

UNIVERSITÉ DE RENNES 1

MASTER 2 CALCUL SCIENTIFIQUE ET MODÉLISATION
RAPPORT DE STAGE

Etude et développement d'outils mathématiques pour estimer, en temps réel, le tassage et le volume d'un silo de maïs à partir de capteurs embarqués

Auteur :
Sébastien HERVIEU

Tuteur de Stage :
Geoffroy ETAIX

Tuteur Universitaire :
Fabrice MAHÉ

3 août 2018



Todo list

COMPLETER remercier Goeffroy	1
REDIGER Remercier Fongecif	1
REDIGER Remercier Ingeus	1
REDIGER Remercier IDApps, Yann, Stéphane, et surtout Laila!	1
COMPLETER Céline et les filles	1
COMPLETER Completer les remerciements	1
MISE EN FORME : Aérer	1
AMELIORER : L'introduction doit-elle être un chapitre ou alors un Abstract ?	5
REDIGER Introduction du context projet : j'ai effectué le stage blabla au sein de l'entreprise bla etc	5
REDIGER Tellus création, mission, activité	5
AMELIORER : Ce § peut servir à l'intro sur Tellus.	5
Figure : Georadar	5
Figure : Magnetomètre	5
Figure : Radargramme	6
VERIFIER est-ce le bon terme ?	6
Figure : sonar	6
Figure : sonargramme	6
REDIGER Expertise Tellus Traitement des données	6
REDIGER Activité R&D	7
REDIGER Context Projet	7
REDIGER SymeterV1	7
REDIGER Symeter V2 : objectifs	7
REDIGER Décrire le fonctionnement et les fonctionnalités générales d'un lidar .	8
REDIGER Décrire les caractéristiques exactes	8
Figure : Photo du Hukyo	8
REDIGER principe de fonctionnement d'un IMU	9
REDIGER données générées par un IMU	9
Figure : graphes temporels des données générées par l'IMU	9
Figure : Photo de l'IMU xsense	9
REDIGER mode de fonctionnement GPS, précision	9
REDIGER précision centimetrique nécessaire pour SymeterV2	9
REDIGER GPS en mode RTK	9
TROUVER REF EXTERNE : GPS en mode RTK	9
Figure : GPS en mode RTK	9
Figure : Photo base + rover RTK	9
PEUT ETRE caméra video pour odométrie visuelle ?	10
COMPLETER modularité, packages, nodes	10
COMPLETER topics, services	10

■ COMPLETER transformations	10
■ COMPLETER Des modules de références intégrés : PCL et OpenCV entre autre	10
■ OBJECTIF : Introduire les concepts, indiquer qu'ils seront approfondis dans la suite du document.	10
■ REDIGER Coordonnées Homogènes	10
■ REDIGER transformations avec les coordonnées homogènes	10
■ REDIGER Rotations avec les Quaternions	10
■ COMPLETER Ros utilise intensivement les coordonnées homogènes et les quaternions dans les transformations -package transformation.py - et expose uniquement des quaternions pour décrire des rotations dans l'espace 3D	10
■ REDIGER Besoin impératif pour le système Symeter V2 de déterminer sa pose dans le chantier en temps réel, pour pouvoir reconstituer la scène à partir du nuage de points	11
■ REDIGER GPS RTK offre une précision de l'ordre du centimètre mais n'est pas continue, et n'offre pas de précision sur l'assiette du véhicule.	11
■ REDIGER L'IMU peut fournir une information d'accélération linéaire mais bruitée, et requiert une double intégration pour obtenir une position : très sensible au bruit et donc imprécise.	11
■ REDIGER Nous devons donc mettre en place un procédé capable de maintenir un vecteur d'état comportant au moins les positions en x, y, z et en assiette, tangage et lacet.	11
■ REDIGER introduire les filtres de Kalman	11
Figure : chaine de traitement de fusion de données	11
■ REDIGER Reconstituer une scène 3D à partir d'échantillons linéaires	11
■ REDIGER Le LIDAR que nous utilisons est de type linéaire, et collecte énormément de points.	11
■ REDIGER Bande passante importante, traitement potentiellement lourd . . .	11
■ REDIGER Besoin d'accumuler les lignes pour reconstituer un volume	11
■ REDIGER Structure de stockage de donnée 3D	11
■ REDIGER utilisation de Point Cloud Library	11
Figure : Architecture de traitement des nuages de points	11
■ REDIGER pas de silo de test réel à disposition	12
■ REDIGER pas de tracteur à disposition	12
■ REDIGER incertitude sur les possibilités d'implantation des capteurs	12
■ REDIGER incertitude sur le nombre de capteurs nécessaires.	12
■ REDIGER Gazebo environnement de simulation très complet, permet de simuler l'environnement physique dans lequel une plateforme semi-robotique peut évoluer	12
■ REDIGER De nombreux capteurs virtuels disponibles, possibilité de faire varier les incertitudes	12
Figure : Capture d'écran Gazebo avec le tracteur et le silo	12
■ REDIGER Maquette avec équipements réels montés sur une camionnette. . . .	12
■ REDIGER comme il est apparent, projet avec de nombreux sujets, et nous partons de zéro.	13
■ REDIGER objectif est donc de monter une plateforme de base avec dans un premier temps et autant que possible des composants "sur étagère", d'en évaluer les performances et de dégager les axes d'améliorations nécessaires.	13
■ REDIGER Montage de la plateforme simulée ROS + Gazebo	13
■ REDIGER Elaboration, prototypage et test du processus de localisation	13

