



**HAL**  
open science

# Développement d'un robot autonome pour l'acquisition active et le traitement d'images

M. Etievant

► **To cite this version:**

M. Etievant. Développement d'un robot autonome pour l'acquisition active et le traitement d'images. Sciences de l'environnement. 2017. hal-02607108

**HAL Id: hal-02607108**

**<https://hal.inrae.fr/hal-02607108v1>**

Submitted on 16 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# DÉVELOPPEMENT D'UN ROBOT AUTONOME POUR L'ACQUISITION ACTIVE ET LE TRAITEMENT D'IMAGES

**Projet de fin d'étude**

**Maxime ETIEVANT**  
**18 septembre 2017**

Institut de recherche :  
**IRSTEA**

Tuteur d'entreprise :  
**Bernard BENET**

Tuteur IFMA :  
**Nicolas BOUTON**

Date :  
**du 20 mars au 19 septembre 2017**



**TITRE DU DOCUMENT : Développement d'un robot autonome pour l'acquisition active et le traitement d'images**

Projet de fin d'étude

**AUTEUR : Maxime ETIEVANT**

Date du document :	Nombre de pages :	Référence du document :
18 septembre 2017	55	Rapport_final.pdf

**ABSTRACT :**

The purpose of this document is to present the work carried out during my internship. These are part of Phenaufol project, led by the Technical Institute of sugar Beet which link AgroSup Dijon and IRSTEA. This project is in line with the work carried out in 2016 to develop a robotic system to autonomously acquire data in sugar beet plots. This document will focus on localization through artificial vision.

Key words : mobile robotics, precision agriculture, artificial vision

**RÉSUMÉ :**

Ce document a pour but de présenter les travaux effectués dans le cadre de mon stage de fin d'étude. Ceux-ci s'incrivent dans le cadre du projet Phenaufol, piloté par l'Institut Technique de la Betterave liant AgroSup Dijon et IRSTEA. Ce projet se place dans la continuité des travaux menés en 2016 consistant à développer un système robotisé destiné à acquérir des données dans des parcelles de betterave sucrière de façon autonome. Ce document s'intéressera particulièrement à la localisation par vision artificielle.

Mots clés : robotique mobile, agriculture de précision, vision artificielle



# Remerciements

Avant toute chose, je tiens à remercier chaleureusement Jean-Pierre CHANET et Roland LENAIN de m'avoir accueilli dans l'unité de recherche TSCF et l'équipe ROMEA. Je souhaite également exprimer toute ma gratitude à Bernard BENET (tuteur entreprise) et Nicolas BOUTON (tuteur école) de m'avoir encadré et accordé leur confiance au cours de ces six mois.

J'aimerais aussi remercier Luc, Guillaume et Serhii d'avoir pris le temps de répondre à toutes mes questions, aussi nombreuses furent-elles, mais aussi Abdourahamane, Adrian, Mathieu et Vianney pour leur bonne humeur et leur humour. Merci également à Camille ainsi que Thibault qui connaissaient la musique et ont su me donner le "la" tout au long de ce stage.

Je tiens également à remercier l'ensemble des personnes qui m'ont permis d'effectuer ce stage dans les meilleures conditions.

# Table des matières

<b>I</b>	<b>Présentation de l'entreprise</b>	<b>1</b>
1	L'entreprise . . . . .	1
2	IRSTEA en Auvergne . . . . .	2
3	L'équipe ROMEA . . . . .	2
<b>II</b>	<b>Présentation du stage</b>	<b>3</b>
1	Contexte . . . . .	3
2	Objectifs . . . . .	3
3	Matériel utilisé . . . . .	4
3.1	Systèmes robotisés . . . . .	4
3.2	Outils informatiques . . . . .	5
<b>III</b>	<b>Suivi de lignes par vision</b>	<b>9</b>
1	État de l'art . . . . .	9
1.1	Aide à la conduite . . . . .	9
1.2	Suivi de cultures . . . . .	10
1.3	Choix d'une approche . . . . .	13
2	Construction de l'approche . . . . .	14
2.1	Notations utilisées . . . . .	14
2.2	Modélisation du problème . . . . .	15
2.3	Architecture de l'approche . . . . .	16
3	Initialisation et création des zones de recherche . . . . .	19
3.1	Hypothèse initiale sur la position / construction de $X_l^0$ . . . . .	19
3.2	Passage dans le repère image / construction de $X_d^0$ . . . . .	19
3.3	Construction des zones de recherche . . . . .	19
4	Phase de recherche . . . . .	20
4.1	Principe . . . . .	20
4.2	Détection par régression linéaire . . . . .	22
4.3	Détection par méthode mixte . . . . .	25
5	Résultat de la localisation . . . . .	26
5.1	Commande classique . . . . .	27
5.2	Commande référencée vision . . . . .	27
6	Prédiction . . . . .	28
6.1	Cas $X_l^0$ fixe . . . . .	28
6.2	Cas $X_l^0 = \hat{X}_l$ . . . . .	28
6.3	Cas $X_l^0 = \hat{X}_l + vdt$ . . . . .	29
6.4	Utilisation d'un filtre de Kalman . . . . .	29
7	Synthèse sur l'approche développée . . . . .	30

