde et affiche l'aiguille.

FP2 : Le système doit pouvoir interagir avec une banque d'images afin de proposer différents scénarios d'anesthésie à l'utilisateur.

Comportement : Le simulateur propose plusieurs scénarios possibles à l'utilisateur.

Donnée : Banque d'images échographiques.

Service : Démarre scénario choisi par l'utilisateur en prenant en compte son choix à travers l'IHM.

FC1 : Le système doit être assez réaliste pour que l'utilisateur pratique l'anesthésie comme sur un vrai système échographique.

Comportement : Détermine l'intersection entre le plan de la sonde et l'aiguille puis affiche le fluide injecté à l'extrémité de l'aiguille.

Données : Données de localisation acquises par les capteurs.

Service : Affiche l'intersection de l'aiguille avec le plan de la sonde (un segment ou un point) et le fluide injecté.

II.2 Architecture physique

La dernière étape de la méthode APTE consiste à représenter l'architecture physique du système à partir de son architecture fonctionnelle. Cela permet de finaliser la décomposition du système en sous-systèmes et également de choisir une solution technique pour chaque étape dans la chaîne fonctionnelle. Même si nous avons décidé de conserver l'architecture fonctionnelle définie lors de la première partie du projet, la mise en place de nouvelles hypothèses simplificatrices et l'expérimentation de quelques solutions techniques nous ont amené à reconsidérer l'architecture physique de notre système.

Tout d'abord, la possibilité d'utiliser un ordinateur pour traiter les données et gérer l'affichage des images échographiques au lieu d'une carte Raspberry PI conduit à modifier la façon de réaliser la fonction FP 1.3. dans l'architecture définie ci-dessous.

Nous avons également déjà décidé de l'utilisation d'accéléromètres et d'un capteur de distance pour mesurer les inclinaisons de la sonde et de l'aiguille ainsi que la profondeur d'introduction de l'aiguille cependant nous hésitions entre deux solutions pour mesurer la position de la sonde et de l'aiguille. La première solution consistait à utiliser des transducteurs piézoélectriques pour générer et recevoir des ultrasons afin de localiser les objets dans l'espace. Ayant rencontré de nombreux problèmes lors de l'implémentation de cette solution, nous avons décidé de réutiliser la solution déjà mise en place l'année dernière pour localiser la sonde dans l'espace afin de l'appliquer à la localisation de l'aiguille dans le but d'avoir un démonstrateur fonctionnel à la fin de la date de rendu du projet. Cette solution consiste à détecter les objets grâce à des caméras afin d'en déduire leur position dans l'espace.

De plus, nous avons également décidé de simuler l'injection du fluide anesthésiant par l'appui sur un simple bouton poussoir.

Ces décisions nous ont finalement amenés à définir l'architecture physique suivante :

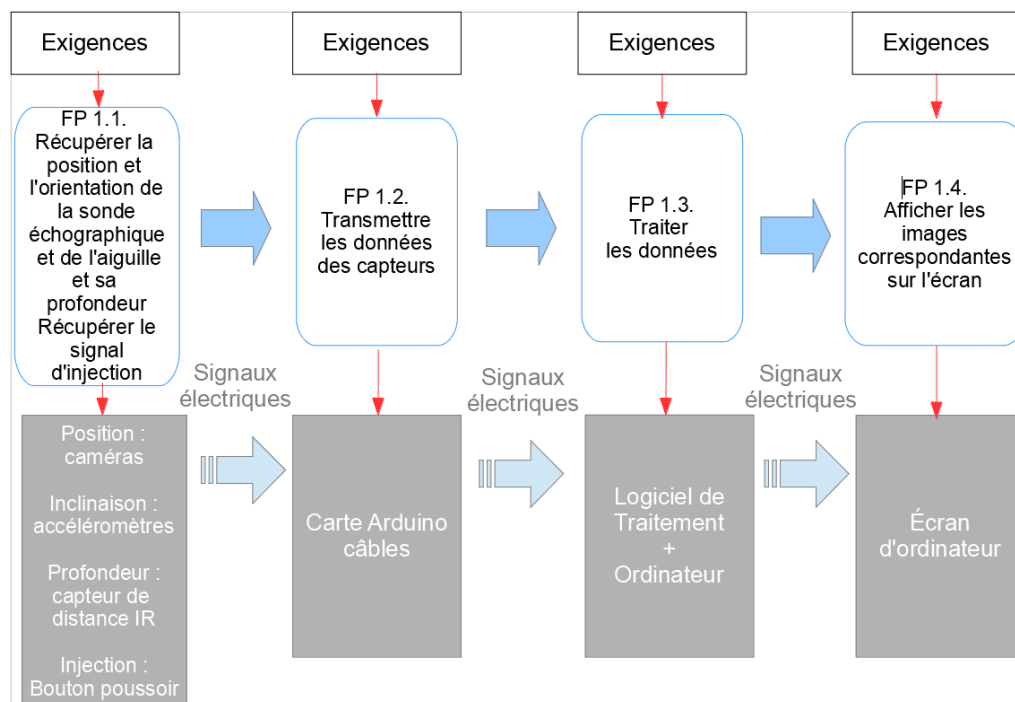


Figure 5- Architecture physique du simulateur

III. Description des développements

L'analyse des différentes fonctions du système nous a permis de le diviser en plusieurs sous-systèmes afin de travailler sur plusieurs parties indépendantes en parallèle. Dès lors, l'organisation de l'équipe s'est mise en place naturellement et correctement puis des responsabilités ont été attribuées à chaque membre du groupe. Ensuite, chacun a réalisé ses propres développements en alliant recherches et expériences, ce qui a enrichi à chacun sa culture technique. Des réunions avec les clients nous ont permis de nous mettre d'accord sur des solutions techniques et de les adapter au fur et à mesure de l'avancée du projet afin de trouver un compromis entre le respect des exigences et la mise en place de solutions techniques réalisables et suffisamment simples pour être implémentées avant la date de fin du projet.

III.1 Localisation de la sonde et de l'aiguille

III.1.1 Localisation par traitement d'images

Afin d'améliorer les performances du simulateur au niveau du temps de calcul et donc au niveau de la fluidité de l'affichage des images échographiques et de l'aiguille, nous avons décidé d'acquérir les coordonnées des objets à l'aide d'une seule caméra au lieu d'effectuer une triangulation ^[10]. Ceci est possible grâce aux fonctionnalités d'OpenCV.

Nous plaçons une balle de marquage jaune sur la sonde échographique. Nous travaillons tout d'abord dans l'espace de couleur HSV puis nous isolons la couleur HSV correspondant au jaune :

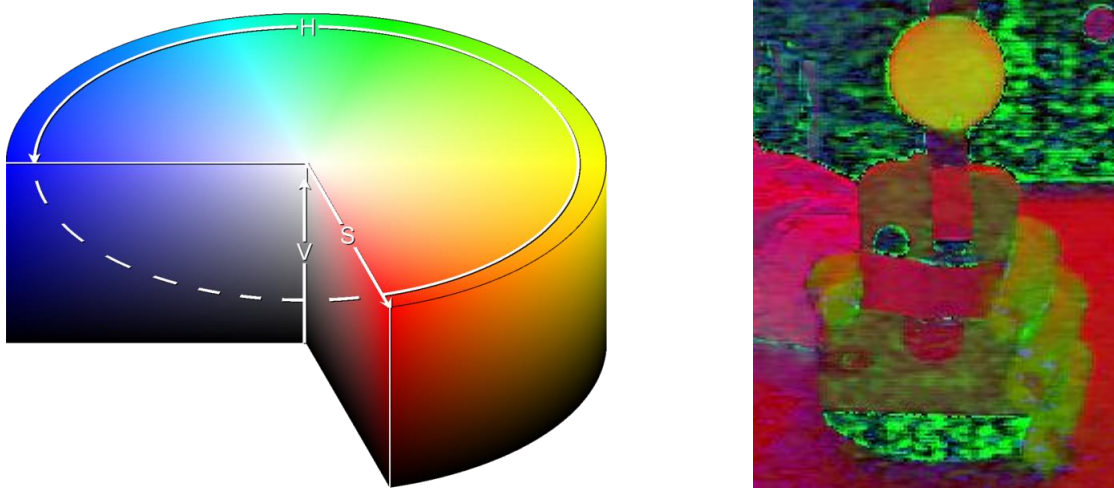


Figure 6- Représentation de l'espace de couleur HSV ^[10]

Le H (Hue en anglais) correspond à la teinte. Elle varie de 0° à 360° (en programmation avec OpenCV, l'angle correspondant est divisé par deux : 0 correspond à 0° et 180 à 360° puisqu'on ne peut pas dépasser 255). Le S pour la saturation va définir si la couleur est foncée ou non (plus elle augmente plus la couleur est foncée). Le V (Value) est la luminosité (plus elle augmente plus la couleur est vive).

Nous avons choisi la plage de couleurs HSV (16, 101, 129) - (26, 255, 255) pour isoler la couleur de la balle jaune sans prendre en compte les objets environnants s'approchant de cette couleur (chaise en bois, bureau, main...).

Nous obtenons le résultat suivant :



Figure 7- Isolement de l'image de la balle par sa couleur

Il est maintenant simple de détecter les contours sur cette image binaire et de l'approximer par un cercle dont la fonction OpenCV nous renvoie les coordonnées du centre et le rayon en pixels. Nous obtenons le résultat suivant :

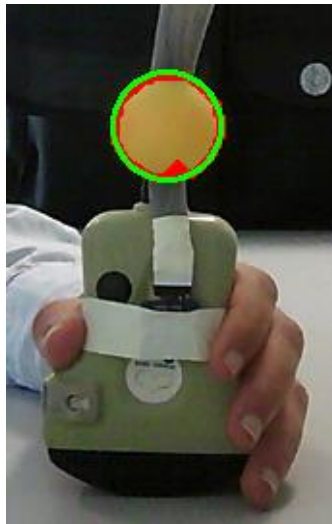


Figure 8- Détection du contour de la balle

En connaissant le diamètre de la balle (4cm), le rayon du cercle en pixels nous permet de mesurer la distance entre la balle et la caméra. Il suffit de mesurer le diamètre de la balle en pixels à une distance donnée pour obtenir une constante : la focale de la caméra F .

$$F = \frac{\text{Diamètre}_{\text{pixels}} \cdot \text{Distance}_{\text{caméra-balle connue}}}{\text{Diamètre}_{\text{centimètres}}}$$

Nous pouvons ensuite en déduire la distance caméra-balle :

$$\text{Distance}_{\text{caméra-balle}} = \frac{\text{Diamètre}_{\text{centimètres}} \cdot F}{\text{Diamètre}_{\text{pixels}}}$$

Nous procédons de même pour la distance entre la caméra et l'aiguille équipée d'une balle bleue :

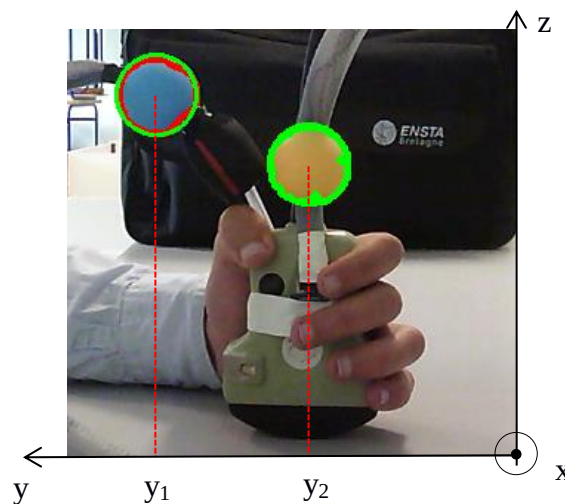
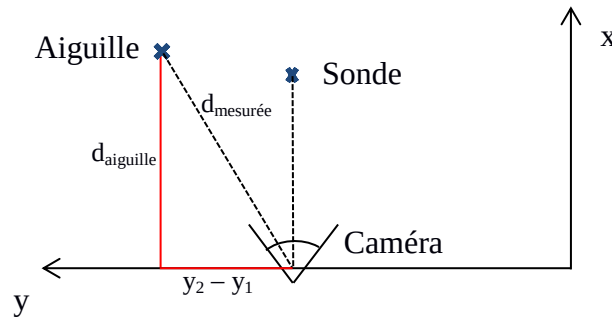


Figure 9- Détection de contours pour la sonde et l'aiguille

Grâce aux coordonnées des centres des cercles fournies par la fonction d'OpenCV *minEnclosingCircle*, il nous est possible de mesurer la distance entre la sonde et l'aiguille selon l'axe y . En effet il suffit de convertir en centimètres (comme précédemment avec la focale) la quantité $y_2 - y_1$. De plus, le signe de cette quantité nous indiquera si l'aiguille est à gauche ou à droite de la sonde, ce qui sera utile pour l'affichage de l'aiguille sur l'écran.

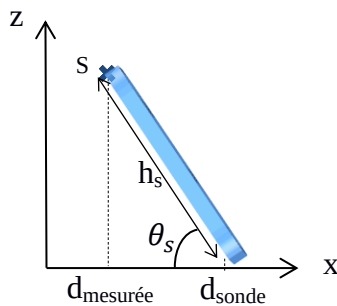
Quelques corrections sont nécessaires pour améliorer notre programme :

En ce qui concerne l'aiguille, la distance mesurée par la caméra n'est pas tout à fait celle souhaitée comme le démontre le schéma ci-dessous. En effet nous souhaitons la distance projetée sur l'axe x. Nous corrigeons cela simplement grâce au théorème de Pythagore.



$$d_{aiguille} = \sqrt{d_{mesurée}^2 - [y_2 - y_1]^2}$$

Pour la sonde, son inclinaison perturbe les mesures de distance. Ayant accès à cet angle, nous sommes en mesure de corriger cette erreur de mesure simplement avec la formule :



$$d_{sonde} = d_{mesurée} - h_s \cos(\theta_s)$$

Ce point pose une convention lors de l'utilisation du simulateur. L'utilisateur doit placer l'accéléromètre de la sonde face à la caméra pour que le signe devant le cosinus ne s'inverse pas.