■ REDIGER Elaboration, prototypage et test du procédé de capture de la scène 3D	13
■ REDIGER Test en grandeur à l'aide d'un équipement monté sur la Tellus Car.	13
■ REDIGER introduction du chapitre simulation tracteur	14
■ REDIGER Présenter succinctement ROS, montrer quelques projets, exposer les outils	14
Figure : Photo d'exemples de robots, roues, bras, humanoïdes, etc	14
■ REDIGER ROS est un environnement permettant le montage de plateformes robotiques complètes.	14
■ REDIGER Architecture logicielle très modulaire	14
■ REDIGER Nodes, services, topics, capteurs, etc	14
Figure : Exemple de node	14
■ REDIGER précision d'évolution d'un plateforme robotique dépend de la prise en compte de la position relative de ses différents capteurs et actionneurs . .	15
■ REDIGER un petit laïus sur les capteurs supportés	15
■ REDIGER Qu'est ce que Gazebo, à quoi ça sert	15
■ REDIGER Décrire les éléments nécessaires à la description du robot	15
■ REDIGER URDF et SDF, world, contrôleurs	15
■ REDIGER Besoin de vérifier la disponibilité et la performance des capteurs cibles de Symeter V2	15
■ REDIGER Lister les problèmes et bugs qui ont nécessité la mise en place de contournements	15
■ REDIGER La nature des "matériaux" utilisés dans gazebo est très "rigide" : les chocs ne sont pas amortis.	15
■ COMPLETER Mise en forme	16
■ Ecrire type de tracteur	16
Figure : Image du tracteur modélisé	16
■ Rédiger Actuateur et Contrôleurs	16
■ Description géométrique du problème de direction différentielle	16
■ Description de l'implémentation sous ROS	16
Figure : Schéma des flux topic qui permet de transformer une commande twist en 4 commandes de vitesses.	16
■ REDIGER Exposer la problématique : fusionner des données avec potentiellement des fréquences d'échantillonnage différentes	17
■ REDIGER déterminer la pose du tracteur à partir des différents capteurs . . .	17
■ REDIGER théorie des filtres de Kalman	17
■ REDIGER pour les systèmes dynamiques non linéaires	17
■ REDIGER Utilisation du module robot_localization	17
■ REDIGER Exposer le plan du chapitre	18
■ REDIGER quelle est la qualité de mesure d'un LIDAR sur une surface potentiellement inclinée ?	18
Figure : Chaîne de traitement LIDAR → Nuage de Point	18
Figure : Transformer un ensemble de lignes en un nuage de points cohérent	18
Figure : Inclinaison LIDAR par rapport au tracteur	18
■ REDIGER Biblio sur l'influence de l'inclinaison du lidar par rapport à la surface	18
■ REDIGER arg1	18
■ REDIGER mettre en forme les données lidar de manière à pouvoir les utiliser .	19
■ REDIGER réduire la bande passante, ne garder que les données exploitables	19
■ REDIGER LIDAR : ensemble d'angles, temps de vol, puissance reçue	19