---

<b>IV Mesures sur les plants</b>	<b>31</b>
1 Modélisation du bras UR5 . . . . .	31
1.1 Description du bras . . . . .	32
1.2 Modèle géométrique direct . . . . .	33
1.3 Modèle géométrique inverse . . . . .	33
2 Détection des plants . . . . .	34
3 Synthèse . . . . .	35
<b>V Conclusion</b>	<b>36</b>
<b>Bibliographie</b>	<b>38</b>
<b>A Note de calcul pour la modélisation</b>	<b>39</b>
<b>B Calcul de la matrice Jacobienne <math>J_C</math></b>	<b>41</b>
1 Définition de $P_{i,j}$ . . . . .	43
2 Calcul de $\frac{\partial P_{i,j}}{\partial q_i}$ . . . . .	44
3 Définition de $\frac{\partial C}{\partial q_i}$ . . . . .	45
<b>C Principe de discrimination par SVM</b>	<b>46</b>
1 Principe général . . . . .	46
2 Cas d'une fonction discriminante linéaire . . . . .	47
2.1 Formalisation du problème . . . . .	47
2.2 Résolution du problème . . . . .	47
3 Cas d'une fonction discriminante non-linéaire . . . . .	48
<b>D Introduction au filtre de Kalman</b>	<b>49</b>
<b>E Méthode de Denavit-Hartenberg</b>	<b>50</b>
<b>F Note de calcul du bras UR5</b>	<b>51</b>
1 Matrices de passage . . . . .	51
2 Modèle géométriques direct . . . . .	52
3 Modèle géométrique inverse . . . . .	52

# Table des figures

I.1	Répartition des différents centres . . . . .	1
II.1	Dispositifs BECAM . . . . .	3
II.2	Photographies du robot Jaguar . . . . .	4
II.3	Photographie du bras UR5 . . . . .	5
II.4	Principe général de ROS . . . . .	6
II.5	Aperçu des possibilités d'OpenCV . . . . .	8
III.1	Détection de ligne par méthode de Hough . . . . .	10
III.2	Détection de ligne par superposition . . . . .	11
III.4	Détection de ligne par double méthode de Hough . . . . .	12
III.6	Détection autour d'un motif . . . . .	13
III.7	Notations utilisées . . . . .	15
III.8	Schéma de l'algorithme utilisé . . . . .	17
III.9	Notations dans l'espace image . . . . .	18
III.10	Droites horizontales . . . . .	19
III.11	Lignes estimées . . . . .	19
III.12	Intervalle de confiance pour la rangée centrale . . . . .	20
III.13	État initial avec hypothèse . . . . .	21
III.14	Détection dans l'image binarisée . . . . .	23
III.15	État initial . . . . .	23
III.16	État après première détection . . . . .	23
III.17	État après seconde détection . . . . .	24
III.18	Résultat après 27 détections . . . . .	24
III.19	Lignes retrouvées par régressionss linéaires . . . . .	24
III.20	État initial . . . . .	25
III.21	État après première détection . . . . .	25
III.22	État après seconde détection . . . . .	26
III.23	État après troisième détections . . . . .	26
III.24	Résultats . . . . .	26
III.25	Représentation de la prédiction dans le cas où $X_l^0$ est fixe . . . . .	28
III.26	Représentation de la prédiction dans le cas où $X_l^0 = \hat{X}_l$ . . . . .	29
III.27	Représentation de la prédiction dans le cas où $X_l^0 = \hat{X}_l + vdt$ . . . . .	29
III.28	Représentation de la prédiction avec utilisation du filtre de Kalman . . . . .	30
IV.1	Représentation du bras UR5 . . . . .	32
IV.2	Modélisation du bras UR5 . . . . .	32
IV.3	Illustration de la détection des plants . . . . .	35
C.1	Exemple de discrimination . . . . .	46
D.1	Principe du filtre de Kalman . . . . .	49

# I Présentation de l'entreprise

Avant d'exposer le travail réalisé au cours de ce stage de fin d'étude, une brève présentation de l'entreprise est développée ci-dessous. La première partie portera sur l'institut dans sa globalité pour nous intéresser ensuite au site d'Aubière et plus particulièrement à l'équipe ROMEA<sup>1</sup> au sein de laquelle s'est déroulé ce stage.

## 1 L'entreprise

IRSTEA<sup>2</sup> est un établissement public à caractère scientifique et technologique (EPST) placé sous la double tutelle des ministères en charge de la recherche et de l'agriculture.

