

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

8 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 |
| Figure : mode 2D vs mode 3D | 9 |
| ■ REDIGER Décrire les caractéristiques exactes | 10 |
| Figure : Photo du Hukyo | 10 |
| Figure : Trajectoire et mesures | 10 |
| ■ REDIGER données générées par un IMU | 11 |
| Figure : graphes temporels des données générées par l'IMU | 11 |
| Figure : Photo de l'IMU xsense | 11 |
| Figure : GPS en mode simple | 12 |
| Figure : GPS en mode RTK | 13 |
| ■ TROUVER REF EXTERNE : GPS en mode RTK | 13 |
| Figure : Photo base + rover RTK | 13 |
| ■ PEUT ETRE caméra video pour odométrie visuelle ? | 13 |
| Figure : Quelques robots powered by ROS | 13 |
| Figure : Exemple d'une structure en packages, noden topics et services | 14 |
| ■ OBJECTIF : Introduire les concepts, indiquer qu'ils seront approfondis dans la suite du document. | 14 |

| | |
|--|----|
| Figure : Pose 3D d'un objet | 15 |
| ■ AJOUTER REF Lien vers l'annexe | 16 |
| ■ AJOUTER REF Quaternion et quaternions unité | 16 |
| Figure : Flux de données IMU + GPS $\rightarrow X$ | 16 |
| ■ REDIGER introduire les filtres de Kalman | 17 |
| Figure : chaine de traitement de fusion de données | 17 |
| ■ AJOUTER REF Filter de Kalman pour la fusion de données. | 17 |
| ■ COMPLETER Comme le GPS fourni une position absolue dans le référentiel de la scène nous ne sommes pas dans une problématique SLAM. Le véhicule n'est pas autonome, la navigation est assurée par l'opérateur du tracteur. . . | 17 |
| Figure : Architecture de traitement des nuages de points | 17 |
| ■ REDIGER De nombreux capteurs virtuels disponibles, possibilité de faire varier les incertitudes | 18 |
| Figure : Capture d'écran Gazebo avec le tracteur et le silo | 18 |
| Figure : Maquette avec équipements réels montés sur une camionnette. | 19 |
| ■ REDIGER comme il est apparent, proejt avec de nombreux sujets, et nous partons de zéro. | 19 |
| ■ 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. | 19 |
| ■ REDIGER Montage de la plateforme simulée ROS + Gazebo | 19 |
| ■ REDIGER Elaboration, prototypage et test du processus de localisation | 19 |
| ■ REDIGER Elaboration, prototypage et test du procédé de capture de la scène 3D | 19 |
| ■ REDIGER Test en grandeur à l'aide d'un équipement monté sur la Tellus Car. | 19 |
| ■ REDIGER introduction du chapitre simulation tracteur | 20 |
| ■ REDIGER Présenter succinctement ROS, montrer quelques projets, exposer les outils | 20 |
| Figure : Photo d'exemples de robots, roues, bras, humanoïdes, etc | 20 |
| ■ REDIGER ROS est un environnement permettant le montage de plateformes robotiques complètes. | 20 |
| ■ REDIGER Architecture logicielle très modulaire | 20 |
| ■ REDIGER Nodes, services, topics, capteurs, etc | 20 |
| Figure : Exemple de node | 20 |
| ■ REDIGER précision d'évolution d'un plateforme robotique dépend de la prise en compte de la position relative de ses différents capteurs et actionneurs . . | 21 |
| ■ REDIGER un petit laïus sur les capteurs supportés | 21 |
| ■ REDIGER Qu'est ce que Gazebo, à quoi ça sert | 21 |
| ■ REDIGER Décrire les éléments nécessaires à la description du robot | 21 |
| ■ REDIGER URDF et SDF, world, contrôleurs | 21 |
| ■ REDIGER Besoin de vérifier la disponibilité et la performance des capteurs cibles de Symeter V2 | 21 |
| ■ REDIGER Lister les problèmes et bugs qui ont nécessité la mise en place de contournements | 21 |
| ■ REDIGER La nature des "matériaux" utilisés dans gazebo est très "rigide" : les chocs ne sont pas amortis. | 21 |
| ■ COMPLETER Mise en forme | 22 |
| ■ Ecrire type de tracteur | 22 |
| Figure : Image du tracteur modélisé | 22 |