■ REDIGER Nuage de points : positionnement x,y,z dans la scène	19
■ REDIGER Il est nécessaire de connaître précisément la pose du LIDAR pour générer le nuage de points.	19
Figure : Illustration nature donnée LIDAR	19
Figure : Illustration nature donnée Nuage de Point	19
Figure : Diagramme fonctionnel LIDAR → Nuage de Point	19
■ REDIGER Enjeu : réduire la bande passante, ne pas sur-échantillonner au pied	19
Figure : suréchantillonnage au pied du tracteur, sous-échantillonnage sur les côtés.	19
■ REDIGER Exposer le principe du filtrage	20
■ REDIGER mise en oeuvre grâce à la point cloud library	20
■ REDIGER Objectif : reconstituer la scène à partir des nuages de points "ligne" obtenus à partir du processus d'acquisition	20
■ REDIGER Garder en mémoire les nuages de points correspondant à des lignes précédemment acquise	20
■ REDIGER ne pas dupliquer les informations : éliminer les points doubles . . .	20
■ REDIGER les structures de données de stockage de l'information 3D	20
■ TROUVER REF EXTERNE : les structures de données 3D	20
■ REDIGER Les b-trees, principes	20
■ REDIGER Point forts : calculs performants	20
■ REDIGER points faible : arbre équilibrés → difficile d'ajouter de nouveaux points.	20
■ REDIGER Octree, principes	20
■ REDIGER points forts : structure déséquilibrées sans problème, possibilité d'ajouter de nouveau points avec une bonne performance	20
■ REDIGER point faible : beaucoup d'overhead de memoire si pas implémenté correctement.	20
■ REDIGER Utilisation d'octomap car composant sur étagère	21
■ REDIGER permet d'implémenter rapidement la chaine de traitement pour vé- rifier la validité de la faisabilité	21
■ REDIGER implémentation d'un monde avec un silo et un tas d'ensilage	21
■ REDIGER capture à l'aide d'un lidar monté à l'arrière du tracteur	21
■ REDIGER Tracteur effectue une acquisition initiale de la scène	21
■ REDIGER Une fois l'acquisition effectuée, on invoque le node de sauvegarde de nuage, qui le sauve dans un fichier	21
■ REDIGER Outil de visualisation utilisé : Paraview	21
■ REDIGER besoin de convertir pcd en vtk à l'aide de l'outil approprié	21
■ REDIGER permet de naviguer de manière efficace dans le nuage de point, d'effectuer des projections, etc, etc	21
■ REDIGER Le nuage est épais	21
■ REDIGER octomap génère un nuage de point basé sur le centre des voxel occupés → problème de quantification volumique implique perte de précision de la mesure	21
■ REDIGER Soit réduire la taille du voxel → augmentation de la taille	21
■ REDIGER Soit retourne le point moyen de chaque voxel → non supporté par octomap → à implémenter nous même.	21
■ AJOUTER REF octomap	21
■ REDIGER données sans RTK → besoin d'avoir un	22

Remerciements

Je tiens à remercier les personnes qui m'ont permis de près ou de loin à accomplir ce stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à Messieurs **Fabrice Mahé et Eric Darigand** professeurs de l'Université de Rennes 1, qui m'ont permis de suivre cette formation et qui m'ont accompagné lors de la recherche de stage.

Je tiens à remercier Monsieur **Geoffroy ETAIX**, qui m'a accordé sa confiance pour le stage et pour etc

COMPLETER remercier Goeffroy

REDIGER Remercier Fongecif

REDIGER Remercier Ingeus

REDIGER Remercier IDApps, Yann, Stéphane, et surtout Laila!

Enfin, last but not least, je tiens à remercier mon épouse Céline et mes deux filles pour leur encouragements, leurs soutient et leur patience.

COMPLETER Céline et les filles

COMPLETER Completer les remerciements

MISE EN FORME : Aérer

Table des matières

Remerciements	1
1 Introduction	5
1.1 Tellus Environment - Missions Principales	5
1.1.1 Géophysique et cartographie haute-définition	5
1.1.2 Collecte des données géophysique	5
1.1.3 Traitement des données	6
1.2 Activité R&D	7
1.3 Projet Symeter V2	7
1.3.1 Symeter V1	7
1.3.2 Symeter V2 : Objectifs	7
2 Technologies, Contraintes et Plan de Projet	8
2.1 Capteurs	8
2.1.1 LIDAR Hokuyo UTM-30LX-EW	8
2.1.2 IMU	9
2.1.3 GPS en mode RTK	9
2.2 Environnement de programmation : ROS	10
2.3 Les outils mathématiques	10
2.3.1 Positionnement en Robotique : Poses	10
2.3.2 Localisation par fusion de données	11
2.3.3 Traitement des nuages de points	11
2.4 Contraintes de développement	12
2.4.1 Capacités de tests en grandeur limitées	12
2.4.2 Plateformes de test disponibles	12
2.4.2.1 Environnement de simulation Gazebo	12
2.4.2.2 Prototype monté sur Camionnette	12
2.5 Les grandes parties du projet	13
3 Simulation d'un tracteur évoluant sur un chantier d'ensilage à l'aide de ROS/Gazebo	14
3.1 Présentation de ROS	14
3.1.1 Architecture de ROS	14
3.1.2 Gestion des transformations	15
3.1.3 Gestions des Capteurs	15
3.2 Présentation de Gazebo	15
3.2.1 Construction d'un robot virtuel	15
3.2.2 Vérification de disponibilité des capteurs	15