III.1.2 Localisation par l'utilisation de transducteurs à ultrasons

L'avantage de cette solution est d'éviter de passer par l'utilisation de caméras pour déterminer la position de la sonde et de l'aiguille, car l'utilisation de ces caméras demande beaucoup de ressources de calcul au processeur, ce qui impacte fortement la fluidité de notre système, contrairement à ce que pourrait apporter une localisation par l'utilisation de transducteurs à ultrasons.

Circuit d'émission

L'objectif du circuit d'émission est de générer un signal sinusoïdal ou carré de la fréquence de fonctionnement des transducteurs (39 ± 1 kHz). Le processeur de la carte Raspberry Pi 3 n'étant pas adapté pour générer un signal d'une telle fréquence, nous avons décidé d'utiliser un oscillateur.

La carte Raspberry nous sera donc utile pour générer des signaux carrés d'amplitude 0-5V qui seront multipliés avec le signal de l'oscillateur afin d'alimenter un des transducteurs pendant un certain temps t . Ce paramètre t doit être choisi en fonction du temps de traitement qu'imposent les composants électroniques et devra être supérieur à ce retard afin d'éviter des erreurs. À l'issue de cette multiplication, nous obtenons un signal de 5V de fréquence proche de 39kHz.

L'amplitude de ce signal doit être amplifiée afin d'assurer une puissance d'entrée suffisante pour alimenter les transducteurs (malheureusement la documentation technique du composant n'indique pas la puissance minimale de fonctionnement) ainsi que pour assurer la bonne réception de l'onde acoustique. La valeur minimale devra donc être déterminée par des tests et dépendra de la distance entre les capteurs.

Finalement une fois que le signal sinusoïdal d'amplitude suffisante et de fréquence 39 ± 1 kHz est construit, il reste à l'envoyer tour à tour sur un des 4 émetteurs à l'aide de 4 relais. Les 4 relais commandés par la Raspberry permettront de choisir quel transducteur émettra à un instant donné. Cela permet au récepteur de savoir d'où vient l'onde acoustique sans avoir recours à une fonction d'identification (encodeur/décodeur par exemple).

Circuit de réception

Le circuit de réception se chargera de traiter le signal reçu pour le rendre exploitable par la carte Raspberry. L'objectif est d'amplifier le signal pour que le micro-processeur puisse le détecter et acquérir l'instant de la réception sur le front montant avec un certain seuil.

Anticipations des problèmes potentiels de cette solution

Il est difficile de déterminer le gain des amplificateurs sans pratiquer de tests. Les valeurs des résistances ne devront alors pas être fixées au début, les tests devront être effectués avec des résistances variables.

Les temps de traitement des composants peuvent fausser les mesures. Les composants les plus chronophages dans notre circuit sont les relais: leur temps de transition est de l'ordre de la milliseconde d'après la fiche technique, ce qui n'est pas négligeable à l'échelle électronique. Une calibration est donc nécessaire pour évaluer les erreurs qu'engendrent ces composants.

La tension d'alimentation des transducteurs peut atteindre 160V maximum. Il est donc nécessaire d'utiliser des composants dans cette plage d'alimentation. Par exemple, notre choix de relais (HE700) est adapté puisqu'ils résistent à une tension de plus de 160V.

Le placement des capteurs est également une problématique. Nous avons choisi des transducteurs à cône d'émission large (80°) pour nous permettre de maximiser l'aire où la sonde et l'aiguille peuvent être détectées.

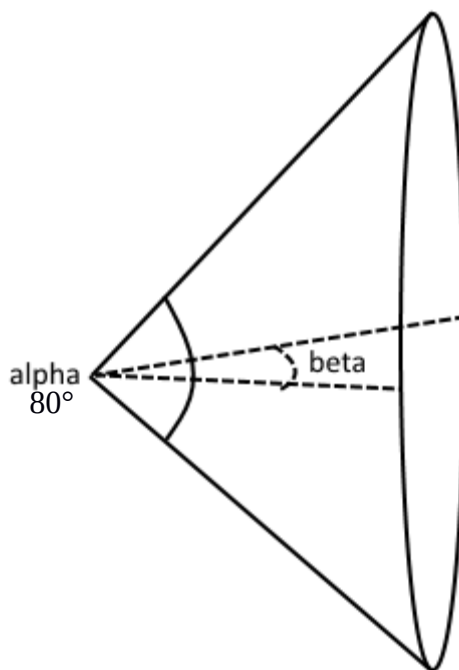


Figure 10- Cône d'émission du transducteur

Le placement de ces capteurs à ultrasons doit permettre aux récepteurs de se retrouver face aux émetteurs. Le plus simple serait de placer 4 récepteurs sur chaque objet pour que chaque récepteur soit face à un émetteur. Un phénomène doit également être pris en compte, celui de la réflexion éventuelle des ondes acoustiques par des objets environnants (ordinateurs, murs...).

A cause de toutes les difficultés d'implémentation liées à cette solution, nous avons décidé d'abandonner cette voie et de nous concentrer sur la solution basée sur le traitement d'images.

III.2 Mesure de l'inclinaison de la sonde et de l'aiguille

Pour mesurer l'inclinaison de la sonde et de l'aiguille, nous avons décidé d'utiliser un accéléromètre à trois axes afin d'obtenir l'inclinaison de ces objets dans l'espace. Du point de vue du design, nous avons fixé l'accéléromètre sur la sonde comme sur l'image ci-dessous. Ensuite nous avons recouvert le composant électronique avec du sparadrap afin qu'il ne soit pas endommagé par l'utilisateur.



Figure 11- Mise en place de l'accéléromètre sur la sonde échographique

Nous avons fixé l'accéléromètre de la même façon sur l'aiguille, en le recouvrant par une couche de mousse afin de faciliter la prise en main de l'aiguille par l'utilisateur tout en évitant la détérioration des capteurs.

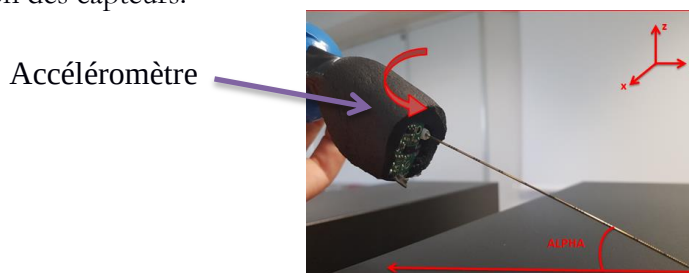


Figure 12- Mise en place de l'accéléromètre sur l'aiguille

Du point de vue logiciel, pour mesurer les différents angles d'inclinaison des accéléromètres (roulis, tangage, lacet), nous avons créé un code en langage C avec le logiciel Arduino pour obtenir l'accélération angulaire associée aux différents angles, ensuite un calcul mathématique a permis de déterminer la valeur des trois angles.

Une fois ces trois angles obtenus, leur valeur est transmise au processeur principal (ordinateur) puis un calcul permet de convertir ces valeurs en un seul angle alpha nécessaire à l'affichage de l'image de l'aiguille sur l'écran comme indiqué sur la figure 12.

$ALPHA = \text{atan2}(\text{angleX}, \text{angleY}) * 57.3$ où angleX et angleY sont les angles correspondant à l'axe X et l'axe Y de l'accéléromètre.

III.3 Mesure de la profondeur d'introduction de l'aiguille

La mesure de la profondeur d'introduction de l'aiguille a pour objectif de simuler l'introduction de l'aiguille dans un corps humain. Pour ce faire, nous avons utilisé un capteur de distance infrarouge. La mesure de distance est ainsi basée sur la méthode Time-Of-Flight qui permet de mesurer précisément les distances grâce à des impulsions infrarouges avec une résolution de 1mm. Le principe de fonctionnement du capteur est le suivant : le capteur émet une onde infrarouge qui se réfléchit sur une surface située à une certaine distance et reçoit l'onde réfléchie à l'aide d'une photodiode. Ensuite, le capteur mesure la durée entre l'émission de l'onde et sa réception, ce qui permet d'en déduire la distance entre la cellule du capteur et l'objet.

Pour que le capteur mesure correctement la profondeur d'introduction de l'aiguille, il doit fonctionner dans les conditions suivantes :

- La direction d'émission de l'onde infrarouge doit être parallèle à l'aiguille et située sur la côté de l'aiguille, du côté intérieur de l'angle entre l'aiguille et la surface d'introduction, afin de ne pas générer d'erreurs de distance dues au placement de la cellule du capteur lorsque l'on tourne l'aiguille. Cela implique la contrainte suivante : l'aiguille ne peut se tenir que d'une certaine façon. Afin de faciliter la prise en main de l'aiguille de cette façon, des bandes rouges ont été fixées sur l'aiguille afin de guider l'utilisateur.

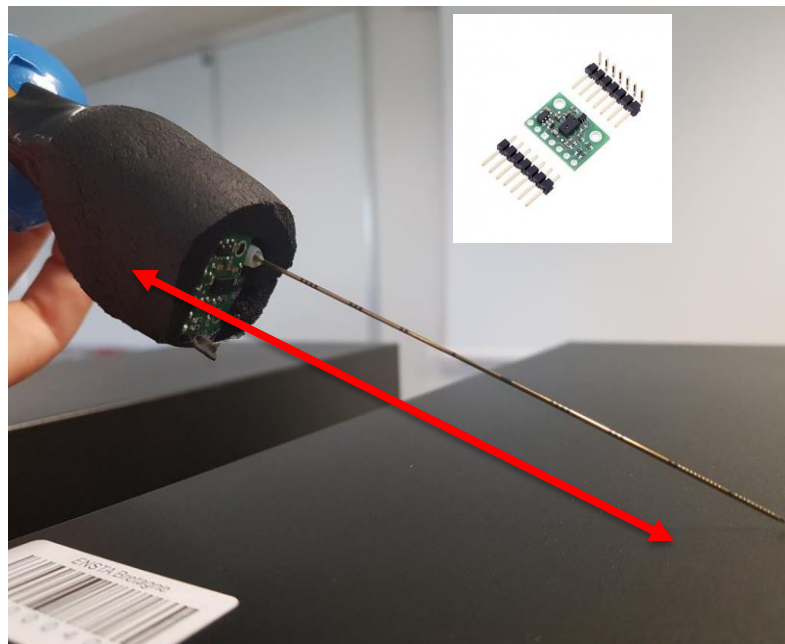


Figure 13 - Mise en place du capteur sur l'aiguille et prise en main de l'aiguille par l'utilisateur

- Le capteur doit être fixé sur le côté de l'aiguille.
- Pour plus de précision lors du traitement des données, nous enregistrons plusieurs valeurs à la suite renvoyées par le capteur afin de calculer une valeur moyenne et d'en déduire une mesure plus précise de la distance à mesurer.

Par exemple pour les valeurs ci-dessous, on en déduit une valeur plus précise de la distance à renvoyer :

t0	t1	t2	t3	t4	t5		Moyenne
88	89	88	87	88	89	➔	88

La distance renvoyée par le capteur correspond en fait à la distance entre le capteur et la surface de réflexion de l'onde. Pour mesurer la réelle distance d'introduction de l'aiguille, il suffit de soustraire la distance mesurée par le capteur par la longueur de l'aiguille (distance entre le point d'émission de l'onde du capteur et le bout de l'aiguille).

La figure 13 présente le montage du capteur de distance sur l'aiguille afin d'assurer le confort de l'utilisateur et la non détérioration des capteurs. Le capteur de distance est fixé juste en dessous de la couche de mousse. Cela permet à l'utilisateur de ne pas interférer dans l'émission du flux lumineux, puisque ce dernier devra tenir l'aiguille uniquement au niveau de la couche de mousse située au-dessus du capteur de distance. La double flèche rouge représente la distance calculée par le capteur de distance.

III.4 Simulation de l'injection du fluide anesthésiant

La simulation de l'injection du fluide anesthésiant a pour objectif de permettre à l'utilisateur de savoir où se trouve le bout réel de l'aiguille par lequel sera injecté le fluide anesthésiant lorsqu'il déclenchera l'action correspondante.

Pour mettre en place cette fonctionnalité, nous avons décidé d'utiliser un simple bouton poussoir.



Figure 14- Bouton poussoir utilisé pour simuler l'injection de fluide anesthésiant

Le but recherché est le suivant : lorsque l'utilisateur presse le bouton poussoir, il faut qu'un point rouge s'affiche sur l'image échographique à l'endroit où se trouve le bout de l'aiguille, pas forcément visible sur l'image échographique, comme lors des situations réelles. En effet, lors d'une vraie anesthésie locorégionale, l'aiguille n'est pas toujours affichée entièrement sur l'image échographique selon la façon avec laquelle elle coupe le plan de la sonde comme on peut le voir sur les figures ci-dessous.

