



# Rapport de projet

## Introduction

Le projet S.A.S.A. (Système Autonome de Stabilisation Active) a pour but de rendre les fusées amateurs actuelles plus permissive en leurs appliquant une autonomie.

Avec une stabilisation passive, la trajectoire ne peut être que légèrement parabolique ou rectiligne et aucune intervention ne peut être effectuée pour modifier celle-ci une fois les moteurs allumés.

On passerait donc d'un système passif purement basé sur l'aérodynamique et dont l'allumage du/des moteur(s) scellerait quelque action pour modifier la trajectoire à un système 'intelligent' pouvant effectuer des actions durant le vol permettant de stabiliser la trajectoire de la fusée.

Le système n'utilisera pas de RCS ou d'actionneur gyroscopique à cause du coût et de la masse du dispositif, de plus des actionneurs gyroscopiques peuvent provoquer des vibrations et déstabiliser la fusée.

## Les actions du dispositif

Une stabilisation active sera amenée grâce à des empennages amovibles et un moteur à propulsion vectorielle.

Un seul moteur ne pouvant effectuer de correction de roulis, c'est donc les empennages qui auront ce rôle et le moteur effectuera les corrections de tangage et de lacet.

Couplé à un ordinateur de bord, ce système peut alors devenir autonome.

Le système dispose également d'un module Bluetooth, si un appareil est appairé à celui-ci, il recevra les informations de pré-lancement tel que les valeurs de base du 9-DOF.

Le BNO055 (9-DOF) détectera le lancement, il s'en suivra donc l'activation complète du dispositif puis lorsque la phase d'atterrissage sera complète, l'accélération sera nulle, ce qui pourra arrêter le système.



## **Les contraintes**

Le système doit être le plus léger possible afin d'avoir le moins d'impact sur les performances, il doit également être le plus petit possible.

Les mouvements des servos se doivent d'être précis et fluides afin d'avoir le moins de marge d'erreur et ne pas avoir d'accoups.

Aucune intervention humaine doit être nécessaire durant tout le procédé post-lancement.

## **Matériel**

- 1 Carte microcontrôleur
- 3 Servos
- 1 Contrôleur servos
- 1 Module bluetooth
- 1 Module SD
- 1 Capteur de pression
- 1 9-DOF

La carte microcontrôleur utilisé est un Arduino Mega 2560, l'utilisation d'une carte nano est possible pour réduire la taille et la masse.

Les servos sont des FS0403 de Feetech.

Le contrôleur servos est un Micro Maestro 6 canaux.

Le module bluetooth est un module HC-05.

Le module SD est une breakout SD avec régulateur 3.3v.

Le capteur de pression est le BMP280 et est utilisé pour obtenir l'altitude.

La centrale inertielle 9-DOF est le BNO055.

## **Recherches**

De nombreuses recherches ont été effectués durant l'accomplissement de ce projet à commencer par trouver le matériel nécessaire respectant les contraintes.

De conséquentes recherches sur la stabilité et l'aérodynamisme ont également été effectués, le système exercera son rôle dans un environnement contenant énormément de variables toutes cruciales, le moteur vectoriel et les empennages mobiles sont des acteurs importants et auront donc une incidence importante, ils ne doivent évidemment pas être négligés.



## Model 3D

Le model 3D a été d'une grande aide, il a servi de référence pour le prototype et à aider à visualiser la mobilité de certains éléments ainsi que les contraintes mécaniques liés à l'espace.

Ce model a également été utile pour l'impression d'éléments.

Une animation d'assemblage a été réaliser, ce qui permet d'obtenir une perspective différente.

La modélisation a été faite sur Blender, un logiciel libre permettant également de faire de l'animation et du texturage.