3.3	Contraintes de mise en oeuvre	15
3.3.1	Quelques bugs gênants	15
3.3.2	Simulation mécanique, frottements, adhérence	15
3.3.3	Conclusions sur les contraintes	15
3.4	Mise en Oeuvre : simulation d'un environnement de tassage de silo	16
3.4.1	Montage d'un tracteur simulé	16
3.4.1.1	Description Physique	16
3.4.1.1.1	Chassis	16
3.4.1.1.2	Actuateurs et Contrôleurs	16
3.4.1.2	Propulsion et Guidage	16
3.4.1.2.1	Algorithme	16
3.4.1.2.2	Implémentation sous ROS	16
4	Mise en place du processus de localisation	17
4.1	Présentation du problème	17
4.2	Filtres de Kalman	17
4.2.1	Filtres de Kalman Linéaires	17
4.2.2	Filtres de Kalman Etendus	17
4.3	Mise en oeuvre	17
5	Exploitation des données LIDAR	18
5.1	Présentation de la chaine de traitement des données LIDAR	18
5.2	Acquisition des données LIDAR	19
5.2.1	Transformation trame LIDAR en un nuage de points	19
5.2.2	Filtrage de la ligne par traitement Voxel	19
5.3	Accumulation des nuages de points	20
5.3.1	Le principe de fonctionnement	20
5.3.2	Principe de stockage des données 3D	20
5.3.2.1	Les B-Trees	20
5.3.2.2	Les Octrees	20
5.3.2.3	Le choix : octree correspond à notre besoin.	21
5.3.3	Mise en oeuvre : octomap	21
5.4	Mise en oeuvre sous Gazebo	21
5.4.1	Test sous gazebo	21
5.4.2	Analyse du nuage de point généré	21
5.4.2.1	Outil pour l'analyse de nuage de points : Paraview	21
5.4.2.2	Points à améliorer sur le nuage de points	21
6	Mise en oeuvre à partir de mesures réelles	22
6.1	Protocol de test	22
6.2	Données générée - visualisation sous google maps	22
6.3	Exploitation des données	22
7	Reste à faire et Améliorations	23
A	Positionnement En Robotique	24
A.1	Géométrie projective, Coordonnées Homogènes	24
A.2	Une autre descriptions des rotations en 3D : Quaternions Unitaires . . .	24

A.3 Application : Simulation de couverture d'un faisceau LIDAR orienté vers le sol	24
B Filtres de Kalman	25
C ROS : Architecture et Concepts	26
D Point Cloud Library	27
Bibliographie	28

Chapitre 1

Introduction

REDIGER Introduction du context projet : j'ai effectué le stage blabla au sein de l'entreprise bla etc

AMELIOR
L'introduction
doit-elle être
chapitre ou
un Abstract

1.1 Tellus Environment - Missions Principales

REDIGER Tellus création, mission, activité

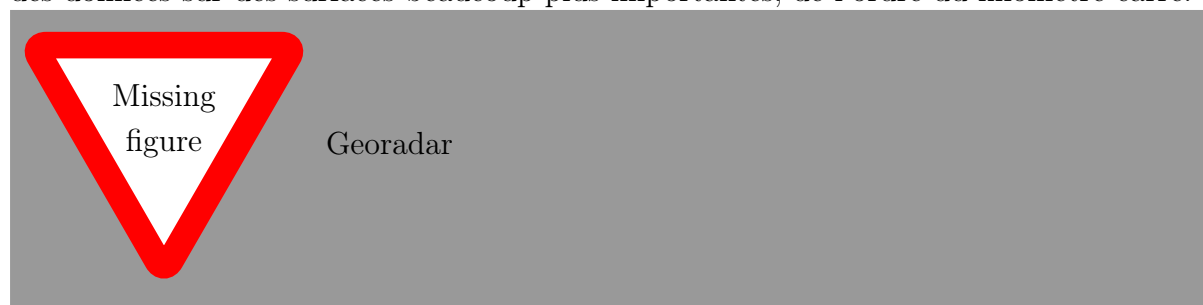
1.1.1 Géophysique et cartographie haute-définition

Tellus Environnement est une startup spécialisée dans la cartographie haute définition des sous-sols et des fond-marins. Elle propose à ses clients une offre bout en bout d'acquisition et de traitement des données pour permettre à ses clients d'agir en fonction de ces conclusions.

AMELIOR
Ce § peut s
l'intro sur T

1.1.2 Collecte des données géophysique

Tellus Environment met en oeuvre ses propres équipements géoradar et magnétomètre pour l'acquisition de données sur le sous-sol sur des surfaces de quelques mètres-carré à quelque hectares. Elle est aussi en mesure de planifier et coordonner la mise en oeuvre d'équipements plus lourds - LIDAR, géoradar, magnétomètres aéroporté - pour obtenir des données sur des surfaces beaucoup plus importantes, de l'ordre du kilomètre carré.





Tellus Environment peut aussi coordonner la mise en oeuvre d'équipements d'acquisition marins - sonar, multibeam, percuteur - pour permettre la cartographie des fonds marins et des sous-sols aquatiques, que ce soit en environnement eau douce - rivières, lacs étangs - ou marins.