| | |
|---|----|
| ■ Rédiger Actuateur et Contrôleurs | 22 |
| ■ Description géométrique du problème de direction différentielle | 22 |
| ■ Description de l'implémentation sous ROS | 22 |
| Figure : Schéma des flux topic qui permet de transformer une commande twist en 4 commandes de velocities. | 22 |
| ■ REDIGER Exposer la problématique : fusionner des données avec potentiellement des fréquences d'échantillonnage différentes | 23 |
| ■ REDIGER déterminer la pose du tracteur à partir des différents capteurs . . . | 23 |
| ■ REDIGER théorie des filtres de Kalman | 23 |
| ■ REDIGER pour les systèmes dynamiques non linéaires | 23 |
| ■ REDIGER Utilisation du module robot_localization | 23 |
| ■ REDIGER Exposer le plan du chapitre | 24 |
| ■ REDIGER quelle est la qualité de mesure d'un LIDAR sur une surface potentiellement inclinée? | 24 |
| Figure : Chaîne de traitement LIDAR → Nuage de Point | 24 |
| Figure : Transformer un ensemble de lignes en un nuage de points cohérent | 24 |
| Figure : Inclinaison LIDAR par rapport au tracteur | 24 |
| ■ REDIGER Biblio sur l'influence de l'inclinaison du lidar par rapport à la surface | 24 |
| ■ REDIGER arg1 | 24 |
| ■ REDIGER mettre en forme les données lidar de manière à pouvoir les utiliser . | 25 |
| ■ REDIGER réduire la bande passante, ne garder que les données exploitables | 25 |
| ■ REDIGER LIDAR : ensemble d'angles, temps de vol, puissance reçue | 25 |
| ■ REDIGER Nuage de points : positionnement x,y,z dans la scène | 25 |
| ■ REDIGER Il est nécessaire de connaître précisément la pose du LIDAR pour générer le nuage de points. | 25 |
| Figure : Illustration nature donnée LIDAR | 25 |
| Figure : Illustration nature donnée Nuage de Point | 25 |
| Figure : Diagramme fonctionnel LIDAR → Nuage de Point | 25 |
| ■ REDIGER Enjeu : réduire la bande passante, ne pas sur-échantillonner au pied | 25 |
| Figure : suréchantillonnage au pied du tracteur, sous-échantillonnage sur les côtés. | 25 |
| ■ REDIGER Exposer le principe du filtrage | 26 |
| ■ REDIGER mise en oeuvre grâce à la point cloud library | 26 |
| ■ REDIGER Objectif : reconstituer la scène à partir des nuages de points "ligne" obtenus à partir du processus d'acquisition | 26 |
| ■ REDIGER Garder en mémoire les nuages de points correspondant à des lignes précédemment acquise | 26 |
| ■ REDIGER ne pas dupliquer les informations : éliminer les points doubles . . . | 26 |
| ■ REDIGER les structures de données de stockage de l'information 3D | 26 |
| ■ TROUVER REF EXTERNE : les structures de données 3D | 26 |
| ■ REDIGER Les b-trees, principes | 26 |
| ■ REDIGER Point forts : calculs performants | 26 |
| ■ REDIGER points faible : arbre équilibrés → difficile d'ajouter de nouveaux points. | 26 |
| ■ REDIGER Octree, principes | 26 |
| ■ REDIGER points forts : structure déséquilibrées sans problème, possibilité d'ajouter de nouveau points avec une bonne performance | 26 |
| ■ REDIGER point faible : beaucoup d'overhead de memoire si pas implémenté correctement. | 26 |

| | |
|--|----|
| ■ REDIGER Utilisation d'octomap car composant sur étagère | 27 |
| ■ REDIGER permet d'implémenter rapidement la chaîne de traitement pour vérifier la validité de la faisabilité | 27 |
| ■ REDIGER implémentation d'un monde avec un silo et un tas d'ensilage | 27 |
| ■ REDIGER capture à l'aide d'un lidar monté à l'arrière du tracteur | 27 |
| ■ REDIGER Tracteur effectue une acquisition initiale de la scène | 27 |
| ■ REDIGER Une fois l'acquisition effectuée, on invoque le node de sauvegarde de nuage, qui le sauve dans un fichier | 27 |
| ■ REDIGER Outil de visualisation utilisé : Paraview | 27 |
| ■ REDIGER besoin de convertir pcd en vtk à l'aide de l'outil approprié | 27 |
| ■ REDIGER permet de naviguer de manière efficace dans le nuage de point, d'effectuer des projections, etc, etc | 27 |
| ■ REDIGER Le nuage est épais | 27 |
| ■ REDIGER octomap génère un nuage de point basé sur le centre des voxels occupés -> problème de quantification volumique implique perte de précision de la mesure | 27 |
| ■ REDIGER Soit réduire la taille du voxel -> augmentation de la taille | 27 |
| ■ REDIGER Soit retourner le point moyen de chaque voxel -> non supporté par octomap -> à implémenter nous même. | 27 |
| ■ AJOUTER REF octomap | 27 |
| ■ REDIGER données sans RTK -> besoin d'avoir un | 28 |

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 | 8 |
| 2.1.1.1 Détecteur Actif : | 8 |
| 2.1.1.2 Scan de l'environnement | 9 |
| 2.1.1.3 LIDAR Hokuyo UTM-30LX-EW | 9 |
| 2.1.2 IMU | 10 |
| 2.1.2.1 Les grandeurs mesurées | 10 |
| 2.1.2.2 IMU XSense | 11 |
| 2.1.3 GPS en mode RTK | 12 |
| 2.1.3.1 GPS simple | 12 |
| 2.1.3.2 Principe du mode RTK | 12 |
| 2.2 Environnement de programmation : ROS | 13 |
| 2.3 Les outils mathématiques | 14 |
| 2.3.1 Positionnement en Robotique : Poses | 15 |
| 2.3.2 Localisation par fusion de données | 16 |
| 2.3.3 Traitement des nuages de points | 17 |
| 2.4 Contraintes de développement | 18 |
| 2.4.1 Capacités de tests en grandeur limitées | 18 |
| 2.4.2 Plateformes de test disponibles | 18 |
| 2.4.2.1 Environnement de simulation Gazebo | 18 |
| 2.4.2.2 Prototype monté sur Camionnette | 19 |
| 2.5 Les grandes parties du projet | 19 |