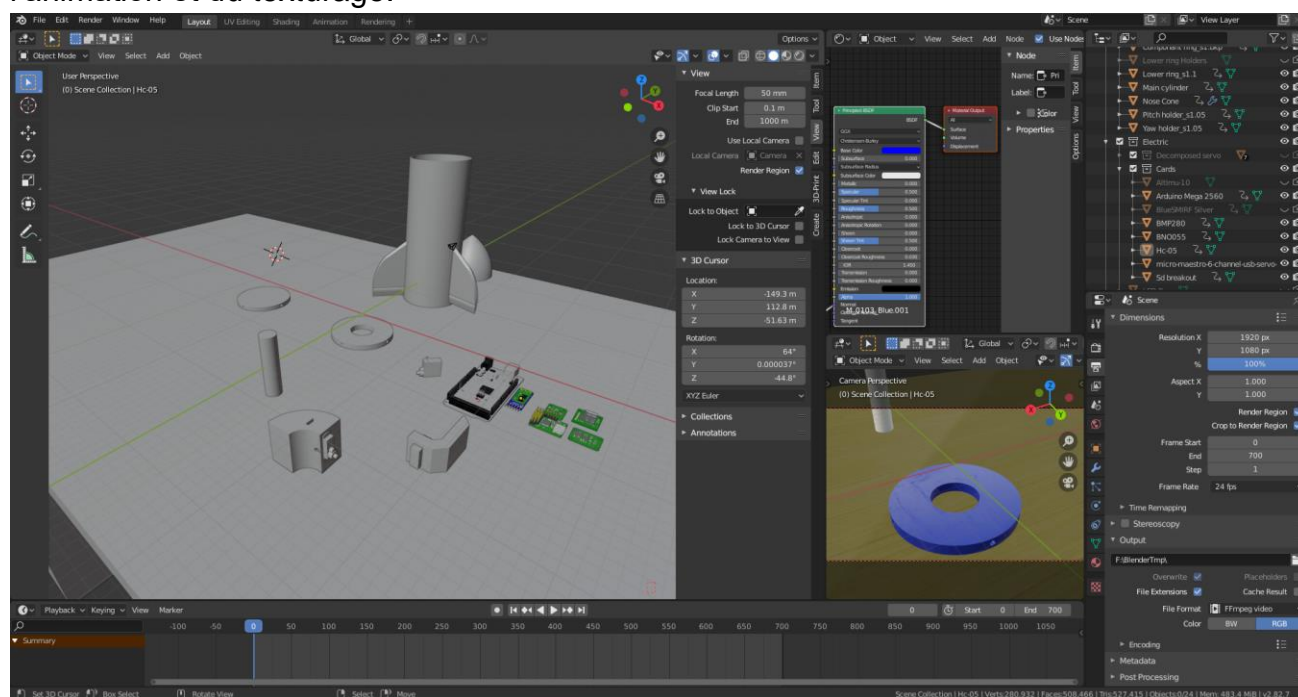


Figure 1 - Interface Blender

Le fichier compte une vingtaine d'objets constituant le système.



## Prototypage

Après la finalisation du model 3D, chaque pièce étant destinée à être imprimée doit généralement subir une transformation pour que celle-ci soit imprimable, surtout dans le cas où la pièce en question a été créée dans un logiciel non spécialisé dans la conception par ordinateur.

Les 2 principaux problèmes pouvant survenir :

- Overhangs : Les parties en surplomb, au-dessus d'un vide.  
Ils doivent être évités au maximum mais peuvent être résolus lors du passage dans un slicer en ajoutant des supports. Le temps d'impression sera donc augmenté.
- Non manifold geometry : Géométries ne pouvant exister dans le monde réel (faces à l'intérieur d'un objet, zones avec une épaisseur nulle, sommets ou arêtes déconnectées).

Après quelques essais d'impressions, un prototype en PLA a été réalisé, le slicer utilisé est Cura, il permet de paramétrer et de prévisualiser l'impression.