VERIFIER
le bon term



Tellus Environment a accès à de nombreuses bases cartographiques pour compléter les données acquises sous sa supervision pour aider à la mise en oeuvre des équipements ainsi qu'à compléter les données acquises en vue de leur traitement.

1.1.3 Traitement des données

REDIGER Expertise Tellus Traitement des données

1.2 Activité R&D

REDIGER Activité R&D

Activité : R&D : développer des composants qui mettent en oeuvre les expertises de TellusEnvironment pour créer des produits innovants, dont les composants seraient de plus réutilisables pour améliorer la productivité des services Géophysique.

1.3 Projet Symeter V2

REDIGER Contexte Projet

1.3.1 Symeter V1

Symeter V1 : rapide rappel

REDIGER SymeterV1

1.3.2 Symeter V2 : Objectifs

REDIGER Symeter V2 : objectifs

Symeter V2 : Objectifs Symeter V2 : Présentation du plan de projet - test du simulateur Gazebo pour évaluer son utilité dans le projet Symeter2 - Simulation couverture lidar - Montage des outils nécessaires au développement simulé du projet symeter - Mise en place de la localisation : installation et tests - Mise en place de l'acquisition des relevés.

Chapitre 2

Technologies, Contraintes et Plan de Projet

Ce chapitre présente dans leurs grandes lignes les entrants et les contraintes mis en jeu par le projet Symeter V2. On y énumère notamment les différents éléments à mettre en oeuvre pour monter le système Symeter V2 : les capteurs l'environnement d'exploitation logicielle, les outils mathématiques et procédés à coordonner pour monter le prototype de Symeter 2.

2.1 Capteurs

Les capteurs à mettre en oeuvre seront un ou plusieurs LIDAR, un ou plusieurs IMU et un ou plusieurs GPS. Cette section décrit les fonctionnalités de ces capteurs de ces capteurs.

2.1.1 LIDAR Hokuyo UTM-30LX-EW

Le Hokuyo UTM-30-LX-EW est le type de LIDAR qui était utilisé dans Symeter V1, et sera donc ré-utilisé pour Symeter V2. C'est un équipement robuste et précis qui est de plus bien connu par Tellus Environment.

Cet équipement est basé sur un laser infrarouge (longueur d'onde de 905nm) pour scanner un champ semi-circulaire de 270°, par pas de 0,25°. Il mesure la distances des objets à sa portée pour chaque incrément. La distance de détection maximale est de 30m.

Angle de Balayage	270°
Résolution Angulaire	env. 0,25°
Temps de Balayage	25ms/balayage (40 balayages par seconde)
Distance de détection	Portée garantie : 0,1 à 30m
Résolution de la mesure	1mm σ

REDIGER Décrire le fonctionnement et les fonctionnalités générales d'un lidar

REDIGER Décrire les caractéristiques exactes

Missing figure

Photo du Hukyo

2.1.2 IMU

REDIGER principe de fonctionnement d'un IMU

REDIGER données générées par un IMU

Missing figure

graphes temporels des données générées par l'IMU

Missing figure

Photo de l'IMU xsense

2.1.3 GPS en mode RTK

REDIGER mode de fonctionnement GPS, précision

REDIGER précision centimétrique nécessaire pour SymeterV2

REDIGER GPS en mode RTK

TROUVER REF EXTERNE : GPS en mode RTK

Missing figure

GPS en mode RTK




Photo base + rover RTK

PEUT ETRE caméra video pour odométrie visuelle ?

2.2 Environnement de programmation : ROS

Du fait de son utilisation dans la première version de Symeter, et du fait de ses qualités en terme de modularité, de disponibilité des drivers pour les capteurs et actionneurs, l'environnement logiciel ROS a été choisi avant même le début du stage pour être la base de la partie logicielle de Symeter V2.

Il s'agit d'un environnement logiciel destiné à la mise en oeuvre de systèmes robotiques qui comporte toutes les briques de bases nécessaires au montage de tels systèmes.

COMPLETER modularité, packages, nodes

COMPLETER topics, services

COMPLETER transformations

COMPLETER Des modules de références intégrés : PCL et OpenCV entre autre

2.3 Les outils mathématiques

OBJECTIF : Introduire les concepts, indiquer qu'ils seront approfondis dans la suite du document.