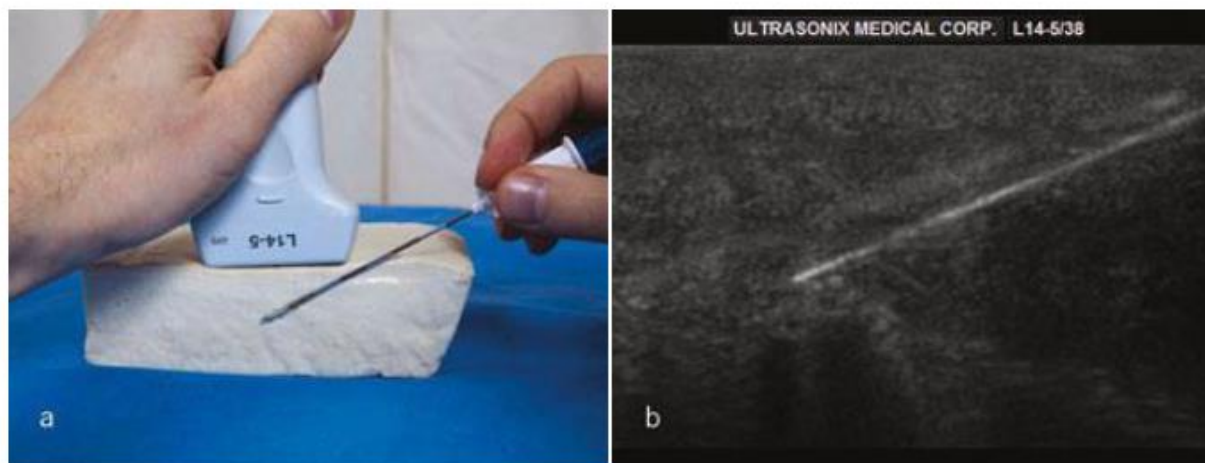


Figure 15- Introduction avec orientation de l'aiguille dans le plan du faisceau de la sonde

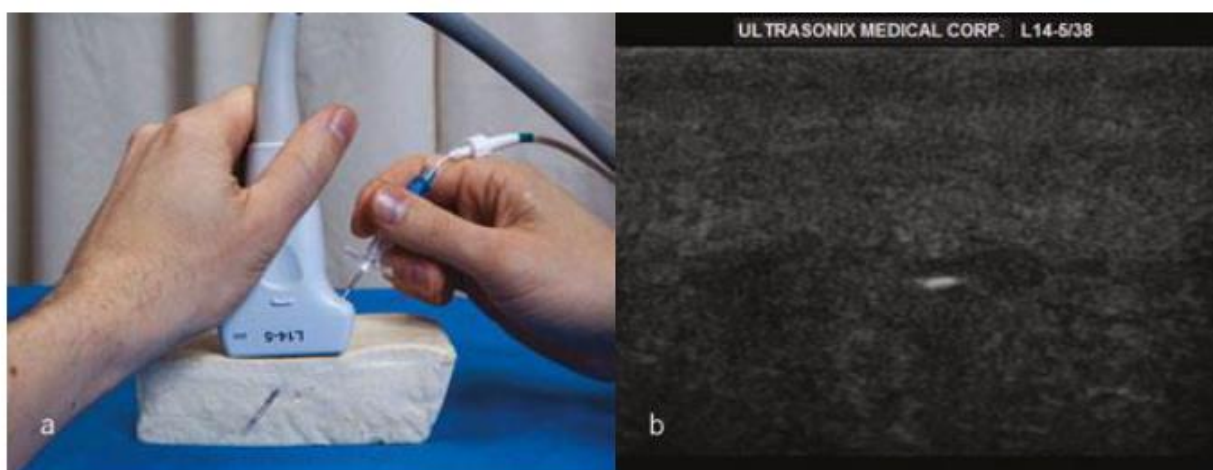


Figure 16- Introduction avec orientation de l'aiguille dans le plan perpendiculaire au faisceau de la sonde

Le bouton est relié à la carte Arduino à l'aide d'un câble électrique. Pour des questions de réalisme, il a été placé sur le piston de la seringue reliée à l'aiguille, censée contenir le fluide anesthésiant.



Figure 17- Mise en place du bouton poussoir sur le système

III.5 Centres de traitement des capteurs

Pour traiter les données des différents capteurs utilisés dans le simulateur, nous avons décidé de séparer le système sur deux centres de calcul différents : le premier centre de calcul a pour objectif de traiter les données des capteurs puis de renvoyer des données brutes (floats) exploitables pour le deuxième centre de calcul. Le deuxième centre de calcul a pour objectif d'exploiter ces données brutes afin de déterminer les caractéristiques des différentes images à afficher au niveau de l'interface graphique. Le deuxième centre de calcul permet également de démarrer le programme principal du simulateur (main) en générant l'Interface Homme Machine (IHM) du système.

Pour exploiter les données des différents capteurs, nous avons décidé d'utiliser une carte Arduino MEGA 2560. Nous avons choisi cette carte malgré le fait que son micro-processeur soit beaucoup moins puissant que celui d'une carte Raspberry pi car nos différents capteurs à exploiter sont très peu consommateurs de ressources (on ne parle pas ici des caméras). Cette carte électronique est donc suffisante pour recevoir les données des deux accéléromètres ainsi que celles du bouton poussoir et du capteur de distance IR sans aucune latence visible.

Pour utiliser les différentes données centralisées par la carte Arduino afin de produire l'intégralité de l'Interface Homme Machine et exploiter le flux vidéo retourné par les caméras nécessaires à la localisation de l'aiguille et de la sonde, nous avons tout d'abord décidé d'utiliser une carte Raspberry PI III. En effet, contrairement à la carte Arduino Mega, cette carte permet de programmer en utilisant le langage de programmation python, ce qui a permis de gagner du temps lors de la réalisation de l'Interface Homme Machine et de la partie liée au traitement d'image car nous maîtrisons mieux ce langage lorsqu'il s'agit de faire des tâches plus complexes, notamment concernant la Programmation Orientée Objet. De plus, cette carte électronique fonctionnant presque exactement comme un ordinateur classique avec le système d'exploitation linux, permet de programmer directement sur la carte sans téléverser le code à chaque modification comme c'est le cas sur la carte Arduino. Cette fonctionnalité nous a permis de faire rapidement des tests, de faciliter l'affichage de l'interface graphique et de relier facilement la carte Arduino à la carte Raspberry comme s'il s'agissait d'un ordinateur classique (simple lien à faire entre la carte Arduino et son port de branchement sur la carte dans le script principal). Enfin, la carte Raspberry PI III est la carte Raspberry avec le processeur le plus puissant actuellement (fréquence du processeur : 1,4 GHz quadricœur contrairement à 16 MHz mono-cœur pour la carte Arduino MEGA 2560 : plus la fréquence du processeur est grande et plus celui-ci pourra faire d'opérations par seconde, un processeur multi-cœur permet d'effectuer plusieurs opérations en même temps, il est donc plus efficace qu'un processeur mono-cœur). Cette dernière caractéristique est très importante car la génération de l'affichage ainsi que l'exploitation des données fournies par la carte Arduino et les caméras demandent beaucoup de ressources en calcul.

D'ailleurs, les premiers tests d'exploitation du flux vidéo à partir de la carte Raspberry ont très vite révélés les limites de notre carte. En effet, nous avons pu constater par des tests que la gestion du flux vidéo ralentissait fortement l'exploitation des données provenant des autres capteurs fournies par la carte Arduino (délai de plus de 5 secondes pour recevoir la mise à jour des valeurs des différents capteurs). Lors d'une réunion avec les clients, nous avons fait part de notre problème car la fluidité de l'affichage constitue une fonction contrainte majeure de notre simulateur. Nous avons peur qu'en utilisant un ordinateur comme ceux présents à l'école, la contrainte liée à la taille du système ne serait plus respectée or, la taille d'un vrai système échographique est plus grande que celle d'un ordinateur de bureau. Il nous a donc été autorisé d'utiliser un ordinateur de bureau ou même un ordinateur portable pour traiter les données transmises par la carte Arduino ainsi que pour générer l'Interface Homme Machine du système. Lors des tests de traitement d'image avec les ordinateurs de l'école, le problème de latence a complètement disparu ce qui nous a permis de confirmer l'utilisation d'un ordinateur comme centre de calcul pour la deuxième étape d'exploitation des données des capteurs.

IV. Description des tests

Pour concevoir notre système, il est nécessaire de rappeler que nous nous sommes appuyés sur un cycle en V dans le cadre de l'ingénierie système de notre système. Le suivi de ce cycle en V consistait à décomposer le système en sous-systèmes puis chaque sous-système en composant. L'objectif de cette décomposition permet de développer chaque fonctionnalité du système global de façon indépendante des autres afin d'avancer plus rapidement sur le projet et surtout afin de pouvoir effectuer des tests à différentes échelles sur le système afin de vérifier au mieux qu'il possède les bonnes fonctionnalités. Ces tests permettent également de remettre en question la partie ingénierie système du simulateur, en effet, lors de la conception d'un produit, il existe toujours des parties pour lesquelles il est impossible de prévoir des résultats précis avant d'avoir commencé la réalisation. Ces phases de tests nous ont alors permis de trouver des solutions alternatives lorsque nous avons rencontrés des problèmes imprévus lors de la phase réalisation.

Ingénierie Système : Cycle en V

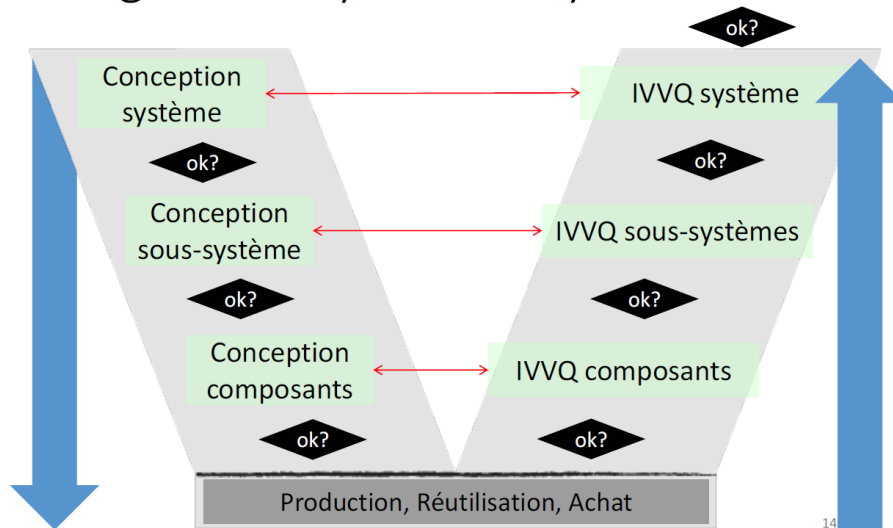


Figure 18- Le cycle en V en ingénierie système

Les différents tests ont été effectués à différentes échelles. Les tests unitaires nous ont permis tout d'abord de tester les fonctionnalités brutes pour chaque composant utilisé. Les tests d'intégration nous ont permis de relier les composants entre eux et de vérifier le fonctionnement du lien entre tous les composants du système et de l'affichage des images échographiques au travers de l'Interface Homme Machine. Enfin, les tests de validation fonctionnelle ont constitué la dernière étape du processus de validation du système. Ils nous ont permis de vérifier que le système final conçu satisfait bien les exigences définies au début du projet.

IV.1 Tests unitaires

IV.1.1 Test unitaire des accéléromètres

Pour tester le bon fonctionnement des accéléromètres dans le cadre de la fonction qui leur a été associée, c'est-à-dire renvoyer l'angle d'inclinaison de la sonde ou de l'aiguille par rapport au pain de gel, nous avons mis en place le protocole suivant.

Nous avons tout d'abord fixé un angle d'inclinaison arbitrairement, ici il s'agit d'un angle à 90° mesuré par un rapporteur. Ensuite, nous avons mesuré 20 fois cet angle avec l'accéléromètre et le programme convertissant les accélérations renvoyées par l'accéléromètre en angle.



Figure 19- Mesure de l'angle avec l'accéléromètre

Nous avons obtenu le tableau suivant où des explications sont données en annexe 1 à propos du calcul de la moyenne, de l'écart type, de l'erreur de fidélité, de l'erreur de justesse et de la précision :

	zAngle en degré
1	88.81
2	88.81
3	88.67
4	88.59
5	88.56
6	88.53
7	89.05
8	88.88
9	89.29
10	89.05
11	89.41
12	89.13
13	88.93
14	88.8
15	88.7
16	88.63
17	88.59
18	88.55
19	88.53
20	89.41

m	88.846
σ	0.29
e_{fidel}	0.29
e_{just}	1.154
e_{prec}	1.18

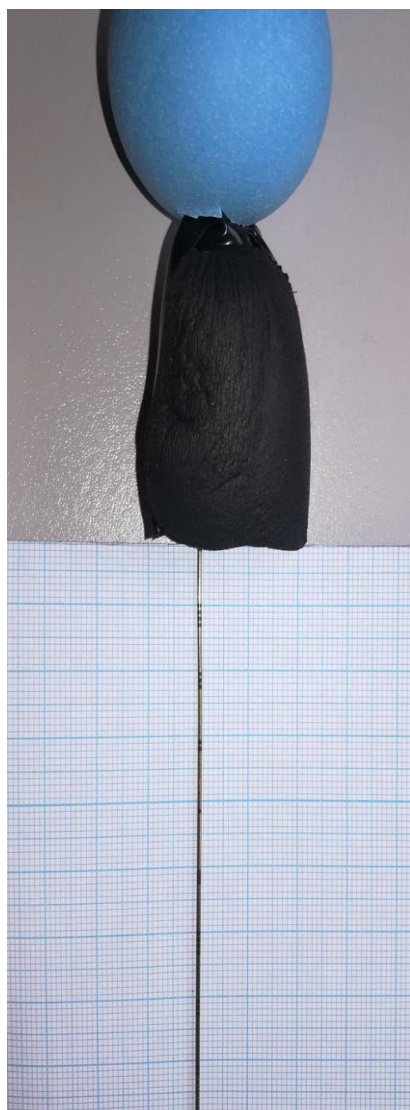
Figure 20- Tableau de test unitaire pour l'accéléromètre

La moyenne, l'écart type, l'erreur de fidélité et l'erreur de justesse ont été calculés pour cet angle afin d'établir la précision de notre capteur, ici pour un angle mesuré de 90° . A partir de ces calculs, nous pouvons en déduire que le capteur mesure un angle de 90° avec une précision de plus ou moins 1.18 degrés.

IV.1.2 Test unitaire du capteur de distance IR

Pour tester le bon fonctionnement du capteur de distance Infra Rouge destiné à simuler la profondeur d'introduction de l'aiguille dans un corps, nous avons mis en place le protocole suivant.

Tout d'abord, nous avons déterminé la précision du capteur de façon similaire à celle de l'accéléromètre. Pour ce faire, nous avons fixé une distance égale à 88 mm entre le point d'émission de la lumière du capteur et une surface plane de façon à ce que le faisceau Infra Rouge arrive de manière perpendiculaire à la surface, puis nous avons mesuré cette distance 20 fois avec le capteur Infra Rouge et enfin nous en avons déduit la précision du capteur pour un angle d'inclinaison de l'aiguille tel que celle-ci soit perpendiculaire à la surface considérée et pour une distance de 88 mm.