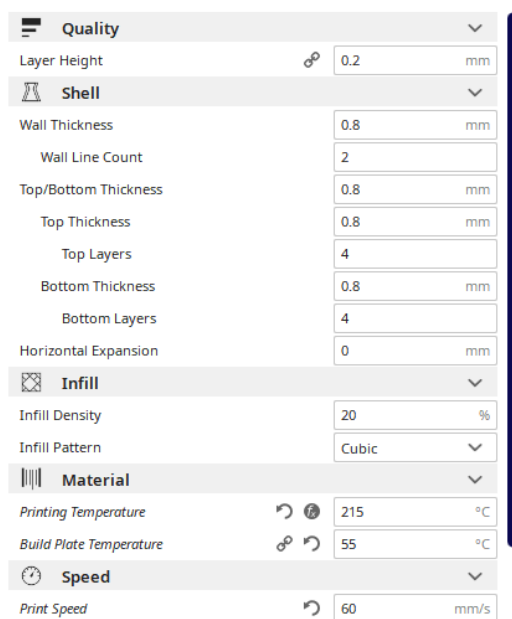


Figure 2 - Exemple de paramètres

Voici quelques-uns des paramètres les plus importants comme la hauteur de couche, l'épaisseur des parois, le pattern de remplissage ainsi que sa densité, la température et la vitesse d'impression.



Après découpage, le logiciel estime un temps d'impression et une quantité de matériaux utilisés, une prévisualisation du parcours de la buse et également disponible.

Figure 3 - Visualisation du découpage

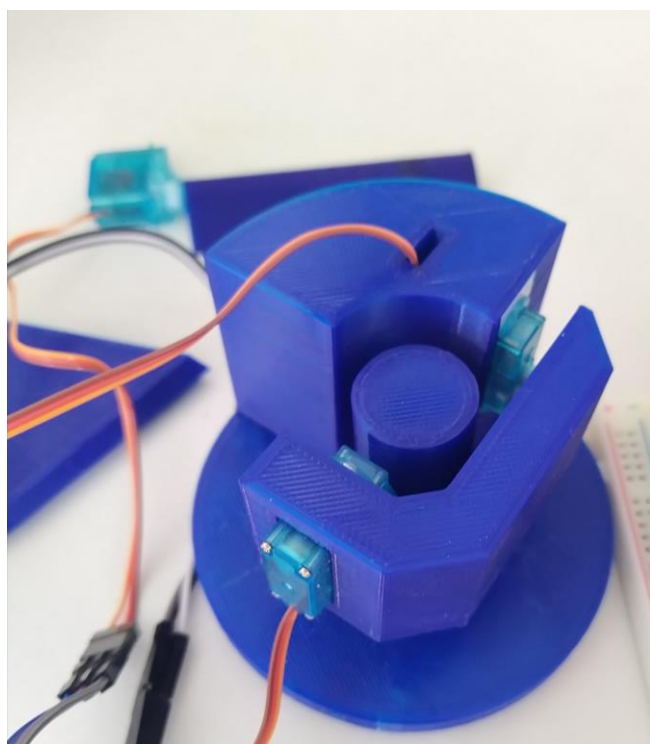
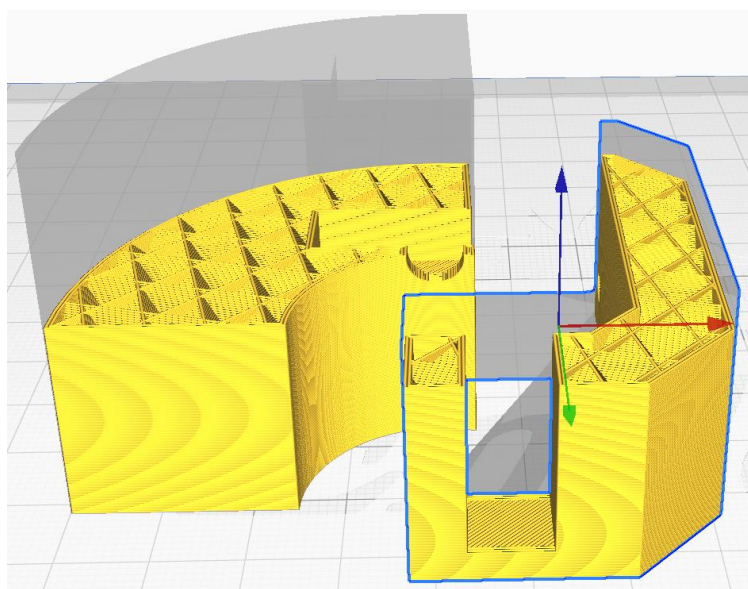


Figure 4 - Résultat d'impression





La représentation graphique a été faite sur Fritzing, le contrôleur servos n'étant pas disponible, il a été représenté par deux cartes à matrice et annoté en conséquence. Le module SD est visible en bas à droite des servos, à côté de celui-ci, le module Bluetooth.