| | | |
|-----------|--|-----------|
| 3 | Simulation d'un tracteur évoluant sur un chantier d'ensilage à l'aide de ROS/Gazebo | 20 |
| 3.1 | Présentation de ROS | 20 |
| 3.1.1 | Architecture de ROS | 20 |
| 3.1.2 | Gestion des transformations | 21 |
| 3.1.3 | Gestions des Capteurs | 21 |
| 3.2 | Présentation de Gazebo | 21 |
| 3.2.1 | Construction d'un robot virtuel | 21 |
| 3.2.2 | Vérification de disponibilité des capteurs | 21 |
| 3.3 | Contraintes de mise en oeuvre | 21 |
| 3.3.1 | Quelques bugs gênants | 21 |
| 3.3.2 | Simulation mécanique, frottements, adhérence | 21 |
| 3.3.3 | Conclusions sur les contraintes | 21 |
| 3.4 | Mise en Oeuvre : simulation d'un environnement de tassage de silo | 22 |
| 3.4.1 | Montage d'un tracteur simulé | 22 |
| 3.4.1.1 | Description Physique | 22 |
| 3.4.1.1.1 | Chassis | 22 |
| 3.4.1.1.2 | Actuateurs et Contrôleurs | 22 |
| 3.4.1.2 | Propulsion et Guidage | 22 |
| 3.4.1.2.1 | Algorithme | 22 |
| 3.4.1.2.2 | Implémentation sous ROS | 22 |
| 4 | Mise en place du processus de localisation | 23 |
| 4.1 | Présentation du problème | 23 |
| 4.2 | Filtres de Kalman | 23 |
| 4.2.1 | Filtres de Kalman Linéaires | 23 |
| 4.2.2 | Filtres de Kalman Etendus | 23 |
| 4.3 | Mise en oeuvre | 23 |
| 5 | Exploitation des données LIDAR | 24 |
| 5.1 | Présentation de la chaine de traitement des données LIDAR | 24 |
| 5.2 | Acquisition des données LIDAR | 25 |
| 5.2.1 | Transformation trame LIDAR en un nuage de points | 25 |
| 5.2.2 | Filtrage de la ligne par traitement Voxel | 25 |
| 5.3 | Accumulation des nuages de points | 26 |
| 5.3.1 | Le principe de fonctionnement | 26 |
| 5.3.2 | Principe de stockage des données 3D | 26 |
| 5.3.2.1 | Les B-Trees | 26 |
| 5.3.2.2 | Les Octrees | 26 |
| 5.3.2.3 | Le choix : octree correspond à notre besoin. | 27 |
| 5.3.3 | Mise en oeuvre : octomap | 27 |
| 5.4 | Mise en oeuvre sous Gazebo | 27 |
| 5.4.1 | Test sous gazebo | 27 |
| 5.4.2 | Analyse du nuage de point généré | 27 |
| 5.4.2.1 | Outil pour l'analyse de nuage de points : Paraview . . . | 27 |
| 5.4.2.2 | Points à améliorer sur le nuage de points | 27 |

| | | |
|----------|---|-----------|
| 6 | Mise en oeuvre à partir de mesures réelles | 28 |
| 6.1 | Protocol de test | 28 |
| 6.2 | Données générée - visualisation sous google maps | 28 |
| 6.3 | Exploitation des données | 28 |
| 7 | Reste à faire et Améliorations | 29 |
| A | Positionnement En Robotique | 30 |
| A.1 | Géometrie projective, Coordonnées Homogènes | 30 |
| A.2 | Une autre descriptions des rotations en 3D : Quaternions Unitaires . . . | 30 |
| A.3 | Application : Simulation de couverture d'un faisceau LIDAR orienté vers le sol | 30 |
| B | Filtres de Kalman | 31 |
| C | ROS : Architecture et Concepts | 32 |
| D | Point Cloud Library | 33 |
| | Bibliographie | 34 |

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 un 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'introduire sur

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.



Georadar



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

Un LIDAR est un équipement qui permet de prendre de nombreuses mesures de l'environnement à partir d'un laser. En exploitant les informations d'angle de pointage et en observant les caractéristiques du signal réfléchi tant du point de vue temporel (temps de vol, phase) que du point de vue de la forme du signal (puissance réfléchie, déphasage, distorsion, effet Doppler , ...), il est possible de mesurer de nombreuses caractéristiques de l'environnement.

Le LIDAR est utilisé par Symeter V2 pour détecter la forme précise du chantier d'ensilage.

2.1.1.1 Détecteur Actif :

Le LIDAR est essentiellement un détecteur actif dont le principe est identique à celui du radar : une pulse électromagnétique est émise dans une direction privilégiée par une partie émettrice, constituant un signal incident.

Ce signal incident est réfléchi par un objet de l'environnement, générant un signal réfléchi qui retourne en direction du LIDAR. Ce signal réfléchi est reçu par un détecteur.

En comparant le signal émis et le signal reçu nous pouvons en déduire certaines caractéristiques de l'environnement.

Différents types de mesure : Il existe deux grands types de LIDAR :

LIDAR à "temps de vol" : les lidars de ce type déterminent la distance de la cible en mesurant le temps écoulé entre le moment de l'émission du pulse et la réception du signal réfléchi par la cible.

LIDAR "full waveform" : les LIDAR de ce type émettent un signal périodique continu et mesurent la distance de l'objet détecté essentiellement par mesure de déphasage du signal réfléchi par rapport au signal émis. Les modifications de la forme du signal réfléchi et de son intensité permettent d'extraire d'autres informations à propos de la cible.