2.3.1 Positionnement en Robotique : Poses

REDIGER Coordonnées Homogènes

REDIGER transformations avec les coordonnées homogènes

REDIGER Rotations avec les Quaternions

COMPLETER Ros utilise intensivement les coordonnées homogènes et les quaternions dans les transformations -package transformation.py - et expose uniquement des quaternions pour décrire des rotations dans l'espace 3D

. Positionnement en Robotique : Géometrie projective, Coordonnées Homogènes, Quaternions

2.3.2 Localisation par fusion de données

REDIGER Besoin impératif pour le système Symeter V2 de déterminer sa pose dans le chantier en temps réel, pour pouvoir reconstituer la scène à partir du nuage de points

REDIGER GPS RTK offre une précision de l'ordre du centimètre mais n'est pas continue, et n'offre pas de précision sur l'assiette du véhicule.

REDIGER L'IMU peut fournir une information d'accélération linéaire mais bruitée, et requiert une double intégration pour obtenir une position : très sensible au bruit et donc imprécise.

REDIGER Nous devons donc mettre en place un procédé capable de maintenir un vecteur d'état comportant au moins les positions en x, y, z et en assiette, tangage et lacet.

REDIGER introduire les filtres de Kalman



chaîne de traitement de fusion de données

2.3.3 Traitement des nuages de points

REDIGER Reconstituer une scène 3D à partir d'échantillons linéaires

REDIGER Le LIDAR que nous utilisons est de type linéaire, et collecte énormément de points.

REDIGER Bande passante importante, traitement potentiellement lourd

REDIGER Besoin d'accumuler les lignes pour reconstituer un volume

REDIGER Structure de stockage de donnée 3D

REDIGER utilisation de Point Cloud Library



Architecture de traitement des nuages de points

2.4 Contraintes de développement

2.4.1 Capacités de tests en grandeur limitées

REDIGER pas de silo de test réel à disposition

REDIGER pas de tracteur à disposition

REDIGER incertitude sur les possibilités d'implantation des capteurs

REDIGER incertitude sur le nombre de capteurs nécessaires.

Les équipements à mettre en oeuvre sont relativement coûteux et leur mise en oeuvre requiert une certaine expertise.

De plus la mise en silo du maïs n'intervient qu'à de courtes périodes au cours de l'année. Il est donc nécessaire de pouvoir mettre en oeuvre le développement du système par le biais de mises en oeuvre alternatives, à savoir la simulation et la mise en oeuvre en grandeur des équipements pour effectuer des tests simples de reconstitution du terrain.

2.4.2 Plateformes de test disponibles

2.4.2.1 Environnement de simulation Gazebo

REDIGER Gazebo environnement de simulation très complet, permet de simuler l'environnement physique dans lequel une plateforme semi-robotique peut évoluer

REDIGER De nombreux capteurs virtuels disponibles, possibilité de faire varier les incertitudes



Capture d'écran Gazebo avec le tracteur et le silo

2.4.2.2 Prototype monté sur Camionnette

REDIGER Maquette avec équipements réels montés sur une camionnette.

ROS est fourni avec un environnement de simulation robotique très intégré nommé "Gazebo", qui permet de simuler en temps réel le comportement mécanique de modèles de robots. Ce logiciel est donc utilisé pour simuler un environnement de Silo pour tester la localisation, l'acquisition du terrain, la mesure d'un modèle de Silo.

Pour la mise en oeuvre en grandeur, un prototype de l'équipement a été monté sur la camionnette de Tellus Environnement, la "Tellus Car".

2.5 Les grandes parties du projet

REDIGER comme il est apparent, proeijt avec de nombreux sujets, et nous partons de zéro.

REDIGER objectif est donc de monter une plateforme de base avec dans un premier temps et autant que possible des composants "sur étagère", d'en évaluer les performances et de dégager les axes d'améliorations nécessaires.

REDIGER Montage de la plateforme simulée ROS + Gazebo

REDIGER Elaboration, prototypage et test du processus de localisation

REDIGER Elaboration, prototypage et test du procédé de capture de la scène 3D

REDIGER Test en grandeur à l'aide d'un équipement monté sur la Tellus Car.

Chapitre 3

Simulation d'un tracteur évoluant sur un chantier d'ensilage à l'aide de ROS/Gazebo

REDIGER introduction du chapitre simulation tracteur

3.1 Présentation de ROS

REDIGER Présenter succinctement ROS, montrer quelques projets, exposer les outils



Photo d'exemples de robots, roues, bras, humanoïdes, etc

3.1.1 Architecture de ROS

REDIGER ROS est un environnement permettant le montage de plateformes robotiques complètes.