	Distance
1	86
2	87
3	87
4	87
5	87
6	87
7	87
8	88
9	87
10	87
11	87
12	88
13	88
14	86
15	87
16	86
17	87
18	86
19	88
20	88
m	87.05
σ	0.68633274
e_{fidel}	0.68633274
e_{just}	0.95
e_{prec}	1.17198662

Figure 21- Tableau de test unitaire pour le capteur de distance IR

Nous avons également remarqué qu'en inclinant l'aiguille par rapport à la surface d'introduction, la distance mesurée par le capteur était légèrement faussée en fonction de l'angle d'inclinaison. Nous avons décidé de mesurer l'erreur sur la distance mesurée en fonction de l'angle d'inclinaison entre le faisceau du capteur et la surface d'introduction de l'aiguille. Nous avons obtenu le graphique suivant en fixant la distance vraie à 100 mm.

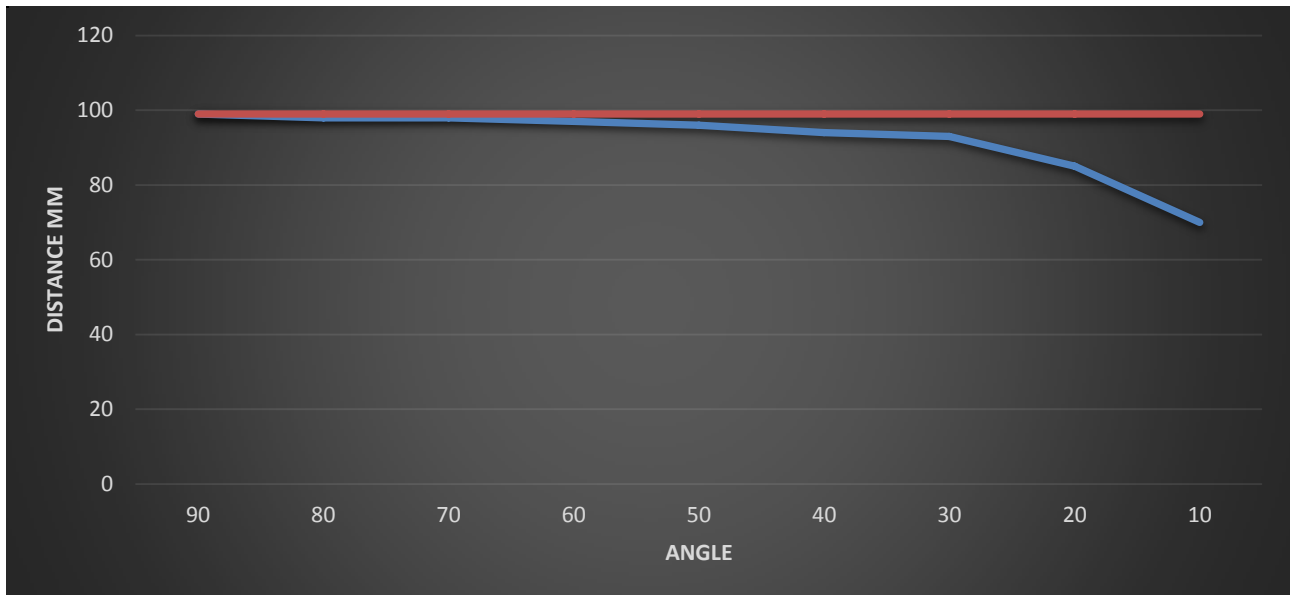


Figure 22- Distance vraie (en rouge) et distance mesurée (en bleu) en fonction de l'angle d'inclinaison de l'aiguille en degrés

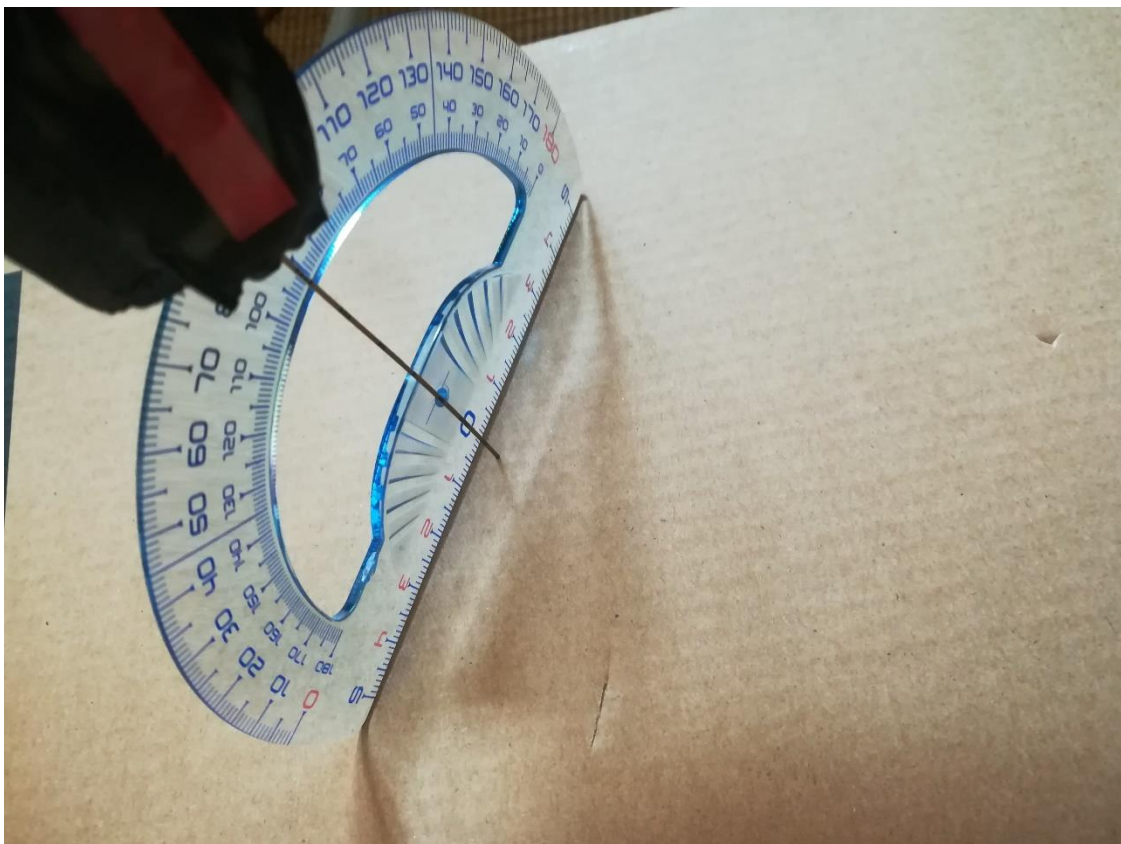


Figure 23- Mesure de la distance par le capteur IR en fonction de l'angle d'inclinaison de l'aiguille

D'après les clients de notre système, l'angle d'inclinaison de l'aiguille peut varier entre 90° et 30° pour un côté de l'aiguille lors d'une utilisation normale du système. De ce fait, nous avons estimé que pour notre système, il ne fallait pas que l'erreur maximale de mesure de distance par le capteur de distance IR ne devait dépasser un centimètre là où l'erreur est maximale donc pour un angle de 30° . D'après le graphe de la figure 23, pour cet angle de 30° , la distance mesurée vaut 93 mm ce qui correspond à une erreur de mesure de 7 mm ce qui est acceptable d'après le critère que nous avons fixé.

Enfin, en testant l'introduction de l'aiguille directement sur le pain de gel, nous nous sommes aperçus que le pain de gel était considéré par le capteur comme une zone transparente, nous avons donc décidé de recouvrir le pain de gel par du papier afin de parer à ce problème.

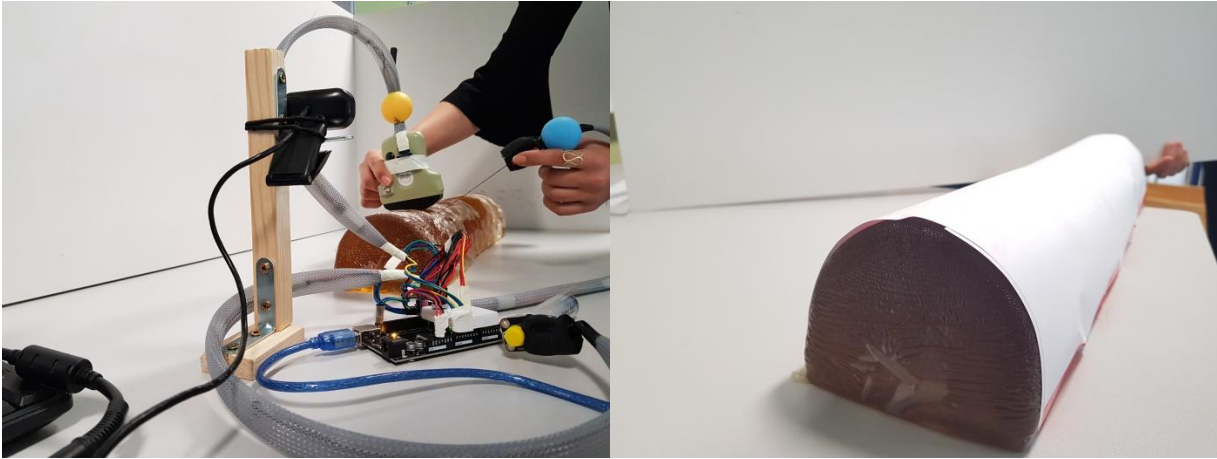


Figure 24- Pain de gel avant et après couverture en papier

IV.1.3 Test unitaire de la caméra

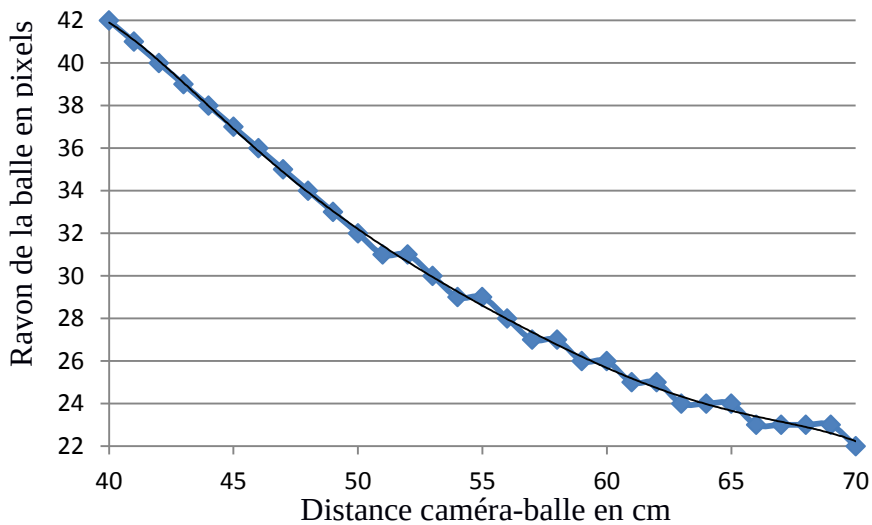


Figure 25- Rayon de la balle sur l'image produite par la caméra en fonction de la distance entre la balle et la caméra



Figure 26- Test unitaire du système balle/caméra

Lors des campagnes de mesures de distance entre la caméra et une balle de marquage, nous avons constaté que plus la distance est importante, moins le programme fait de distinctions entre les différents rayons de la balle en pixels. En effet, nous remarquons sur la plage 40-50cm que la valeur du rayon de la balle en pixels à l'écran est mise à jour tous les centimètres alors qu'entre 60 et 70cm son rayon ne prend que 3 valeurs différentes (cf graphe).

Nous souhaitons maximiser le nombre d'images affichées par centimètre lors du déplacement de la sonde afin d'améliorer la résolution spatiale pour l'affichage des images échographiques et d'éviter d'obtenir un affichage saccadé. Nous avons donc tenu compte de ces observations pour choisir l'origine du repère sur l'axe de translation de la sonde. Une origine à 40cm de la caméra assure une meilleure fluidité et une bonne visibilité de la balle de marquage de l'aiguille. En effet, si l'origine est placée à 30cm, la boule de marquage bleue de l'aiguille sort du champ de vision de la caméra et n'est plus détectée. L'origine est donc fixée à 40cm de la caméra.

IV.2 Tests d'intégration

La phase de tests d'intégration permet de mettre en relation les différentes parties du système et de vérifier le bon fonctionnement du sous-système ainsi formé. Cette étape a principalement consisté à concevoir et à adapter l'Interface Homme Machine du système en fonction des différents capteurs utilisés afin d'afficher correctement les images échographiques du système, de façon efficace et confortable pour l'utilisateur.

IV.2.1 Intégration logicielle

Pour intégrer les différents capteurs au système, nous les avons reliés entre eux dans l'Interface Homme Machine par l'intermédiaire de la Programmation Orientée Objet. En effet, nous avons créé un objet « echogra » de classe « echographie » pour simuler la sonde échographique dans le logiciel en la caractérisant par ses différents paramètres physiques : sa position sur un plan obtenue à partir du traitement du flux vidéo issu de la caméra et son inclinaison obtenue à partir des angles de l'accéléromètre. Nous avons également créé un objet « aigu » de classe « aiguille » afin de simuler l'aiguille caractérisée par sa position sur le plan également obtenue avec la même caméra nécessaire à la localisation de la sonde, son inclinaison obtenue à partir des angles de l'accéléromètre, sa profondeur d'introduction dans le pain de gel obtenue à partir du capteur de distance IR et sa capacité à injecter du fluide anesthésiant modélisée par un bouton poussoir.

Deux objets « fenêtre » ont également été créés pour l'affichage graphique.

A ces différents objets ont été associées divers fonctions. Les fonctions associées à l'objet « echogra » ont pour finalité d'afficher l'image échographique en fonction de la position de la sonde et de fournir à l'aiguille les informations nécessaires à son affichage sur l'image. Les fonctions associées à l'objet « aigu » ont pour objectif d'afficher l'aiguille en fonction de ses différents paramètres en superposant un trait gris symbolique à l'image échographique affichée. Elles permettent également d'afficher le point auquel sera injecté le fluide anesthésiant lorsque l'on appuiera sur le bouton.