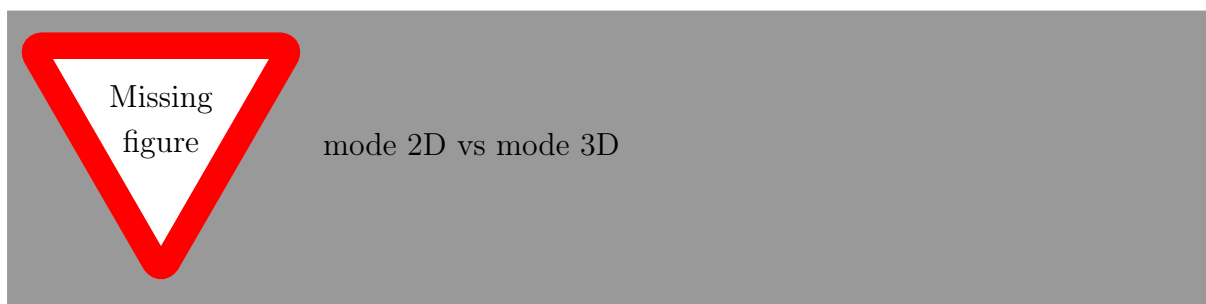
2.1.1.2 Scan de l'environnement

En faisant varier la direction du bloc émetteur/récepteur du LIDAR, il est possible de faire plusieurs mesures de l'environnement. Le LIDAR mettant en oeuvre des procédés optiques, il est possible de faire varier très rapidement son pointage et ainsi d'obtenir de très nombreux relevés de l'environnement.

Deux grands modes de scan de l'environnement : Il existe deux grands modes de scan de l'environnement pour un LIDAR

Mode 2D ou "planar" : Dans ce mode le LIDAR mesure l'environnement dans un plan unique, généralement en faisant tourner le faisceau de manière circulaire autour d'un axe.

Mode 3D : Dans ce mode le LIDAR mesure des portions d'espace, généralement en combinant deux rotations simulées du faisceau, soit en faisant tourner sur un même axe de révolution plusieurs faisceaux.



2.1.1.3 LIDAR Hokuyo UTM-30LX-EW

Le projet Symeter V2 met en oeuvre un LIDAR de type Hokuyo UTM-30-LX-EW, le même que celui utilisé par Symeter V1. 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°. C'est un LIDAR de type "planar, temps de vol".

Il mesure la distances des objets à sa portée par pas de $0,25^\circ$. La distance de détection maximale est de 30m.

| | |
|-------------------------|--|
| Angle de Balayage | 270° |
| Résolution Angulaire | env. $0,25^\circ$ |
| 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 les caratéristiques exactes



Photo du Hukyo

2.1.2 IMU

Un IMU - Inertial Measurement Unit - est un équipement comportant plusieurs capteurs inertiels, d'accélération et d'angles qui permettent avec un traitement des données approprié d'établir de suivre la pose du véhicule sur lequel il est monté.

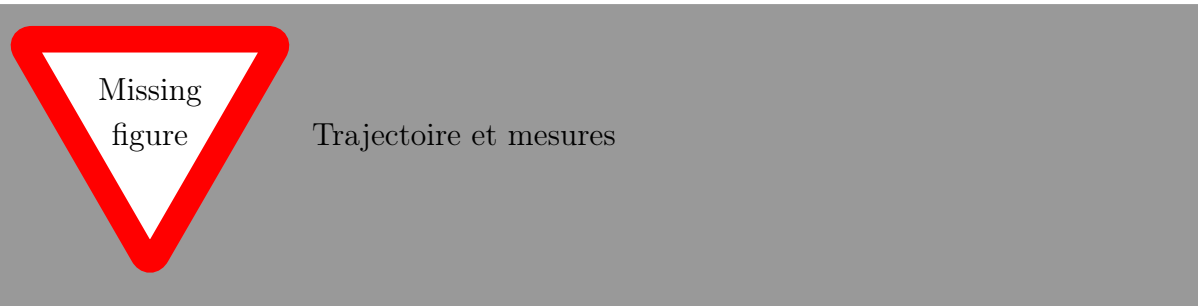
2.1.2.1 Les grandeurs mesurées

Un IMU comporte généralement 3 accéléromètres linéaires et un gyroscope, et souvent un magnétomètre.

Le **gyroscope** permet de mesurer, avec précision et en continu, l'inclinaison du véhicule en terme de **roulis**, **tanguage** et **lacet**.

Les **accéléromètres** mesurent en continu les accélérations linéaires de la partie du véhicule sur lequel l'IMU est fixé, dans trois directions différentes. Ces directions sont généralement dénotées x, y, z , par analogie avec le repère cartésien. Simplement, les accélérations sont mesurées dans le repère du véhicule, mobile et variable et non dans un repère absolu immobile.

En combinant les variations d'inclinaison et les mesures d'accélérations linéaires dans le repère mobile du véhicule, il est possible de reconstruire avec une certaine certitude la trajectoire du véhicule dans le repère absolu, en utilisant les équations de dynamique Newtonnienne.



Comme indiqué ci-dessus, un IMU possède souvent un **magnétomètre**. Celui-ci permet de mesurer la direction et l'intensité du champ magnétique terrestre par rapport à l'IMU et permet d'en dériver le cap, la direction vers lequel le véhicule pointe sur la surface de la Terre.

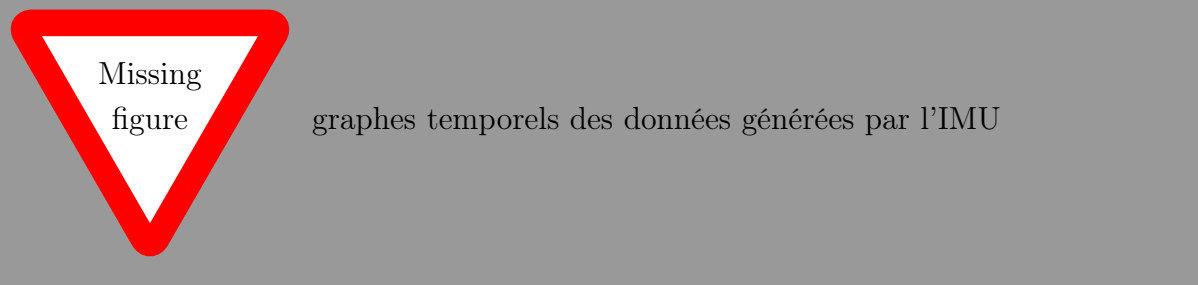
Ce permet, en démarrant l'IMU alors que le véhicule est parfaitement immobile, d'obtenir les conditions initiales du vecteur d'état du véhicule, par rapport au repère odométrique.

