



Fiche résumé - PIE - SA005

Simulation d'inspection d'une éolienne par drone convertible

BENCHEKROUN Mohammed Wassim
DE-JAUREGUIBERRY Adrien
DROUARD Matthieu
JEHANNIN Brendan
LOIZON Noha
POIRIER Clément

14 mars 2017

Pour ce PIE, notre client a été Sterblue, une entreprise qui inspecte des infrastructures à l'aide de drones. Le besoin qu'il nous a initialement demandé a été le développement d'un simulateur "assez réaliste".

1 Objectif

Développer une simulation réaliste du Stork en fonction du plan de vol et des paramètres d'entrée pour tester des scénarios de mission et quantifier le taux de réussite.

2 Organisation

La simulation a reposé sur un ensemble d'outils imposés par notre client : le logiciel de modélisation 3D Blender, sa surcouche Morse développée par le LAAS

pour la simulation du robot et le gestionnaire de version Gitlab pour organiser le développement du simulateur.

Compte tenu de la difficulté pour définir les éléments nécessaires à la réalisation de l'objectif, nous avons effectué un processus de développement itératif. Ainsi, nous échangeons régulièrement avec notre client sur les fonctions prioritaires de simulateur. Il y a eu deux itérations avec les jalons suivants :

- 2 novembre : plan de développement
- 15 novembre : fin de documentation
- 30 novembre : architecture du code
- 17 décembre : 1^{re} simulation fonctionnelle (1^{re} itération)
- 23 février : 2^e version du simulateur (2^e itération)
- 24 mars : maintenance et remise des livrables

L'échéance du 17 décembre n'a pas pu être tenue suite à un problème d'accès aux ressources en lignes de Sterblue. Nous avons ensuite décalé en conséquence les jalons suivants.

3 Répartition du travail

Une partie des tâches ont été effectuées par l'ensemble du groupe : l'installation des outils de travail et la prise en main des outils. Ensuite, les tâches ont été réparties selon les affinités et les compétences de chacun :

- Mohammed : modélisation de l'aérodynamique
- Adrien : chef de projet, gestion des différentes versions et de la fusion des travaux, modélisation de l'environnement
- Mathieu : modélisation des caméras

- Brendan : modélisation du cardan
- Noha : Gestion du contrôleur, interface entre la simulation et l'autopilote PX4
- Clément : modélisation de la batterie et des moteurs

4 Livrables

Deux livrables sont demandés : le simulateur et son code source commenté. Le simulateur a toutes les fonctionnalités suivantes :

Simulation physique quadrotor et avion : Les collisions avec l'environnement sont gérées. Les calculs de dynamique du vol sont gérés. L'inertie du drone est approximative et non réglable par faute de fonctionnalités suffisantes dans le Blender Game Engine.

Simulation du cardan, support de caméra frontale : La simulation utilise l'affichage auxiliaire de Morse pour montrer le pointage de la caméra principale. Celui-ci est géré suivant un ordre envoyé par Perception ou en appelant un script.

Prise de photos de la caméra frontale : Le simulateur autorise l'enregistrement instantané d'une image de la caméra frontale en vue d'un traitement postérieur. L'ordre est envoyé soit par Perception, soit en appelant un script.

Génération automatique d'un environnement de simulation avec les APIs google : Il est possible de générer, à partir d'un template de base pré-construit, un nouvel environnement 3D focalisé sur une coordonnée géographique. Le script télé-chargera des données d'altitude et de photos satellite et les assem-

blera pour former un fichier Blender exploitable par Morse.

Intégration d'un PX4 en SITL : L'interfaçage MAVlink du simulateur autorise l'insertion d'un firmware de type PX4 au sein de la simulation. Anisi, la partie Perception/PX4 est quasi identique à la version utilisée dans la réalité.

Visualisation 3D temps réel : En supposant la machine suffisamment puissante, la simulation produira une visualisation en temps réel avec suivi du drone dans l'environnement.

Personalisation du Stork : La fichier de configuration à la racine du simulateur permet, en le modifiant, une paramétrisation détaillée du drone.

Contrôle du Stork via un joystick : Un script paramétrable est également fourni pour pouvoir contrôler le Stork dans la simulation avec un joystick.

La documentation est un élément très important pour la poursuite du projet par Sterblue. Ainsi, les codes sources sont commentés, des fichiers expliquants le fonctionnement de chacun des modules existent et nous avons fait un retour écrit sur les difficultés que nous avons rencontrées et la façon dont nous les avons surmontées.

Références

- [1] OPENROBOTS.ORG, *Morse Simulator* : <https://www.openrobots.org/morse/doc/stable/morse.html>