A droite de l'écran se trouve le BNO055 et le BMP280.

## **Programmation**

Un souci matériel a causé un problème de communication avec le contrôleur de servos empêchant donc toute interaction avec ceux-ci, ce problème a été résolu.

Les acquisitions de données du capteur initial ont montré que celui-ci avait besoin d'être calibré pour fonctionner de correctement, la complexité d'utilisation de ce capteur m'a donc freiné, c'est pour cette raison que je l'ai remplacé par deux autres capteurs.

Un module Bluetooth (HC-05) envoie les données de pré-lancement à l'appareil appairé, pour des raisons d'intégrité le système n'est pas pilotable à distance, le module ne traite donc aucune données et ne réalise que des envoies.

Une carte microSD enregistre constamment l'état du programme et des capteurs, ce qui permet une persistance de données et une traçabilité d'éventuelles erreurs.

Le fichier résultant de cet enregistrement est un simple fichier texte, cependant, sa mise en page permet d'exploiter ce document dans un tableur afin d'en faire une analyse (graphique, calcul etc...).

Les valeurs d'axes fournies par le 9-DOF possèdent une échelle différente des servos, j'effectue donc une conversion afin de les rendre compréhensibles par ceux-ci.





## Simulation

Les simulateurs ne peuvent pas prendre en compte une stabilisation active, les simulations ont donc été utilisées pour voir l'impact du dispositif sur la fusée. Le principal outil utilisé est OpenRocket, il permet une création et une visualisation en 2D de la fusée.

Des paramètres environnementaux peuvent être spécifiés.

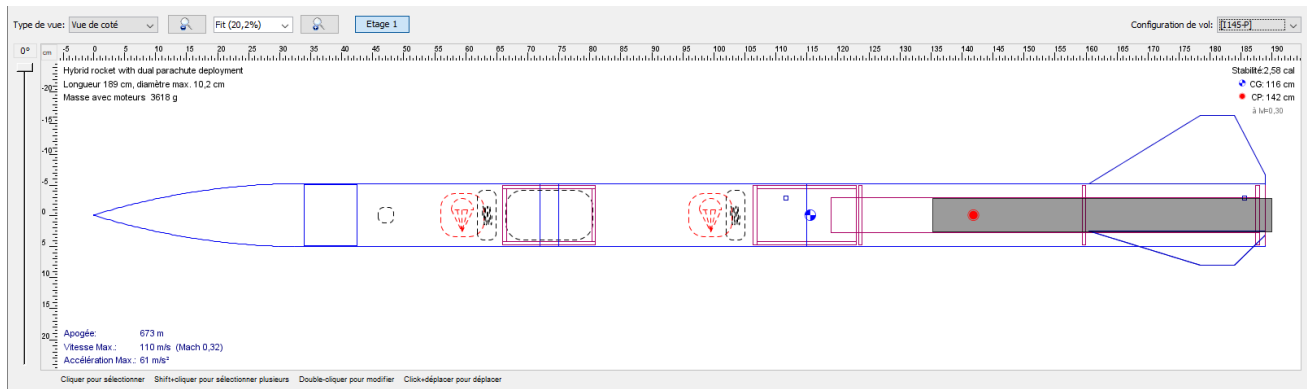


Figure 6 - Visuel de conception

Deux simulations ont été effectuées :

- La première avec la stabilisation passive, donc sans système

- La seconde avec la stabilisation active

La différence se traduit par la diminution des turbulences et perturbations et en ajoutant une masse simulant celle du système.