Par ailleurs, reconstituer une position à partir d'informations d'accélération implique mathématiquement de procéder à une double intégration numérique. Ce procédé est donc sujet à une dérive dans le temps et doit donc être recalé par une mesure de pose issue d'un autre capteur.

Par contre, un IMU peut fournir ses informations avec un très haut taux de rafraîchissement, sans nécessiter d'information extérieure au véhicule. Il peut donc se révéler indispensable lorsqu'aucune autre nouvelle information sur la position du véhicule n'est disponible sur des périodes de temps plus ou moins longue.

2.1.2.2 IMU XSense

REDIGER données générées par un IMU



2.1.3 GPS en mode RTK

Le système Symeter V2 ayant besoin d'une précision centimétrique pour permettre la mesure du tas d'ensilage avec précision du même ordre de grandeur, il mettra en oeuvre d'un GPS en mode RTK. Ce mode fonctionnement particulier du système GPS est en mesure de fournir position centimétrique du véhicule relativement à une borne GPS (base) situé à proximité du chantier.

2.1.3.1 GPS simple

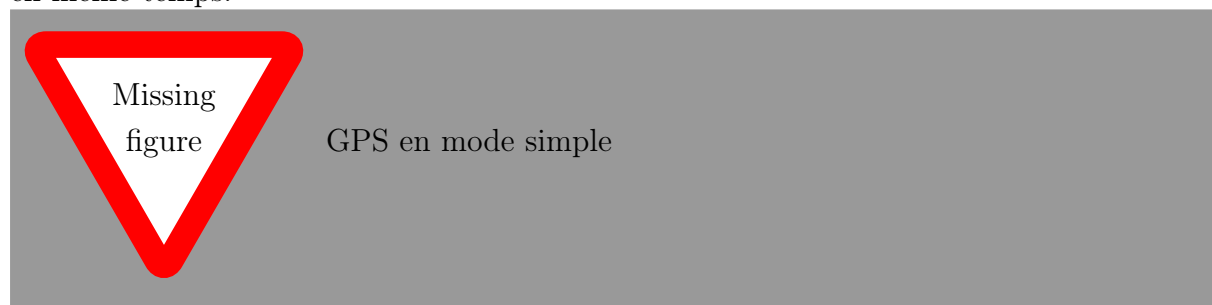
Pour expliquer le principe du GPS en mode RTK, il est nécessaire de comprendre comment le système GPS "simple" fonctionne.

Le système GPS est un système de positionnement par satellite qui, en se basant sur les signaux émis par une constellation de satellites, est en mesure de calculer la position sur la surface de la Terre d'un récepteur avec une précision de l'ordre de la dizaine de mètres, pour une utilisation courante.

Le principe de base est celui de la triangulation : les satellites GPS émettent en permanence un signal comportant une horloge (temps GPS). En connaissant les éphémérides des satellites et en écoutant les signaux de plusieurs de ces satellites, notamment leur signal d'horloge respectif, un récepteur GPS peut calculer avec une grande précision la position de chacun des satellites émetteurs dans le référentiel géocentrique.

Une fois le positionnement des satellites établi, une simple triangulation permet de calculer avec une grande précision du récepteur sur la surface de la Terre.

Pour obtenir une précision suffisante, un récepteur doit écouter au moins 4 satellites en même temps.



2.1.3.2 Principe du mode RTK

Le mode RTK du système GPS est un mode de fonctionnement qui permet de mesurer avec une très grande précision la position relative de deux récepteurs GPS distincts. L'un, dénommé **la base** est un récepteur fixe, dont la position GPS est connue avec une grande précision. L'autre dénommé **le rover** est en communication constante avec la base au moyen d'une liaison radio adaptée.

Les deux récepteurs étant présents sur le même site à la surface de la Terre, ils seront à l'écoute tous les deux mêmes satellites. En comparant les signaux reçus d'un même satellite par la base et le rover, le rover est en mesure de calculer avec une très grande précision - de l'ordre du centimètre - la différence de distance entre la base et le satellite d'une part, et le rover et le satellite de l'autre part.



GPS en mode RTK

En combinant ces différences de distance à partir de plusieurs satellites, le rover est donc capable de calculer très précisément sa position relativement à la base.

TROUVER REF EXTERNE : 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.



Quelques robots powered by ROS

ROS est un environnement logiciel destiné à la mise en oeuvre de plateforme robotique. Cet environnement très modulaire permet de programmer et de déployer de nombreux modules interdépendants, appelés **packages**, constitués eux de noeuds exécutables appelés **nodes**.