REDIGER Architecture logicielle très modulaire

REDIGER Nodes, services, topics, capteurs, etc



Exemple de node

3.1.2 Gestion des transformations

REDIGER précision d'évolution d'un plateforme robotique dépend de la prise en compte de la position relative de ses différents capteurs et actionneurs

3.1.3 Gestions des Capteurs

REDIGER un petit laïus sur les capteurs supportés

3.2 Présentation de Gazebo

REDIGER Qu'est ce que Gazebo, à quoi ça sert

3.2.1 Construction d'un robot virtuel

REDIGER Décrire les éléments nécessaires à la description du robot

REDIGER URDF et SDF, world, contrôleurs

3.2.2 Vérification de disponibilité des capteurs

REDIGER Besoin de vérifier la disponibilité et la performance des capteurs cibles de Symeter V2

3.3 Contraintes de mise en oeuvre

3.3.1 Quelques bugs gênants

REDIGER Lister les problèmes et bugs qui ont nécessité la mise en place de contournements

3.3.2 Simulation mécanique, frottements, adhérence

REDIGER La nature des "matériaux" utilisés dans gazebo est très "rigide" : les chocs ne sont pas amortis.

3.3.3 Conclusions sur les contraintes

Eviter à tout prix que le véhicule entre en dérapage : il ne récupère généralement pas. Raisons pour lesquelles le véhicule peut entrer en dérapage : - mauvaise configuration des paramètres de frottement des éléments en contact avec le sol. - vitesses de rotation différentes des roues : - mauvaise consigne de vitesse sur les roues - différence de vitesse entre roue intérieure et roue extérieure lors d'un virage - en virage : rayons de virages différent entre roue intérieure et roue extérieure

Conduite en terrain accidenté : - Conserver la directivité - 4 roues motrices et/ou suspension

COMPLETER Mise en forme

3.4 Mise en Oeuvre : simulation d'un environnement de tassage de silo

3.4.1 Montage d'un tracteur simulé

Intérêt : simuler l'implantation physique des capteurs avec des dimensions du même ordre de grandeur que les plateformes cibles.

3.4.1.1 Description Physique

3.4.1.1.1 Chassis Le chassis du tracteur à été monté en se basant sur les dimensions générales d'un tracteur de type **TODO : ICI TYPE DE TRACTEUR** , en utilisant cependant des formes géométriques simplifiées.

Ecrire type tracteur

Le tracteur est donc composé d'un pavé en guise de chassis, de 4 cylindres allongés en guise d'essieux, et de 4 grands cylindres en guise de roues.



3.4.1.1.2 Actuateurs et Contrôleurs

Rédiger Act et Contrôle

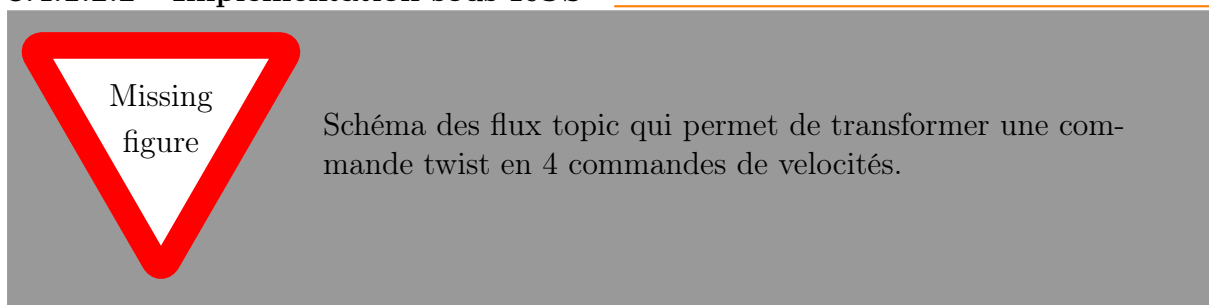
3.4.1.2 Propulsion et Guidage

3.4.1.2.1 Algorithme

Description géométrique du problème de direction différentielle

3.4.1.2.2 Implémentation sous ROS

Description l'implément sous ROS



Chapitre 4

Mise en place du processus de localisation

4.1 Présentation du problème

REDIGER Exposer la problématique : fusionner des données avec potentiellement des fréquences d'échantillonnage différentes

REDIGER déterminer la pose du tracteur à partir des différents capteurs

4.2 Filtres de Kalman

REDIGER théorie des filtres de Kalman

4.2.1 Filtres de Kalman Linéaires

4.2.2 Filtres de Kalman Etendus

REDIGER pour les systèmes dynamiques non linéaires

4.3 Mise en oeuvre

REDIGER Utilisation du module robot_localization

Chapitre 5

Exploitation des données LIDAR

REDIGER Exposer le plan du chapitre

REDIGER quelle est la qualité de mesure d'un LIDAR sur une surface potentiellement inclinée ?

5.1 Présentation de la chaine de traitement des données LIDAR



Chaine de traitement LIDAR -> Nuage de Point



Transformer un ensemble de lignes en un nuage de points cohérent



Inclinaison LIDAR par rapport au tracteur

REDIGER Biblio sur l'influence de l'inclinaison du lidar par rapport à la surface

REDIGER arg1

5.2 Acquisition des données LIDAR

REDIGER mettre en forme les données lidar de manière à pouvoir les utiliser

REDIGER réduire la bande passante, ne garder que les données exploitables

5.2.1 Transformation trame LIDAR en un nuage de points

REDIGER LIDAR : ensemble d'angles, temps de vol, puissance reçue

REDIGER Nuage de points : positionnement x,y,z dans la scène

REDIGER Il est nécessaire de connaître précisément la pose du LIDAR pour générer le nuage de points.