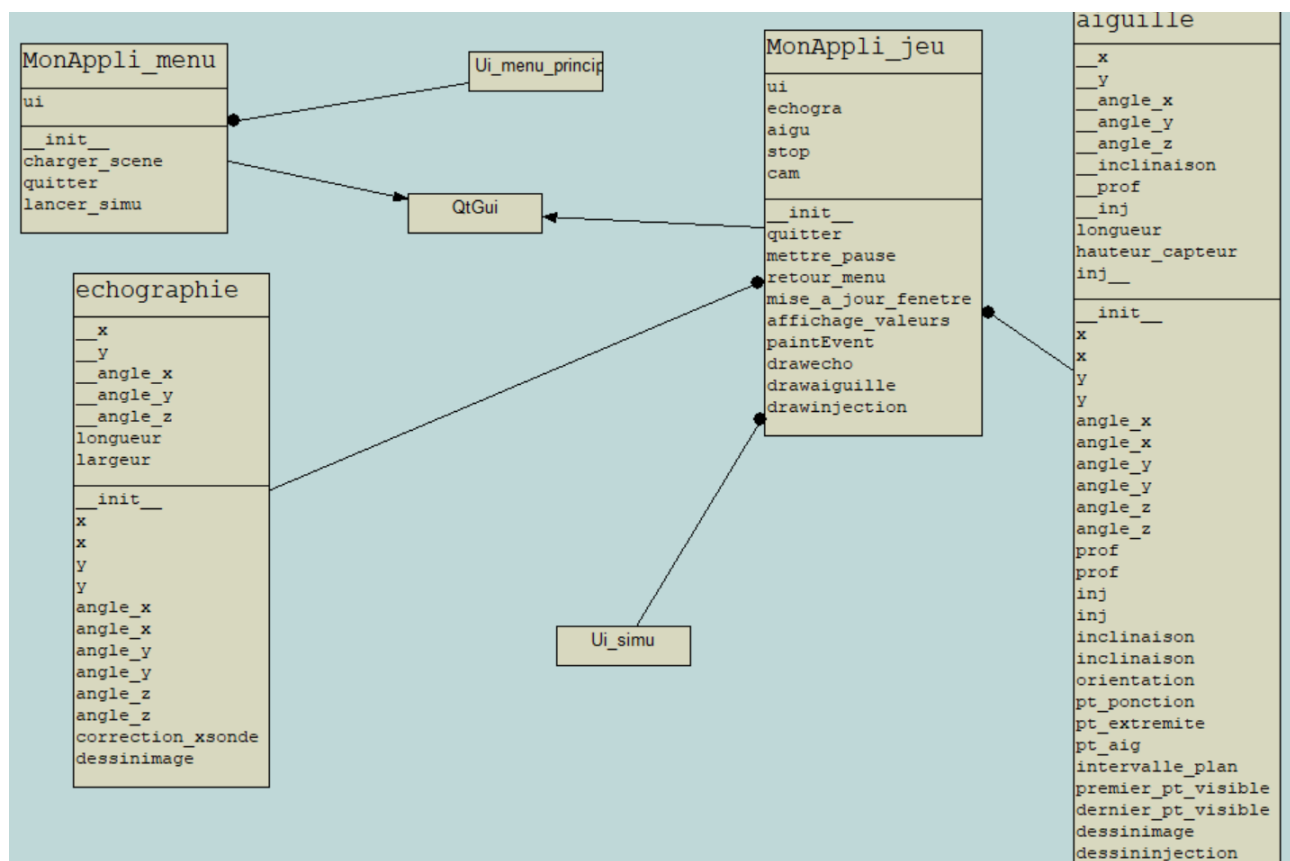


Figure 27- Diagramme de classes du logiciel générant l'Interface Homme Machine

La fenêtre « Menu » permet à l'utilisateur de choisir le scénario qu'il souhaitera aborder lors de la simulation. Le bouton simulation permettra de démarrer la simulation en ouvrant une fenêtre « Simulation » après avoir chargé toutes les images qui seront nécessaires au scénario.

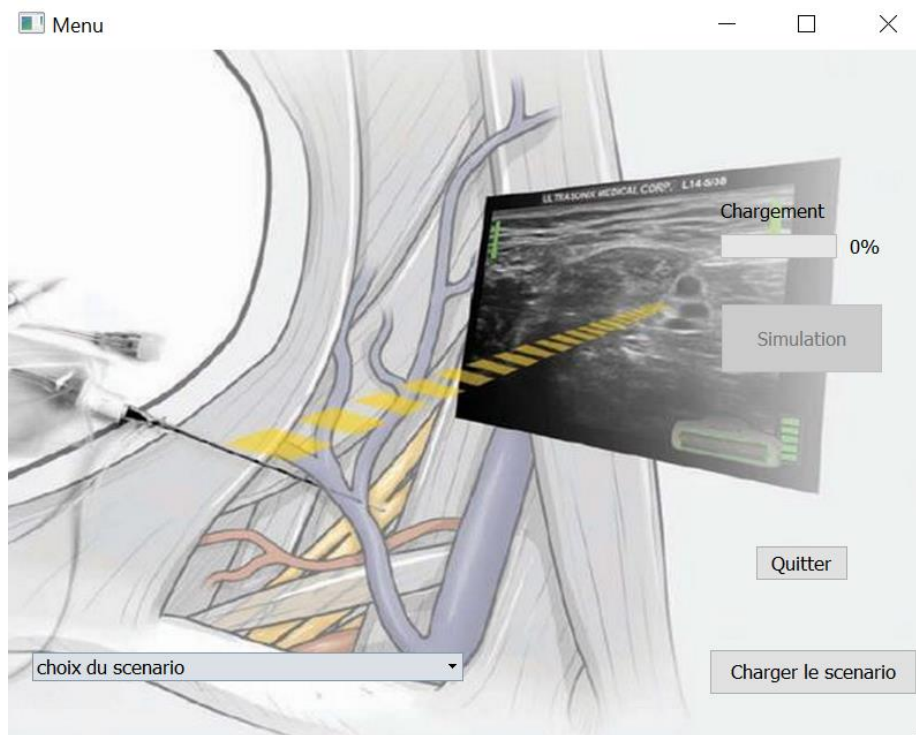


Figure 28- Affichage du menu de l'application

La fenêtre « Simulation » permet d'afficher les images échographiques en temps réel. Depuis cette fenêtre, il est possible de mettre la simulation en pause (images rendues statiques) et également de revenir à la fenêtre « Menu ».

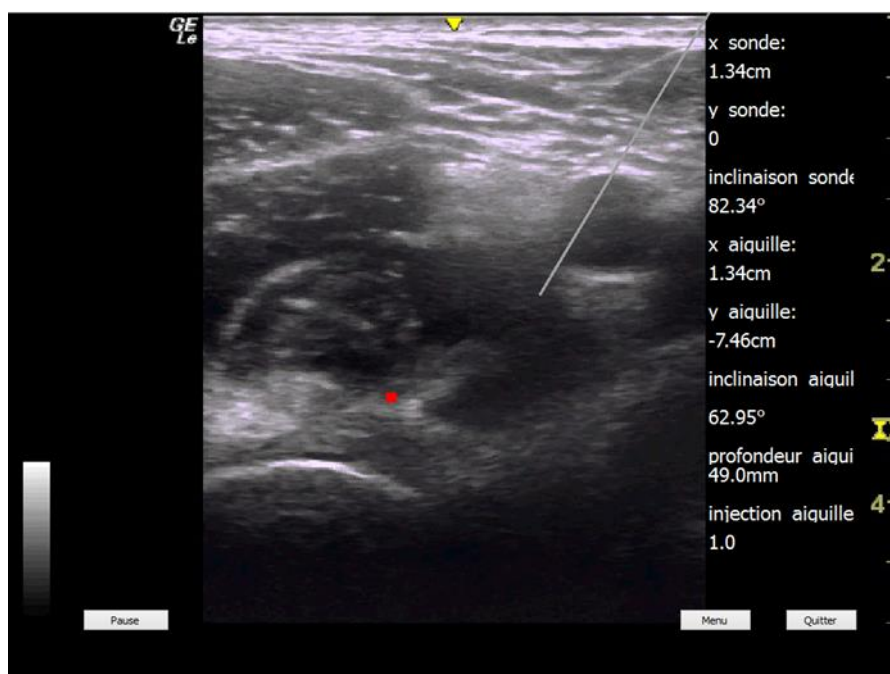


Figure 29- Affichage de la partie simulation de l'application

Afin de réaliser les différents tests pour vérifier la cohérence entre les images affichées et les valeurs données par les capteurs, nous avons affiché les valeurs provenant des différents capteurs sur l'écran puis nous les avons comparées avec les images échographiques affichées à l'écran.

Affichage de l'aiguille sur l'IHM

Dans cette partie, nous allons déterminer les conditions d'affichage de l'aiguille et donc les conditions mathématiques et numériques de l'appartenance de l'aiguille au plan de la sonde. L'aiguille s'affiche selon sa position, son inclinaison, son orientation relative par rapport à la sonde, sa profondeur d'introduction dans le pain de gel, la position et l'inclinaison de la sonde. Finalement, seule l'intersection de l'aiguille avec le plan de la sonde est affichée à l'écran. Dès lors, plusieurs affichages doivent être générés : un point, l'aiguille complète ou partielle avec son extrémité cachée. Dans tous les cas, le liquide simulé doit se manifester à l'extrémité de l'aiguille, même si cette partie de l'aiguille n'est pas visible à l'écran. Cela permet à l'anesthésiste de vérifier s'il n'a pas fait d'erreur lors de l'exercice. L'objectif est d'afficher les points de l'aiguille situés dans le plan de la sonde vu de face comme illustré ci-dessous.

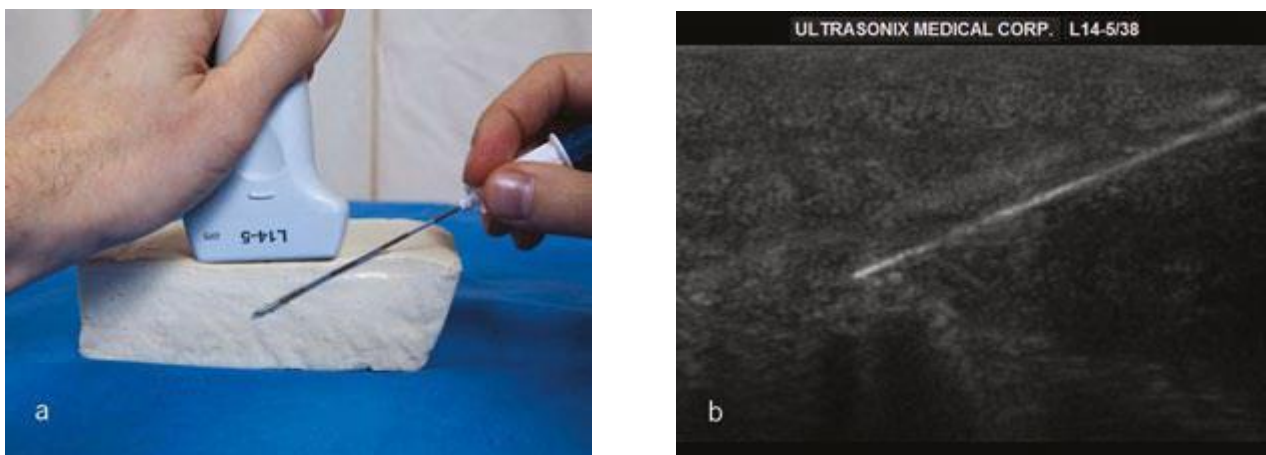
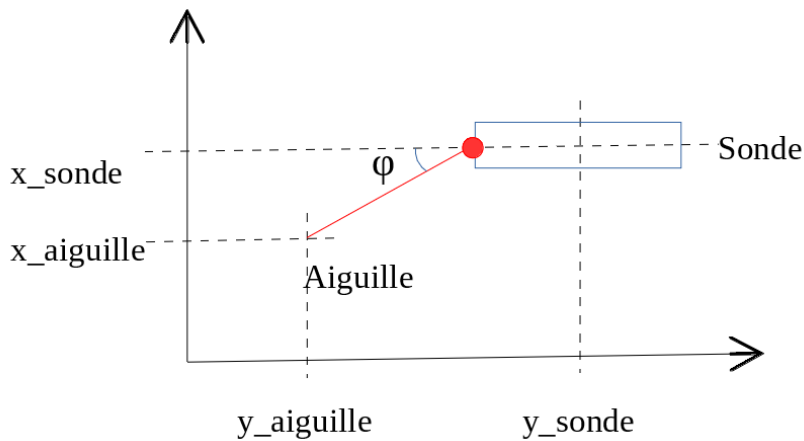


Figure 30- Représentation de l'introduction d'une aiguille dans un pain de gel dans la réalité et sur une image échographique

Le raisonnement est le suivant et est détaillé un peu après:

- Nous calculons tout d'abord l'orientation de l'aiguille par rapport à la sonde.
- Nous calculons ensuite le point d'intersection P entre l'aiguille et le pain de gel (point de ponction) pour que la position de l'aiguille ne dépende plus de son inclinaison ou orientation.
- À partir de ce point nous calculons les coordonnées du point de l'extrémité de l'aiguille E où le fluide anesthésiant sera injecté.
- Enfin, nous parcourons numériquement les points entre P et E en affichant uniquement ceux qui appartiennent au plan de la sonde.