Projet :  
Système Autonome de  
Stabilisation Active

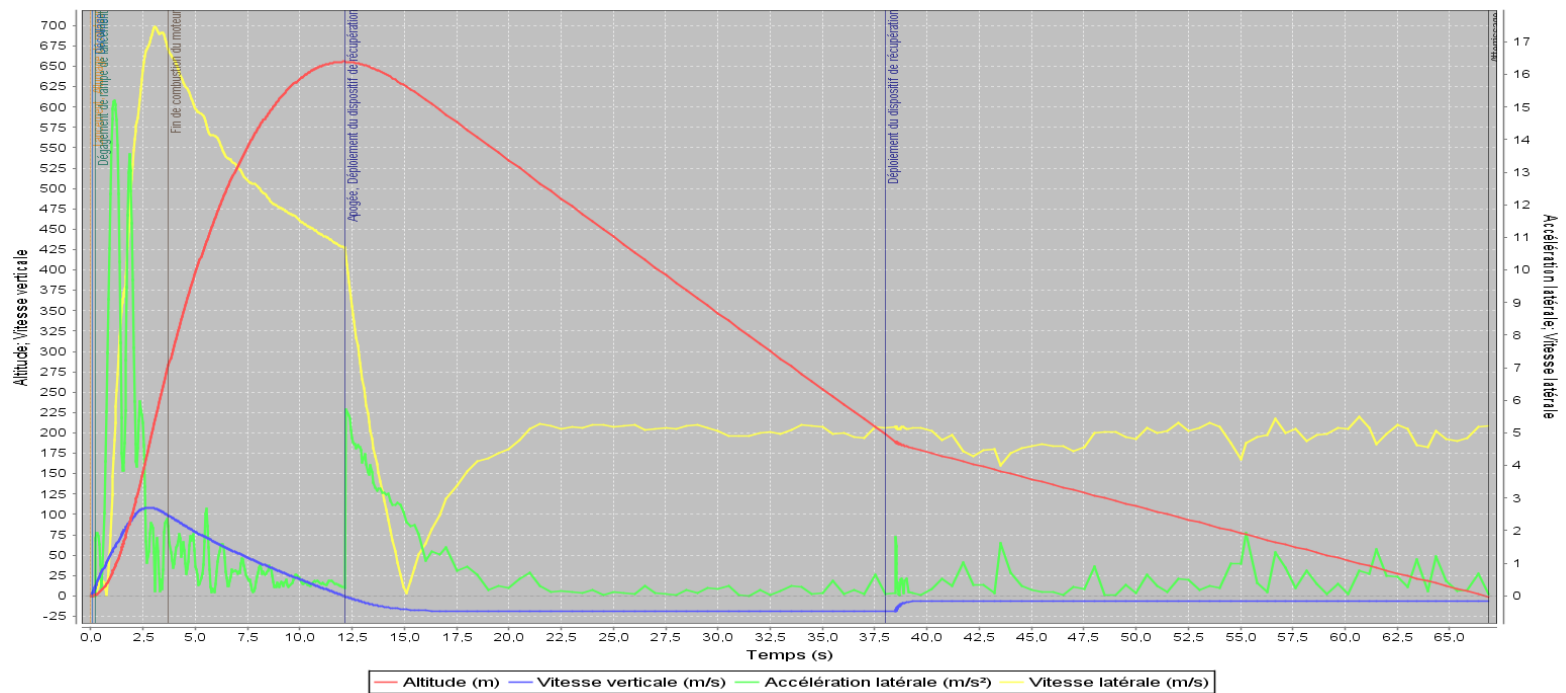


Figure 7 - Simulation stabilité passive

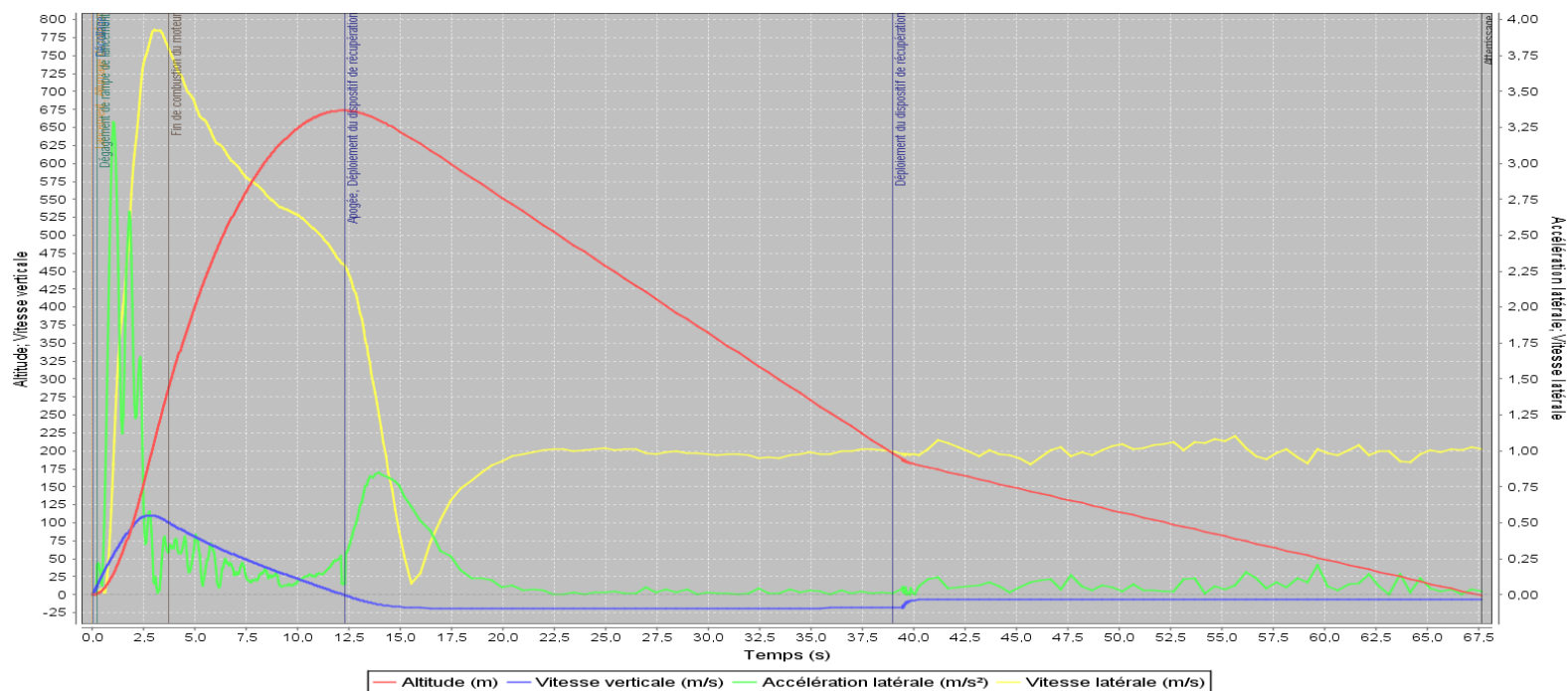


Figure 8 - Simulation stabilité active



Nous pouvons voir que la simulation avec stabilité active a une accélération latérale moindre car le système compense les aléas environnementaux.

De ce fait son altitude max (apogée) est plus grande, 674m contre 657m sur le 1<sup>e</sup> graphique.

Donc la masse que représente le dispositif est compensée par son action, une différence de 2.58% est visible sur l'apogée.

Ce pourcentage peut sembler dérisoire, en effet, appliqué à une apogée initiale de 10m, le système n'est que peu avantageux.

Cependant, certaines fusées expérimentales tel que la fusée « Elyjia » lancée en 2007 atteignent une altitude supérieure à 1km (1700m dans le cas d'Elyjia).

## **Code couleur est sonore du système**

---

- Led bleu, vert et rouge : Standby
- Led bleu, vert, rouge et bip sonore discontinue : Standby post atterrissage / amerrissage
- 2 courts bips sonores : Carte SD non détectée
- Led rouge et 3 longs bips sonores : Erreur critique (BNO055 ou BMP280 non détectée)
- 1 long bip sonore : Acquisition et mise en place des valeurs par défaut terminé

## **Glossaire**

---

- Axes aérodynamiques :
  - Roulis : Rotation autour de l'axe longitudinale de la fusée
  - Tangage : Rotation autour de l'axe latéral
  - Lacet : Rotation autour de l'axe vertical
- BMP280 : Capteur de pression / Altimètre
- BNO055 / 9-DOF : 9 Degrees Of Freedom, 3 axes données par 3 capteurs, un gyroscope, un accéléromètre et un magnétomètre.
- RCS : Reaction Control System