Illustration nature donnée LIDAR




Illustration nature donnée Nuage de Point



Diagramme fonctionnel LIDAR -> Nuage de Point

5.2.2 Filtrage de la ligne par traitement Voxel

REDIGER Enjeu : réduire la bande passante, ne pas sur-échantillonner au pied



Missing figure

suréchantillonnage au pied du tracteur, sous-échantillonnage sur les côtés.

REDIGER Exposer le principe du filtrage

REDIGER mise en oeuvre grace à la point cloud library

5.3 Accumulation des nuages de points

REDIGER Objectif : reconstituer la scène à partir des nuages de points "ligne" obtenus à partir du processus d'acquisition

5.3.1 Le principe de fonctionnement

REDIGER Garder en mémoire les nuages de points correspondant à des lignes précédemment acquise

REDIGER ne pas dupliquer les informations : éliminer les points doubles

5.3.2 Principe de stockage des données 3D

REDIGER les structures de données de stockage de l'information 3D

TROUVER REF EXTERNE : les structures de données 3D

5.3.2.1 Les B-Trees

REDIGER Les b-trees, principes

REDIGER Point forts : calculs performants

REDIGER points faible : arbre équilibrés -> difficile d'ajouter de nouveaux points.

5.3.2.2 Les Octrees

REDIGER Octree, principes

REDIGER points forts : structure déséquilibrées sans problème, possibilité d'ajouter de nouveau points avec une bonne performance

REDIGER point faible : beaucoup d'overhead de memoire si pas implémenté correctement.

5.3.2.3 Le choix : octree correspond à notre besoin.

5.3.3 Mise en oeuvre : octomap

REDIGER Utilisation d'octomap car composant sur étagère

REDIGER permet d'implémenter rapidement la chaîne de traitement pour vérifier la validité de la faisabilité

5.4 Mise en oeuvre sous Gazebo

REDIGER implémentation d'un monde avec un silo et un tas d'ensilage

REDIGER capture à l'aide d'un lidar monté à l'arrière du tracteur

5.4.1 Test sous gazebo

REDIGER Tracteur effectue une acquisition initiale de la scène

REDIGER Une fois l'acquisition effectuée, on invoque le node de sauvegarde de nuage, qui le sauve dans un fichier

5.4.2 Analyse du nuage de point généré

5.4.2.1 Outil pour l'analyse de nuage de points : Paraview

REDIGER Outil de visualisation utilisé : Paraview

REDIGER besoin de convertir pcd en vtk à l'aide de l'outil approprié

REDIGER permet de naviguer de manière efficace dans le nuage de point, d'effectuer des projections, etc, etc

5.4.2.2 Points à améliorer sur le nuage de points

REDIGER Le nuage est épais

REDIGER octomap génère un nuage de point basé sur le centre des voxel occupés -> problème de quantification volumique implique perte de précision de la mesure

REDIGER Soit réduire la taille du voxel -> augmentation de la taille

REDIGER Soit retourner le point moyen de chaque voxel -> non supporté par octomap -> à implémenter nous même.

AJOUTER
octomap

Chapitre 6

Mise en oeuvre à partir de mesures réelles

6.1 Protocol de test

6.2 Données générée - visualisation sous google maps

6.3 Exploitation des données

REDIGER données sans RTK -> besoin d'avoir un

Chapitre 7

Reste à faire et Améliorations

Annexe A

Positionnement En Robotique

- A.1 Géométrie projective, Coordonnées Homogènes
- A.2 Une autre descriptions des rotations en 3D : Quaternions Unitaires
- A.3 Application : Simulation de couverture d'un faisceau LIDAR orienté vers le sol

Annexe B

Filtres de Kalman

Annexe C

ROS : Architecture et Concepts

Annexe D

Point Cloud Library

Essai de citation [KH06].

Bibliographie

- [KH06] Elliott D. Kaplan and C. Hegarty, editors. *Understanding GPS : principles and applications*. Artech House mobile communications series. Artech House, Boston, 2nd ed edition, 2006. OCLC : ocm62128065.
- [RS12] TAMAKI RS. Scanning Laser Range Finder UTM-30lx-EW Specification. page 7, December 2012.
- [ZM09] Paul Zarchan and Howard Musoff. *Fundamentals of Kalman filtering : a practical approach*. Number v. 232 in Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, Reston, Va, 3rd ed edition, 2009. OCLC : ocn457170744.