Orientation de l'aiguille par rapport à la sonde

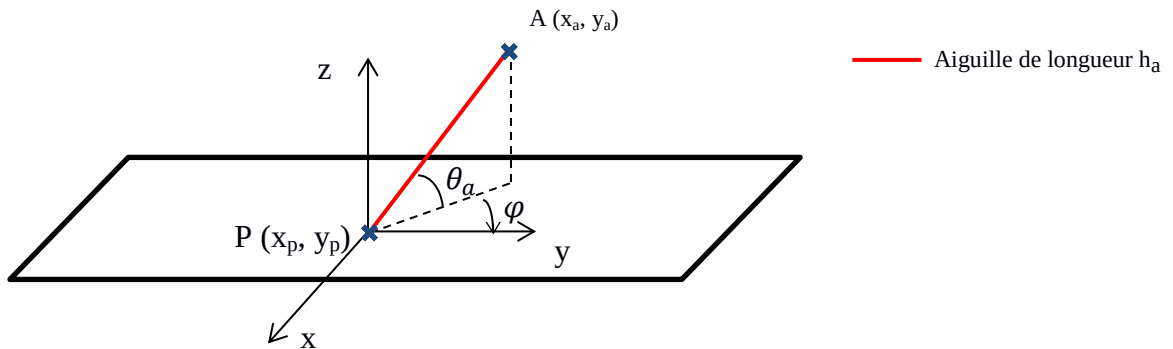


Dans la majorité des cas, l'aiguille est plantée très proche de la sonde. Nous pouvons donc approximer les coordonnées du point de ponction (en rouge ci-dessus) par les coordonnées de la sonde. L'orientation est alors déterminée à l'aide de la caméra par une simple formule de trigonométrie :

$$\varphi = \arctan\left(\frac{x_{aiguille} - x_{sonde}}{y_{aiguille} - y_{sonde}}\right)$$

Cette partie correspond à la fonction *orientation()* sur le diagramme de classes.

Coordonnées du point de ponction



Nous mesurons x_a et y_a à l'aide du traitement d'images. Nous disposons alors de leurs valeurs et nous pouvons ainsi isoler les termes qui nous intéressent à savoir x_p et y_p dans les expressions suivantes :

$$x_a = h_a \cos(\theta_a) \sin(\varphi) + x_p$$

$$y_a = h_a \cos(\theta_a) \cos(\varphi) + y_p$$

Les coordonnées du point de ponction (intersection entre l'aiguille et le pain de gel) sont donc :

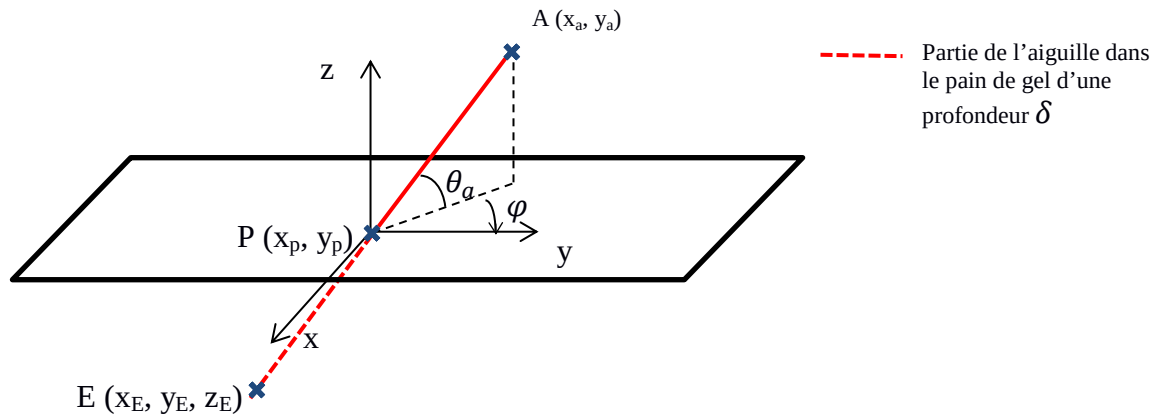
$$x_p = x_a - h_a \cos(\theta_a) \sin(\varphi)$$

$$y_p = y_a - h_a \cos(\theta_a) \cos(\varphi)$$

Nous détaillons les coordonnées de ce point pour alléger les expressions des points suivants.

Cette partie correspond à la fonction *pt_ponction()* sur le diagramme de classes.

Coordonnées de l'extrémité de l'aiguille



Les coordonnées de l'extrémité réelle (visible ou non par la sonde) sont :

$$x_E = x_p - \delta \cos(\theta_a) \sin(\varphi)$$

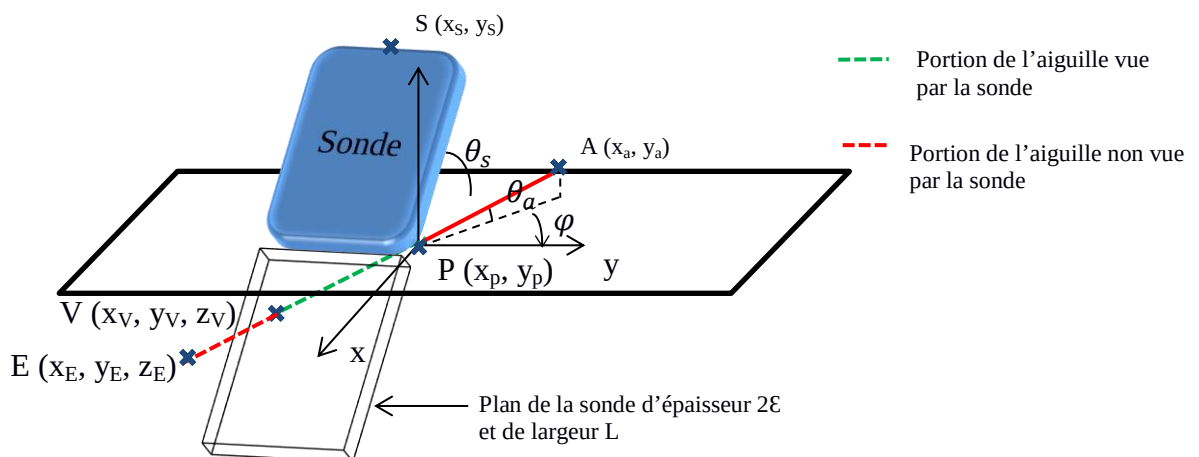
$$y_E = y_p - \delta \cos(\theta_a) \cos(\varphi)$$

$$z_E = -\delta \sin(\theta_a)$$

Cette partie correspond à la fonction *pt_extremite()* sur le diagramme de classes.

Appartenance au plan de la sonde

L'objectif est désormais de parcourir numériquement l'aiguille entre le point de ponction et le point d'extrémité de l'aiguille et de trouver les coordonnées du dernier point visible V.



Nous avons choisi d'introduire une marge d'erreur due aux fluctuations des mesures des capteurs et donc de considérer que le plan de la sonde est en réalité un pavé d'épaisseur 2ϵ .

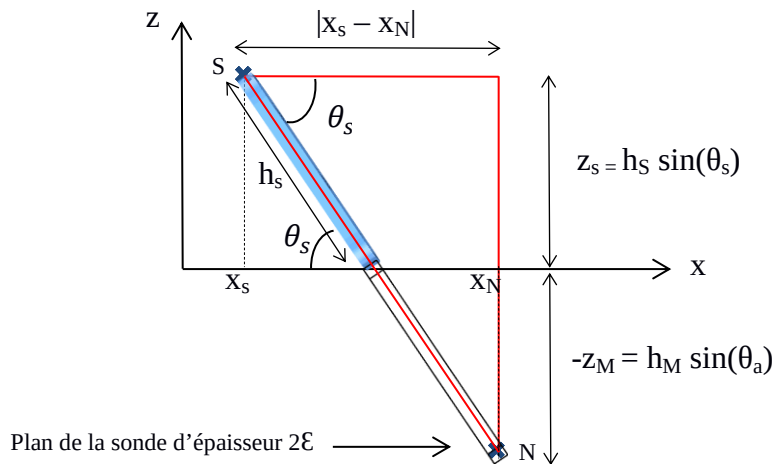
Posons un point M de profondeur h_M (que nous ferons varier avec un pas dh) parcourant l'aiguille entre les points P et E. Ses coordonnées sont :

$$x_M = x_p - h_M \cos(\theta_a) \sin(\varphi)$$

$$y_M = y_p - h_M \cos(\theta_a) \cos(\varphi)$$

$$z_M = -h_M \sin(\theta_a)$$

Il faut vérifier si y_M appartient à $[y_s - L/2, y_s + L/2]$ avec $L = 7\text{cm}$ la largeur de la sonde. En ce qui concerne x_M , intéressons-nous au même schéma que ci-dessus mais vu dans le plan (O,x,z) :



Une condition d'appartenance au plan de la sonde est que x_M appartienne à $[x_N - \mathcal{E}, x_N + \mathcal{E}]$ avec \mathcal{E} la marge d'erreur. L'objectif est donc de donner une expression de x_N .

Considérons le triangle rectangle en rouge sur la figure ci-dessus. Nous savons que

$$|x_s - x_N| = \frac{|z_s| + |z_M|}{\tan(\theta_s)}$$

en supposant que θ_s ne soit pas nul : on ne place jamais la sonde à l'horizontal.

Nous avons un cas particulier si la sonde n'est pas inclinée ($\theta_s = 90^\circ$) : $\tan(\theta_s)$ n'est pas défini mais nous remarquons que dans ce cas-là $x_s = x_N$

La valeur de $|x_s - x_N|$ dépend de l'inclinaison de la sonde :

- Si $\theta_s < 90^\circ$, $x_s > x_N$ et donc $x_N = x_s - \frac{h_s \sin(\theta_s) + h_M \sin(\theta_a)}{\tan(\theta_s)}$
- Si $\theta_s > 90^\circ$, $x_s < x_N$ et donc $x_N = - (x_s - \frac{h_s \sin(\theta_s) + h_M \sin(\theta_a)}{\tan(\theta_s)})$

Ces conditions imposent que θ_s diminue lorsqu'on incline la sonde vers la caméra. L'accéléromètre doit donc être face à la caméra pour respecter cette convention.

Pour conclure, les conditions d'appartenance à la sonde sont :

$$\begin{cases} x_M \in [x_N - \mathcal{E}, x_N + \mathcal{E}] \\ y_M \in [y_S - L/2, y_S + L/2] \end{cases}$$

Avec

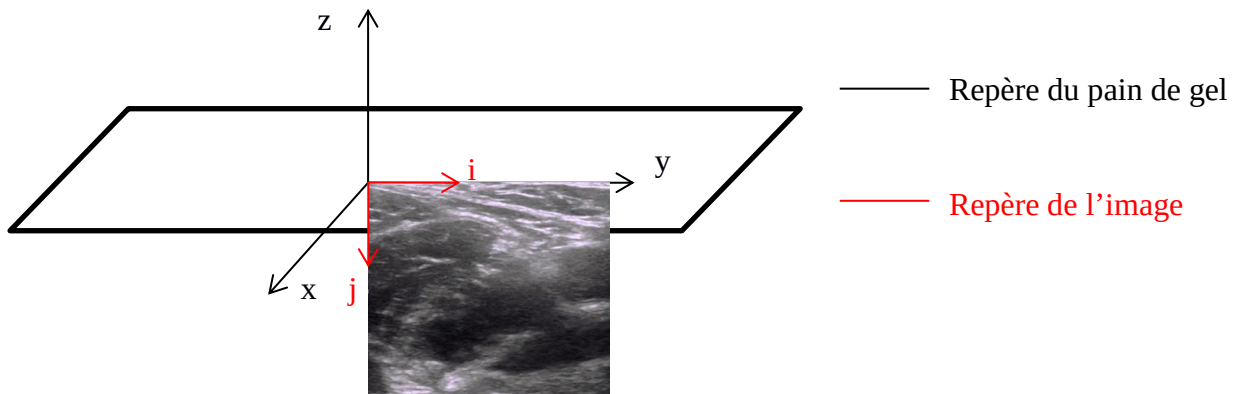
$$\begin{aligned} x_N &= x_S - \frac{h_S \sin(\theta_S) + h_M \sin(\theta_a)}{\tan(\theta_S)} \quad \text{si } \theta_S < 90^\circ \\ x_N &= -(x_S - \frac{h_S \sin(\theta_S) + h_M \sin(\theta_a)}{\tan(\theta_S)}) \quad \text{si } \theta_S > 90^\circ \\ x_N &= x_S \quad \text{si } \theta_S = 90^\circ \end{aligned}$$

Cette partie correspond à la fonction *intervalle_plan()* sur le diagramme de classes. Ensuite en parcourant numériquement l'aiguille entre les points P et E, nous attribuons le dernier point vérifiant ces conditions au point V. Pour commencer, nous afficherons un segment qui reliera un sommet du haut de l'image échographique avec le dernier point visible par la sonde.

Nous avons ensuite amélioré le programme en affichant le segment reliant le premier point visible et le dernier point visible. Cela rend le simulateur le plus réaliste possible. Cette partie correspond à la fonction *premier_pt_visible()* sur le diagramme de classes.

Affichage à l'écran

Nous avons finalement les coordonnées des points V dans le repère du pain de gel. Il nous faut désormais les convertir pour passer dans le repère de l'image échographique.



L'image échographique se trouve dans le plan (O, y, z) du repère du pain de gel. Les coordonnées (en pixels) du point à afficher à l'écran seront donc $(\lambda_i y_v, \lambda_j (-z_v))$ avec λ_i et λ_j les coefficients pour convertir les coordonnées en pixels. Nous les calculons ainsi :

525 pixels pour 7cm

670 pixels pour 5cm

$$\lambda_i = 525 / 7 = 75 \text{ pixels / cm}$$

$$\lambda_j = 670 / 5 = 134 \text{ pixels / cm}$$

oudeville A. – Krayem I. – Mchaymech D.

IV.2.2 Intégration physique

Conformément à l'architecture physique définie précédemment, les deux accéléromètres, le capteur de distance IR ainsi que le bouton poussoir ont été reliés à la carte Arduino Mega par des connexions filaires. La carte Arduino a été ensuite directement reliée à l'ordinateur, de même que la caméra. Cette dernière n'a pas été branchée directement sur la carte Arduino car comme dit précédemment, la puissance du processeur de la carte ne suffit pas à exploiter le flux vidéo.