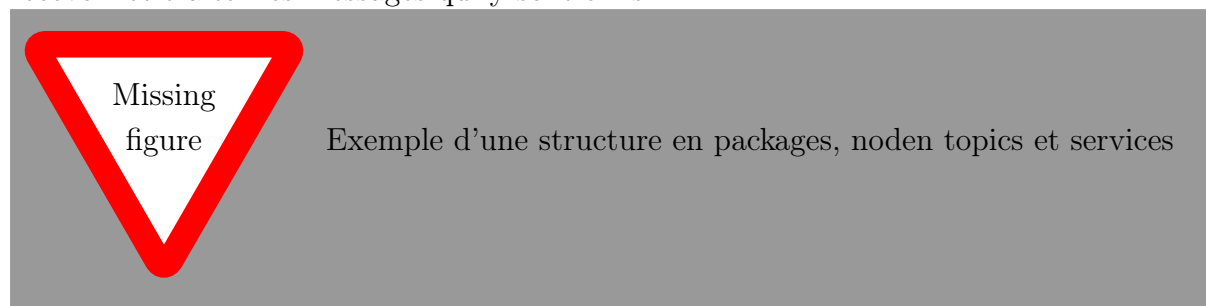
Ces nodes peuvent communiquer entre eux aux moyens de deux mécanismes principaux :

- les services
- les topics

Ces deux mécanismes permettent l'échange de données entre nodes selon des messages de la structure peut être spécifiée par le développeur du système.

Les **services** sont mis à disposition chacun par un node. Ils peuvent être invoqués par d'autre node en effectuant une transaction explicite : le node appelant fourni une donnée d'entrée au node mettant à disposition le service. Ce dernier traite la donnée entrante puis génère une donnée de sortie qui est enfin renvoyée au node, terminant ainsi la transaction.

Les **topics** sont des canaux de diffusion d'information similaire à une autoroute : tout node peut diffuser un message vers un topic et tout node peut écouter un topic pour recevoir et traiter les messages qui y sont émis.



Ces topics pourvoient des données que l'on pourrait qualifier de "sensorielle", dont la durée de vie est courte, telle que les données générées par un capteur fournissant des données en continu.

Ils sont aussi le support de diffusion des **transformations**, qui permettent de décrire l'état de **pose** de chacuns des composants mécaniques d'un véhicule autonome, ainsi que la pose du véhicule dans son environnement.

ROS est un environnement fourni sous une licence Open Source et propose des drivers pour de très nombreux capteurs et actionneurs de tous types. Il intègre aussi des composants logiciels tiers spécialisés tels que, entre autre, la Point Cloud Library pour le traitement des nuages de points et la librairie OpenCV pour le traitement de la vision par ordinateur.

2.3 Les outils mathématiques

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

Comme indiqué en introduction, le système Symeter V2 sera une plateforme semi-autonome, qui devra être en mesure de fournir une aide à la décision à partir de mesures effectuées depuis une plateforme mobile, et ceux avec une intervention humaine minimale.

Symeter V2 devra donc estimer en temps réel sa pose, afin de pouvoir intégrer une représentation 3D du chantier afin de pouvoir effectuer son service.

Les outils mathématiques qui devront être mis en oeuvre sont essentiellement ceux mis en oeuvre dans les systèmes robotiques :

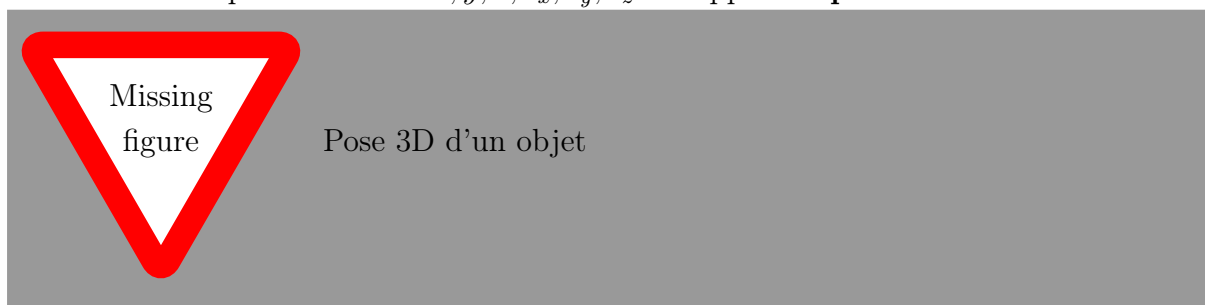
- Les **poses**, c'est à dire la représentation de la position des capteurs et des éléments mécaniques du véhicule les uns par rapport aux autres, et celle du véhicule par rapport à son environnement.
- L'estimation de paramètres à partir d'un ensemble de mesures issues de différents capteurs, au moyen de la **fusion de données**
- Reconstruction d'une scène 3D à partir de nuages de points

Ces outils sont décrits plus en détail dans les sections suivantes.

2.3.1 Positionnement en Robotique : Poses

La robotique a pour enjeu de permettre la mise en oeuvre de systèmes mécaniques autonomes, mobiles ou non, qui mettent en oeuvre une série de capteurs et d'actionneurs pour agir sur leur environnement avec une interaction humaine très limitée.

La capacité de représenter de manière fine la position des différents éléments d'un tel système les uns par rapport aux autres, que ce soit en position x, y, z , mais aussi en inclinaison - θ_x (tangage), θ_y (roulis), θ_z (lacet) - est donc primordiale. L'état de l'élément décrit par ce vecteur $x, y, z, \theta_x, \theta_y, \theta_z$ est appelé sa **pose**.



Les outils mathématiques de prédilection pour décrire une pose sont bien sûr les transformations dans \mathbb{R}^3 , translations et rotations. Cependant le système de coordonnées classique à 3 dimensions spatiales x, y, z ne permet pas de combiner simplement les transformations 3D, ce qui est un besoin de base pour passer d'un référentiel à un autre.

C'est pour cela que la robotique fait une utilisation intensive des **Coordonnées Homogènes**, issues de la géométrie projective, qui permettent de combiner les transformations 3D en passant par un espace à 4 dimensions approprié. Les transformations y sont représentées par des matrices de $\mathbb{R}^{4 \times 4}$, ce qui permet de combiner les transformations par simple multiplication de matrices.