Les accéléromètres ont été fixés sur la sonde et l'aiguille à l'aide de bagues en plastique. Pour éviter d'abimer ces composants, ils ont également été protégés en les mettant à l'intérieur de la mire utilisée pour le traitement d'image. Ces mires (balles de pingpong colorées) ont été fixées sur le sommet de l'aiguille et de la sonde en laissant passer les câbles utiles des capteurs à travers. Le capteur de distance de l'aiguille a été fixé sous un morceau de mousse au niveau de la partie dédiée à la prise en main de l'aiguille. Cela permet à l'utilisateur d'avoir une bonne prise en main de l'aiguille tout en l'empêchant d'interférer avec le flux lumineux nécessaire au fonctionnement du capteur de distance IR. Des bandes colorées ont été ajoutées sur l'aiguille afin de définir sa prise en main. En effet, une bonne prise en main de l'aiguille est indispensable au bon fonctionnement du simulateur puisque les valeurs renvoyées par les accéléromètres de l'aiguille dépendent de la façon avec laquelle l'utilisateur tient l'aiguille. Le bouton poussoir a quant à lui été fixé sur le piston d'une seringue reliée à la carte Arduino par un câble électronique.

Les différents câbles utilisés lors de la conception du simulateur ont été recouverts d'une gaine en plastique afin de les protéger de la détérioration et dans le but de rendre les câbles plus souples et moins désordonnés. La gaine permet également de cacher les différents câbles afin d'améliorer le système du point de vue de son design.

La caméra a été fixée sur un tuteur en bois et sera orientée en direction des mires de couleur (balles de pingpong). Un mur blanc en fond, peut être rajouté derrière les mires afin d'optimiser au mieux le traitement des images qui pourrait être défectueux à cause de la couleur des éléments environnant. Des LED ont également été ajoutées à l'intérieur des mires de couleur (balles de pingpong) afin de faciliter le traitement des images obtenue avec la caméra.

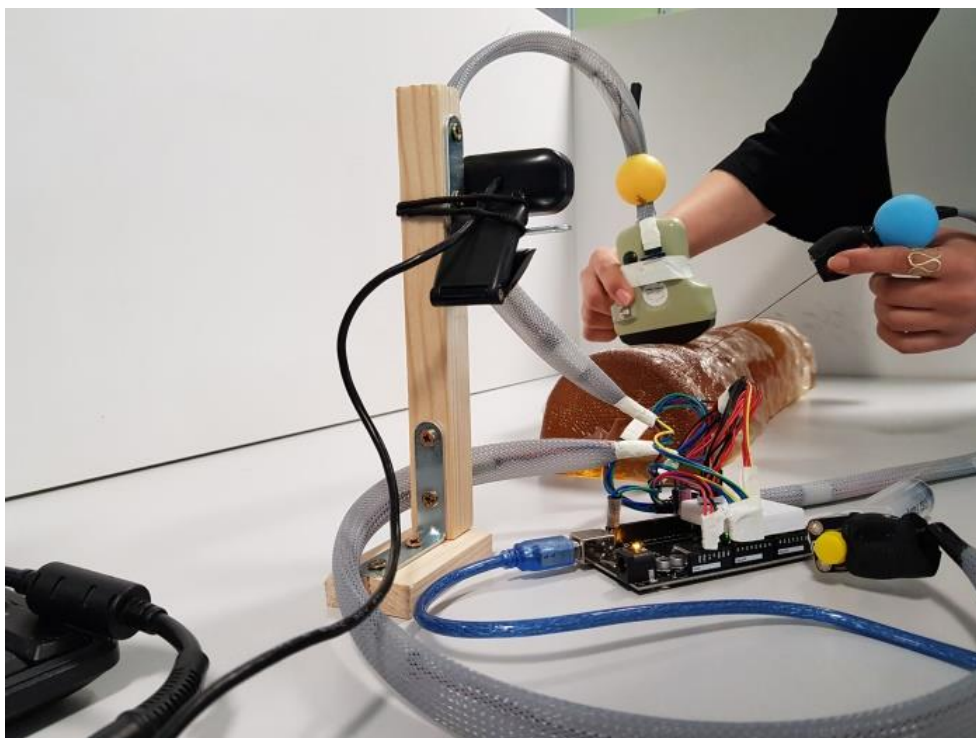


Figure 31- Intégration physique du système final

IV.3 Tests de validation fonctionnelle

Les tests de validation fonctionnelle ont pour objectif de vérifier si les exigences définies au tout début du projet ont bien été traitées et correspondent aux différentes fonctions que possède notre système. Pour vérifier si les différentes exigences ont été respectées, nous les avons testées une par une.

Développer un dispositif d'injection d'anesthésiant par simulation pour l'apprentissage de l'anesthésie locorégionale par des étudiants	
1. Le système reposera sur un milieu physique composé par un patient (simulé par un pain de gel), d'une sonde échographique, de capteurs de position et d'une aiguille	
1.1. La position de la sonde est exprimée dans le repère du pain de gel	exigence non fonctionnelle
1.2. Les mouvements de la sonde sont limités par la surface du pain de gel	contrainte
1.3. Des capteurs vont repérer la position et l'inclinaison de la sonde	exigence fonctionnelle
1.4. Des capteurs repèrent la position et l'inclinaison de l'aiguille	exigence fonctionnelle
1.5. Un capteur mesure la quantité de fluide injectée	exigence fonctionnelle
1.6. Un capteur mesure la profondeur d'introduction de l'aiguille dans le pain de gel	exigence fonctionnelle
2. Le système se composera d'une interface graphique affichant les images échographiques ainsi que les images de synthèse	
2.1. Une interface graphique permet au formateur de choisir le scénario (quelle partie du corps humain, quelle pathologie, ...)	exigence fonctionnelle
2.2. Les images affichées à l'écran proviennent d'une banque d'image	exigence fonctionnelle
2.2.1. Ces images proviennent d'une vraie échographie	exigence fonctionnelle
2.2.1.1. Un système est mis en place pour récupérer les images pour toutes les positions et inclinaisons de la sonde	exigence fonctionnelle
2.2.2. Le nombre d'images est limité par la taille de la zone échographiée	contrainte
2.2.3. L'image est affichée en fonction de la position et de l'inclinaison de la sonde	exigence fonctionnelle
2.3. L'image de l'aiguille doit être simulée et intégrée à l'image de l'échographie	exigence fonctionnelle
2.3.1. L'image de l'aiguille doit respecter le plan de coupe de la sonde	contrainte
2.3.2. L'image de l'aiguille est superposée à l'image affichée	exigence fonctionnelle
2.3.3. L'image de l'aiguille sera affichée en fonction de sa position, de son inclinaison, et de sa profondeur d'introduction	exigence non fonctionnelle
2.4. L'image du fluide doit être simulée et intégrée à l'image de l'échographie	exigence fonctionnelle
2.4.1. Le programme simule la dispersion du fluide	exigence fonctionnelle
2.4.2. Le programme superpose la simulation à l'image affichée	exigence fonctionnelle
2.4.3. L'image du fluide doit respecter le plan de coupe de la sonde	contrainte
2.5. Le système affichera des images de synthèse en coordination avec les positions de la sonde et de l'aiguille	exigence fonctionnelle
3. Le système permettra de transmettre des informations du milieu physique vers l'écran de contrôle	
3.1. Les capteurs communiquent la position et l'inclinaison de la sonde	exigence non fonctionnelle
3.1.1. Le programme sélectionne l'image en fonction de ces données	exigence non fonctionnelle
3.1.2. Le programme construit une image de synthèse en fonction de ces données	exigence non fonctionnelle
3.2. Les capteurs communiquent la position, l'inclinaison, et la profondeur d'introduction de l'aiguille	exigence non fonctionnelle
3.2.1. Le programme simule l'image de l'aiguille en fonction des données liées à la sonde et celles liées à l'aiguille	exigence non fonctionnelle
3.3. Un capteur communique la quantité de fluide injecté	exigence non fonctionnelle
3.3.1. Le programme simule la dispersion du fluide en fonction de ces données	exigence non fonctionnelle

Figure 32- Diagramme des exigences du simulateur

Exigence 1.1 : La position de la sonde est exprimée dans le repère du pain de gel

Protocole : on place la sonde à une extrémité du pain de gel, qui représente l'origine de notre repère. On déplace la sonde, en mesurant ce déplacement à l'aide d'une règle.

Vérification : les mesures renvoyées par le programme Python doivent correspondre au déplacement effectué en réalité par la sonde. > **Exigence modifiée puis validée** car l'espace réel est différent de l'espace de représentation des images car ce dernier a été dilaté afin d'améliorer la résolution spatiale du système : prioritaire d'après le client. Un accord a été convenu avec le client pour laisser de côté cette exigence.

Exigence 1.2 : Les mouvements de la sonde sont limités par la surface du pain de gel

Protocole : On pose la sonde sur le pain de gel. Puis on la déplace de manière à ce qu'il n'y ait plus d'images correspondant à la position de la sonde à afficher.

Vérification : Lorsque la sonde s'éloigne de la caméra, à partir d'une certaine distance, une image signalant que la sonde se trouve en dehors de la zone s'affiche à l'écran. > **Exigence validée.**

Exigence 1.3 : Des capteurs vont repérer la position et l'inclinaison de la sonde

Protocole : On incline et on déplace la sonde puis on vérifie que les informations données correspondent à la réalité.

Vérification : L'angle mesuré par l'accéléromètre ainsi que les coordonnées de la sonde obtenues par la caméra correspondent bien à la réalité. > **Exigence validée.**

Exigence 1.4 : Des capteurs repèrent la position et l'inclinaison de l'aiguille

Protocole : On incline et on déplace l'aiguille puis on vérifie que les informations données correspondent à la réalité.

Vérification : L'angle mesuré par l'accéléromètre ainsi que les coordonnées de l'aiguille obtenues par la caméra correspondent bien à la réalité. > **Exigence validée.**

Exigence 1.5 : Un capteur mesure la quantité de fluide injectée

Protocole : On appuie sur le bouton et on repère sur l'écran un point dont la taille varie en fonction du temps de pression sur le bouton.

Vérification : Lorsqu'on appuie sur le bouton, on observe bien sur l'écran un point à l'endroit correspondant, néanmoins sa taille est fixe car en discutant avec les clients nous avons fait en sorte que la taille du point ne dépende pas de la durée de pression sur le bouton. > **Exigence modifiée puis validée.**

Exigence 1.6 : Un capteur mesure la profondeur d'introduction de l'aiguille dans le pain de gel

Protocole : On introduit l'aiguille dans le pain de gel et on regarde si la longueur de l'aiguille introduite dans le pain de gel correspond à la longueur associée à l'objet aiguille dans le logiciel.

Vérification : **Exigence validée.**

Exigence 2.1 : Une interface graphique permet au formateur de choisir le scénario

Vérification : **Exigence validée** même si nous ne disposons pas encore de banque d'images pour les autres scénarios.

Exigence 2.2 : Les images affichées à l'écran proviennent d'une banque d'images

Vérification : **Exigence validée.**

Exigence 2.2.1 : Ces images proviennent d'une vraie échographie

Vérification : Les images proviennent d'une banque d'images recueillie au CESIM à l'aide d'un vrai échographe avec Mr Yann Jean. > **Exigence validée.**

Exigence 2.2.1.1 : Un système est mis en place pour récupérer les images pour toutes les positions et inclinaisons de la sonde

Vérification : un protocole de prise d'image a été élaboré par l'équipe de l'année dernière afin de faire une acquisition d'images. > **Exigence validée.**

Exigence 2.2.2 : Le nombre d'images est limité par la taille de la zone échographiée

Vérification : Pour 10 cm de la zone du nerf fémoral, avec un nombre de fps = 20 (frames per second), on obtient un nombre total de 2775 images pour cette banque. > **Exigence validée.**

Exigence 2.2.3 : L'image est affichée en fonction de la position et de l'inclinaison de la sonde

Vérification : On simule une position à l'Interface Homme Machine et on affiche à l'écran une image correspondante exacte ou rapprochée si la position n'existe pas dans la banque. > **Exigence validée.**

Exigence 2.3 : L'image de l'aiguille doit être simulée et intégrée à l'image de l'échographie

Protocole : On superpose l'image d'un trait gris sur la fenêtre graphique en fonction des coordonnées et de l'angle d'inclinaison de l'aiguille et de la sonde.

Vérification : **Exigence validée**

Exigence 2.3.1 : L'image de l'aiguille doit respecter le plan de coupe de la sonde

Protocole : On simule plusieurs positions de l'aiguille relativement à la sonde à l'Interface Homme Machine et on visualise les points appartenant au plan de la sonde.

Vérification : **Exigence non validée, fonction correspondante à corriger**

Exigence 2.3.2 : L'image de l'aiguille est superposée à l'image affichée

Protocole : On superpose l'image d'un trait gris sur la fenêtre graphique

Vérification : **Exigence validée**

Exigence 2.3.3 : L'image de l'aiguille sera affichée en fonction de sa position, de son inclinaison, et de sa profondeur d'introduction

Vérification : **Exigence non validée car dépend en partie de l'exigence 2.3.1**

Exigence 2.4 : L'image du fluide doit être simulée et intégrée à l'image de l'échographie

Protocole : On superpose l'image d'un point rouge sur la fenêtre graphique en fonction des coordonnées et de l'angle d'inclinaison de l'aiguille et de la sonde.

Vérification : **Exigence validée**

Exigence 2.4.1 : Le programme simule la dispersion du fluide

Vérification : **Exigence non validée car non traitée : non prioritaire d'après les clients**

Exigence 2.4.2 : Le programme superpose la simulation à l'image affichée

Vérification : **Exigence validée**

Exigence 2.4.3 : L'image du fluide doit respecter le plan de coupe de la sonde

Vérification : **Exigence non validée car non traitée : non prioritaire d'après les clients**

Exigence 2.5 :

Protocole : On pose la sonde sur le pain de gel avec une certaine inclinaison et on vérifie que les raccords entre les images sont faits correctement.

Vérification : Exigence **modifiée puis validée** après avoir convenu avec les clients que les images échographiques correspondant à la sonde ne doivent pas dépendre de l'angle d'inclinaison de la sonde à cause de mauvais raccords.