Les principes de base des coordonnées homogènes sont décrites en Annexe XXX

AJOUTER
Lien vers l'

Les rotations dans l'espace sont aussi représentables par des quaternions de norme unité. Cette représentation est souvent préférée par la robotique par rapport à une représentation basée sur les rotations eulériennes car elle permet de combiner les rotations sans rentrer dans un "gimbal lock".

ROS utilise les quaternions de manière native dans la description de ses transformations.

AJOUTER
Quaternion
quaternions

2.3.2 Localisation par fusion de données

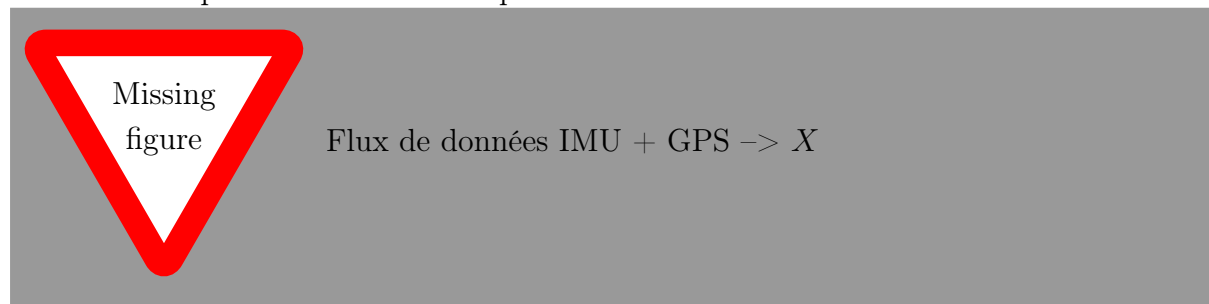
Afin de pouvoir reconstituer le chantier en une représentation 3D avec une précision centimétrique, il est nécessaire que pour chaque mesure LIDAR le système Symeter V2 aie une estimation suffisamment précise de la pose du véhicule, afin de déterminer la pose du LIDAR et pouvoir situer dans le chantier la position x, y, z de chaque point mesuré.

Le problème de la localisation du véhicule est donc le suivant :

Estimer à chaque instant t_n le vecteur d'état :

$$\mathbf{X} = \begin{bmatrix} x_n \\ y \\ z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \quad (2.1)$$

du véhicule à partir des données disponibles.



Les sources de données disponibles pour estimer la pose du véhicule à chaque instant sont :

- L'IMU,
- Le GPS en mode RTK.

L'IMU fournit des informations sur l'inclinaison du véhicule, ainsi que les accélérations linéaires qu'il subit à une fréquence de l'ordre de 30 Hz. Cependant comme indiqué ci-dessus, calculer la position du véhicule à partir de la mesure des accélérations subies requiert une double intégration, sur un signal d'accélération qui est généralement très bruité. Ces éléments impliquent une forte probabilité de dérive dans le temps.

Le GPS en mode RTK peut fournir une mesure de la position du véhicule dans la scène avec une précision de l'ordre de quelques centimètres, mais à un taux de rafraîchissement beaucoup plus bas, de l'ordre de 2 à 5 Hz. De plus le GPS ne fournit aucune information sur l'assiette du véhicule.

REDIGER introduire les filtres de Kalman



chaîne de traitement de fusion de données

COMPLETER Comme le GPS fournit une position absolue dans le référentiel de la scène nous ne sommes pas dans une problématique SLAM. Le véhicule n'est pas autonome, la navigation est assurée par l'opérateur du tracteur.

AJOUTER
Filter de K
pour la fusi
données.

2.3.3 Traitement des nuages de points

Le LIDAR utilisé par le système Symeter V2 étant de type "2D planar", les points retournés lors d'une unique mesure feront tous partie du même plan. L'ensemble des mesures sera donc constitué d'un ensemble de tranches qu'il faudra reconstituer à l'aide d'un traitement adapté.

Le traitement des données LIDAR comportera les étapes suivantes :

1. Filtrage des données pour obtenir une densité de points de mesure qui soient suffisamment espacés
2. Conversion des données LIDAR (direction / distance) en un nuage de points (liste de point P_n avec chacun des coordonnées x_n, y_n, z_n)
3. Stockage de ces points dans une structure de données capable d'accumuler les nouveaux points mesurés au cours du temps, et constituer une carte 3D d'occupation.

La plupart de ces traitements seront implémentés en utilisant le Point Cloud Library qui offre de nombreux composants logiciels intégrés directement dans ROS.



Architecture de traitement des nuages de points

2.4 Contraintes de développement

Le système Symeter V2 st destiné à évoluer sur un véhicule de type tracteur agricole. Il est de plus destiné à construire une représentation 3D d'un chantier d'ensilage, partir de positions qui vont varier non seulement en x et en y , mais aussi et surtout en z .

2.4.1 Capacités de tests en grandeur limitées

Pour développer et tester le système Symeter V2, le développeur ne disposera pas de tracteur en grandeur réelle, ni de chantier d'ensilage, ceux-ci étant des chantiers annuels se déroulant à des moments bien défini dans l'année (printemps et automne).

Par ailleurs, au début du stage il n'est pas encore tout à fait décidé de combien de LIDAR seront nécessaires pour que le système puisse assurer une couverture complète depuis des équipements embarqués sur le tracteur.

Les équipements à mettre en oeuvre sont relativement couteux et leur mise en oeuvre requiert une certaine expertise. Il est donc nécessaire de pouvoir mettre en oeuvre le développement du système par le biais de mises en oeuvre alternatives.

Ces possibilités sont au nombre de 2 : la simulation et la mise en oeuvre en grandeur des équipements sur une camionnette pour effectuer des tests simples de reconstitution du terrain.

2.4.2 Plateformes de test disponibles

2.4.2.1 Environnement de simulation Gazebo

Gazebo est un environnement de simulation très complet, qui permet de simuler un environnement physique dans lequel une plateforme semi-robotique simulée peut évoluer.

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

Gazebo est un logiciel intégré à ROS. Sa mise en oeuvre requiert de programmer un véhicule virtuel similaire à la plateforme cible du système (dans notre cas un tracteur avec les capteurs listés dans les sections précédentes), puis de déployer le logiciel ROS pour Symeter V2 sur ce véhicule simulé.

Il est aussi possible de spécifier le "monde" dans lequel le tracteur simulé évoluera, en y ajoutant des murs, du relief, voire une butte en guise de tas d'ensilage.

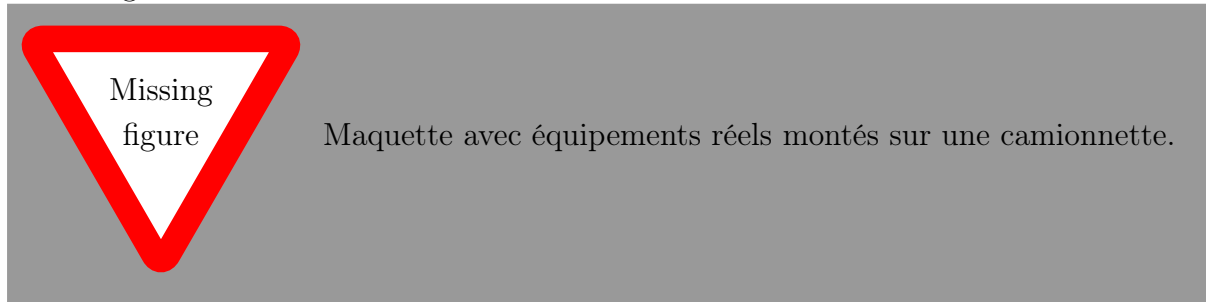


Capture d'écran Gazebo avec le tracteur et le silo

Gazebo est donc utilisé pour simuler un chantier d'ensilage pour tester la localisation, l'acquisition du terrain, la mesure d'un modèle de Silo.

2.4.2.2 Prototype monté sur Camionnette

Une simulation n'étant jamais parfaite (comme nous le verrons par la suite), il est quand même nécessaire d'effectuer des certains tests avec des instruments réels, lors de tests en grandeur.



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

Le prototype comporte :

- Une unité de calcul
- Un GPS en mode RTK
- Un LIDAR de type Hkuyo UTM-30LX-EW
- Un IMU de type XSense

Ce prototype a été utilisé pour effectuer des captures des trois capteur alors que la Tellus Car évoluait dans le parking adjacent aux locaux de Tellus Environnement.

2.5 Les grandes parties du projet

REDIGER comme il est apparent, proeît 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

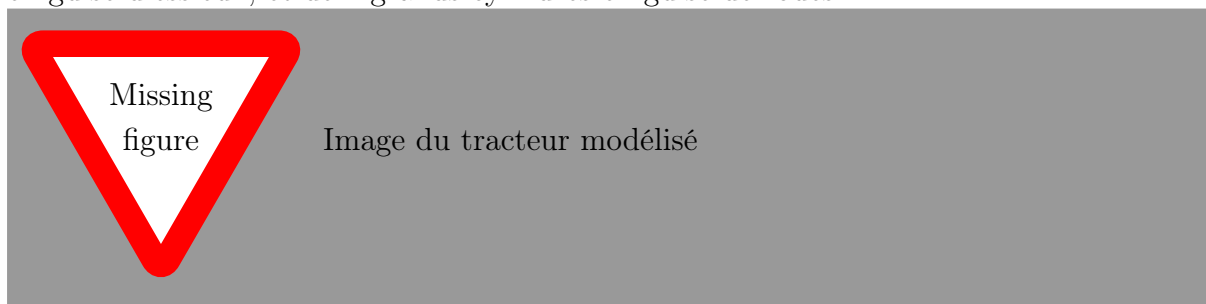
Intérêt : simuler l'implantation physique des capteurs avec des dimensions du même ordre de grandeur que les plateformes cibles.

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

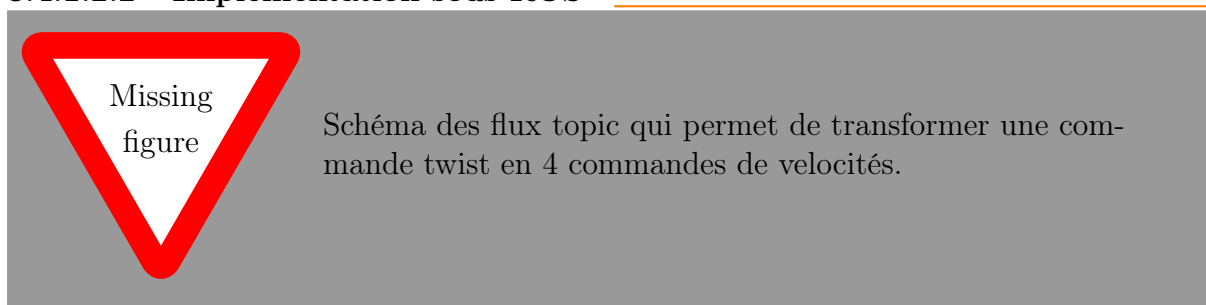
3.4.1.2 Propulsion et Guidage

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.