Exigence 3.1 : Les capteurs communiquent la position et l'inclinaison de la sonde

Vérification : Communication par des câbles entre la caméra et l'ordinateur ainsi qu'entre l'accéléromètre et la carte Arduino >**Exigence validée**

Exigence 3.1.1 : Le programme sélectionne l'image en fonction de ces données

Vérification : Transmission par câble entre la carte Arduino et l'ordinateur >**Exigence validée**

Exigence 3.1.2 : Le programme construit une image de synthèse en fonction de ces données

Vérification : Transmission par câble entre la caméra, la carte Arduino et l'ordinateur >**Exigence validée**

Exigence 3.2 : Les capteurs communiquent la position et l'inclinaison de l'aiguille

Vérification : Communication par des câbles entre la caméra et l'ordinateur ainsi qu'entre l'accéléromètre et la carte Arduino >**Exigence validée**

Exigence 3.2.1 : Le programme simule l'image de l'aiguille en fonction des données liées à la sonde et celles liées à l'aiguille

Vérification : Transmission par câble entre la caméra, la carte Arduino et l'ordinateur >**Exigence validée**

Exigence 3.3 : Un capteur communique la quantité de fluide injecté

Vérification : Transmission par câble entre le bouton et la carte Arduino >Exigence validée

Exigence 3.3.1 : Le programme simule la dispersion du fluide en fonction de ces données

Vérification : Transmission par câble entre la carte Arduino et l'ordinateur>Exigence validée

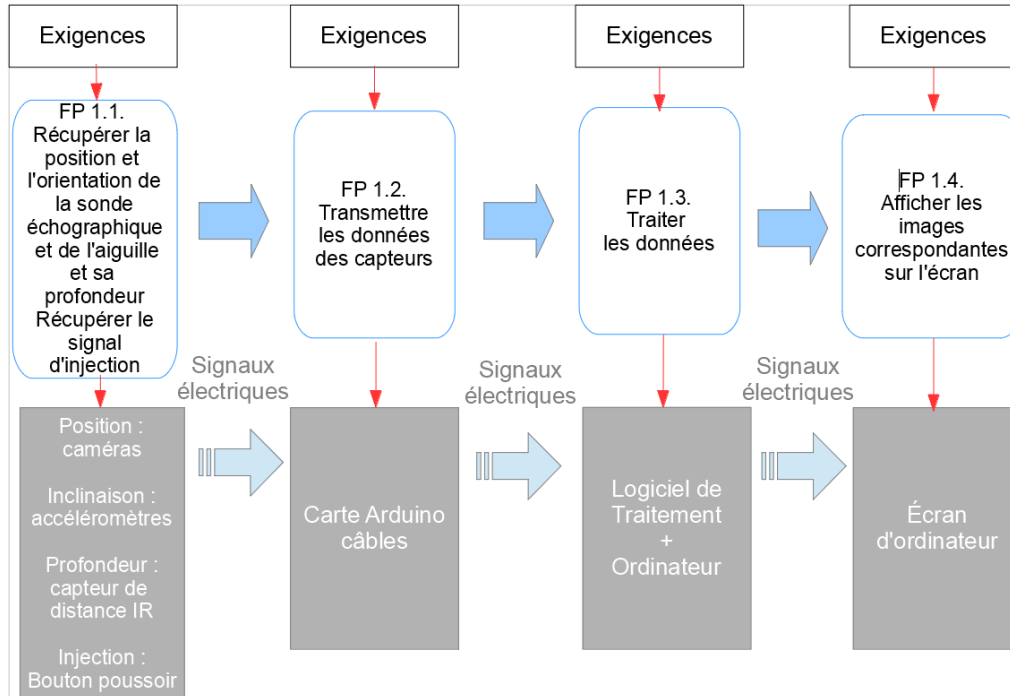


Figure 33- Rappel de l'architecture physique du simulateur

Fonction/composant	C1 : accéléromètre	C2 : capteur de distance IR	C3 : caméra	C4 : bouton poussoir	C5 : carte Arduino	C6 : ordinateur /Interface Homme Machine	C7 : écran d'ordinateur
FP 1.1 Position							
FP 1.1 Inclinaison							
FP 1.1 Profondeur							
FP 1.1 Injection							
FP 1.2 Transmettre							
FP 1.3 Traiter							
FP 1.4 Afficher							

Figure 34- Matrice d'allocation des fonctions

Les résultats des tests de validation fonctionnelle sont plutôt favorables à une bonne réalisation de notre système même si quelques exigences ont dû être modifiées et quelques exigences concernant l'affichage correct de l'aiguille sur l'écran n'ont pas encore été validées au moment où ce rapport a été rédigé. La matrice d'allocation des fonctions montre que tous les composants ont été alloués à une fonction, ce qui ne présente rien d'anormal et ne remet pas en cause la phase de réalisation de notre système.

V. Conclusion et perspectives

Le monde de la médecine étant étranger à notre formation, il nous a été nécessaire de nous immerger dans ce domaine, en particulier dans celui de l'anesthésie locorégionale pour en comprendre les problématiques. Nous avons eu l'opportunité de pouvoir manipuler une vraie sonde échographique et une aiguille sur un pain de gel. Cela nous a permis de confirmer notre compréhension des exigences et ainsi de partir confiants pour la conception et la réalisation du simulateur d'anesthésie locorégionale.

L'analyse de ces exigences nous a permis d'identifier clairement les différentes tâches principales à réaliser et ainsi de répartir le travail assez facilement.

Une fois la programmation des différents capteurs terminée, nous avons intégré physiquement et numériquement les différents composants du système à l'aide d'une Interface Homme Machine pratique pour améliorer au mieux la prise en main du système par l'utilisateur. Le travail d'ingénierie système réalisé nous a permis de gagner en confiance et de prévoir au mieux la phase de réalisation du système même si nous avons rencontré quelques difficultés lors de la réalisation du système. La dernière phase de l'ingénierie système nous a permis d'effectuer différents tests sur notre système afin d'en valider le fonctionnement et de qualifier son efficacité.

Ce projet d'une durée d'un an nous a permis d'en apprendre beaucoup plus sur le travail d'un ingénieur travaillant dans un domaine informatique. Tout d'abord, nous avons dû apprendre à travailler en équipe en répartissant les différentes tâches à faire de façon à optimiser la productivité du groupe à long terme. Ce projet nous a également appris à nous former nous-mêmes de façon autonome sur les différents aspects techniques abordés lors de la conception de notre système, notamment au niveau de la mise en place pertinente de capteurs destinés à remplir des fonctions établies à l'avance lors de la phase d'ingénierie système du projet. Il nous a également appris à travailler en commun lorsqu'il a fallu intégrer ensemble les différents éléments du système à l'aide de l'Interface Homme Machine. Ce projet nous a également confrontés à un certain nombre de responsabilités, en effet à l'origine, le système issu de ce projet est destiné à être utilisé par Yann Jan dans le cadre de sa thèse, dont une partie consiste en l'observation de l'apprentissage des apprentis anesthésistes à l'aide de simulateurs. Un travail de discussion avec les clients et de prise de décision notamment au niveau du choix des solutions techniques a dû être mise en place pour rendre le système fonctionnel avant la date limite.

A l'avenir, notre système pourrait être amélioré en suivant différentes voies. Par exemple, il serait intéressant de fabriquer d'autres banques d'images afin de disposer d'une zone de travail plus diversifiée pour les apprentis anesthésistes. Il serait également très utile de mettre en place un système d'assistance visuelle pour aider l'utilisateur du système à se repérer sur l'écran, par exemple en indiquant la direction dans laquelle se trouve l'aiguille lorsque celle-ci n'apparaît pas sur l'écran ou en légendant si besoin les différents éléments du corps présents sur l'image échographique. Enfin, un travail pourrait aussi être réalisé afin d'améliorer l'apparence du fluide anesthésiant lors de son injection dans la peau, pour l'instant simplement représenté par un point rouge à l'endroit correspondant.

Table des figures

Figure 1- Diagramme de Gantt pour la deuxième phase du projet	9
Figure 2- Tableau récapitulatif des réunions organisées avec les clients et les encadrants du projet	11
Figure 3- Tableau récapitulatif de gestion du budget.....	12
Figure 4- Architecture fonctionnelle du simulateur.....	13
Figure 5- Architecture physique du simulateur.....	15
Figure 6- Représentation de l'espace de couleur HSV ^[1]	16
Figure 7- Isolement de l'image de la balle par sa couleur.....	17
Figure 8- Détection du contour de la balle.....	17
Figure 9- Détection de contours pour la sonde et l'aiguille.....	18
Figure 10- Cône d'émission du transducteur	21
Figure 11- Mise en place de l'accéléromètre sur la sonde échographique.....	22
Figure 12- Mise en place de l'accéléromètre sur l'aiguille	22
Figure 13 - Mise en place du capteur sur l'aiguille et prise en main de l'aiguille par l'utilisateur.....	23
Figure 14- Bouton poussoir utilisé pour simuler l'injection de fluide anesthésiant.....	24
Figure 15- Introduction avec orientation de l'aiguille dans le plan du faisceau de la sonde	25
Figure 16- Introduction avec orientation de l'aiguille dans le plan perpendiculaire au faisceau de la sonde	25
Figure 17- Mise en place du bouton poussoir sur le système.....	25
Figure 18- Le cycle en V en ingénierie système	27
Figure 19- Mesure de l'angle avec l'accéléromètre	28
Figure 20- Tableau de test unitaire pour l'accéléromètre	29
Figure 21- Tableau de test unitaire pour le capteur de distance IR	30
Figure 22- Distance vraie (en rouge) et distance mesurée (en bleu) en fonction de l'angle d'inclinaison de l'aiguille en degrés	31
Figure 23- Mesure de la distance par le capteur IR en fonction de l'angle d'inclinaison de l'aiguille	31
Figure 24- Pain de gel avant et après couverture en papier.....	32
Figure 25- Rayon de la balle sur l'image produite par la caméra en fonction de la distance entre la balle et la caméra	33
Figure 26- Test unitaire du système balle/caméra	33
Figure 27- Diagramme de classes du logiciel générant l'Interface Homme Machine.....	34
Figure 28- Affichage du menu de l'application.....	35
Figure 29- Affichage de la partie simulation de l'application	35
Figure 30- Représentation de l'introduction d'une aiguille dans un pain de gel dans la réalité et sur une image échographique.....	36
Figure 31- Intégration physique du système final.....	41
Figure 32- Diagramme des exigences du simulateur.....	42
Figure 33- Rappel de l'architecture physique du simulateur.....	45
Figure 34- Matrice d'allocation des fonctions.....	45

Bibliographie

- [1] anesthesia, 1ère WEB TV dédiée à la formation en anesthésie. 2013. <http://www.anesthesia.fr> . (consulté le 21/09/2017)
- [2] Wikipédia. Interactions homme-machine. https://fr.wikipedia.org/wiki/Interactions_homme-machine . (consulté le 14/04/2018)
- [3] Wikipédia. Diagramme de Gantt. https://fr.wikipedia.org/wiki/Diagramme_de_Gantt . (consulté le 14/01/2018)
- [4] Go Tronic. Capteur de distance 10cm 2489. 2015. <https://www.gotronic.fr/art-capteur-de-distance-10cm-2489-24230.htm> . (consulté le 09/11/2017)
- [5] Farnell element14. L6202 - Driver pont complet DMOS. 2015. <http://fr.farnell.com/stmicroelectronics/l6202/driver-de-pont-dmos/dp/1271291> . (consulté le 14/01/2017)
- [6] Farnell element14. MCUST18A40B12RS - Capteur ultrasonique. 2017. http://fr.farnell.com/multicomp/mcust18a40b12rs/transmitter-40khz-18mm-metal/dp/2362674?ost=MCUST18A40B12RS&ddkey=http%3Afr-FR%2FElement14_France%2Fsearch . (consulté le 14/01/2018)
- [7] Go Tronic. Module accéléromètre 101020051. 2015. <https://www.gotronic.fr/art-module-accelerometre-101020051-18956.htm> . (consulté le 09/11/2017)
- [8] Go Tronic. Carte arduino MEGA 2560. <https://www.gotronic.fr/art-carte-arduino-mega-2560-12421.htm> . (consulté le 17/02/2018)
- [9] Go Tronic. Carte Raspberry Pi3 B 1 GB. <https://www.gotronic.fr/art-carte-raspberry-pi3-b-1-gb-24532.htm> . (consulté le 17/02/2018)
- [10] Wikipédia. Triangulation. <https://fr.wikipedia.org/wiki/Triangulation> . (consulté le 02/11/2017)
- [11] Wikipédia. Teinte Saturation Valeur. https://fr.wikipedia.org/wiki/Teinte_Saturation_Valeur . (consulté le 19/01/2018)
- [12] Cours de Capteurs et systèmes de mesure : UV 1.7 à l'ENSTA Bretagne. Benoit Zerr. https://moodle.ensta-bretagne.fr/pluginfile.php/44259/mod_resource/content/10/compound_ing1_2017_2018.pdf . (consulté le 26/04/2018)

Annexe 1 : Mesure de la précision d'un capteur ^[12]

La valeur moyenne m est définie par: $m = \frac{\sum_{i=1}^n a_i}{n}$

L'écart-type σ est défini par : $\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (v_i - m)^2}$

La fidélité d'un capteur est sa capacité à reproduire la même mesure dans les mêmes conditions.

L'écart type σ , caractérisant la dispersion des mesures autour de la moyenne est un bon indicateur de la fidélité d'un capteur. L'erreur de fidélité e_{fidel} est définie par:

$$e_{fidel} = \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (v_i - m)^2}$$

La justesse est la capacité d'un capteur à fournir des mesures s'approchant de la valeur "vraie" du mesurande (grandeur à mesurer).

L'écart entre la valeur "vraie" v_{vraie} et la moyenne m , des n valeurs mesurées est un bon indicateur de la justesse du capteur. L'erreur de justesse e_{just} est définie par :

$$e_{just} = |v_{vraie} - m| \text{ Avec } m = \frac{\sum_{i=1}^n v_i}{n}$$

L'erreur de justesse caractérise l'erreur systématique du capteur

La précision, ou plutôt, l'erreur de précision, du capteur est la combinaison :

- des erreurs systématiques définies par l'erreur de justesse
- des erreurs aléatoires définies par l'erreur de fidélité

Les erreurs de justesse e_{just} et de fidélité e_{fidel} sont non corrélées, il est donc possible d'utiliser la formule de combinaison simple :

$$e_{prec} = \sqrt{e_{just}^2 + e_{fidel}^2}$$