La recherche au sein de l'IRSTEA se décline suivant trois axes :

- **l'étude et la gestion des eaux superficielles et des activités liées** : gestion de l'eau, des usages, des services et des infrastructures ; qualité des systèmes aquatiques et restauration écologique ; aléas et risques liés au cycle de l'eau ; réponses biologiques et écologiques aux contaminants du milieu aquatique ; risques liés aux phénomènes gravitaires rapides et sûreté des ouvrages ;
- **l'étude et la gestion des territoires** : développement territorial et agriculture multifonctionnelle ; systèmes écologiques terrestres : dynamique, vulnérabilités et ingénierie ; systèmes d'information spatiale pour la gestion intégrée de l'environnement ;
- **l'écotechnologie**<sup>3</sup> : modèles, systèmes d'information et gestion viable de l'environnement ; structures, procédés, écoulement, énergie ; technologies et procédés pour l'eau et les déchets ; innovation technologique au service de l'agriculture et de l'environnement.

En 2015, l'entreprise employait près de 1600 personnes réparties sur neuf centres régionaux.



FIGURE I.1 – Répartition des différents centres

---

1. RObotique et MObilité pour l'Environnement et l'Agriculture  
2. Institut national de Recherche en Sciences et Technologie pour l'Environnement et l'Agriculture  
3. Ensemble de techniques minimisant la formation et/ou l'émission de polluants durant les procédés de production, ainsi que le savoir-faire écologique et les nouvelles méthodes de travail associées

Chaque centre de recherche est ensuite divisé en thèmes de recherches plus spécifiques dans lesquels interviennent différentes unités de recherche. La partie suivante s'intéressera uniquement au centre de Clermont-Ferrand (lieu du stage).

## 2 IRSTEA en Auvergne

Le centre IRSTEA implanté en Auvergne comporte deux sites (Montoldre dans l'Allier et Aubière sur le campus des Cézeaux). Ensemble, les deux groupements mènent des recherches en faveur du développement d'innovations technologiques pour l'agriculture et les territoires.

Le site de Clermont-Ferrand, regroupe deux unités de recherche, TSCF (Technologies et Systèmes d'information pour les agrosystèmes) et le LISC (Laboratoire d'Ingénierie pour les Systèmes Complexes), ainsi qu'une unité mixte de recherche, Territoires.

L'unité de recherche TSCF de Clermont-Ferrand compte une cinquantaine d'agents se répartissant en trois équipes :

- COPAIN<sup>4</sup> consacrée aux méthodes d'ingénierie des systèmes d'information communicants dédiées à la gestion agri-environnementale. Cet ensemble de méthodes couvre l'analyse des besoins des acteurs, la spécification des systèmes d'information, leur modélisation, leur conception, leur gestion et leur lien avec les sources de données.
- PRT PEE<sup>5</sup> conduit des activités de recherche et développement dans le domaine de la fertilisation, minérale et organique.
- ROMEA<sup>6</sup> conçoit des systèmes reconfigurables et à autonomie partagée, pour accroître les performances et la sécurité des engins œuvrant en milieux naturels, en particulier ceux rencontrés dans l'agriculture. L'équipe s'intéresse aussi la liaison tracteur/sol dans le but d'évaluer les performances des pneumatiques et modéliser la compaction des sols.

## 3 L'équipe ROMEA

Cette équipe dirigée par M. LENAIN mène des recherches axées autour de trois volets principaux :

- La commande des déplacements et le partage d'autonomie avec l'opérateur : les travaux relatifs à ce volet ont pour but de développer des algorithmes avancés de commandes ou d'assistance, en prenant en compte les phénomènes perturbants et les incertitudes liées à l'évolution en milieux tout-terrain et dans différentes conditions.
- La conception de dispositifs de sécurité : l'équipe développe des mécanismes, potentiellement actifs, pour accroître la sécurité et la sûreté de fonctionnement.
- La perception et la caractérisation de l'interaction d'un engin dans son environnement : ce domaine concerne la modélisation du comportement des machines et de l'interaction avec l'utilisateur mais aussi la perception, à des fins de localisation et de représentation de l'environnement.

---

4. Systèmes d'information communicants et agri-environnementaux

5. Plateau de Recherche Technologique Pôle Épandage Environnement

6. RObotique et Mobilité pour l'Environnement et l'Agriculture

# II Présentation du stage

## 1 Contexte

Ce stage s'inscrit dans le cadre du projet «Casdar PHÉNAUFOL<sup>1</sup>» issu d'une collaboration entre l'Institut Technique de la Betterave (ITB), l'AgroSup Dijon avec l'IRSTEA. L'objectif de ce partenariat est de réaliser un phénotypage autonome (mesure d'un caractère observable d'un organisme (forme, croissance, propriétés biochimiques...)) sur des plants de betteraves sucrières, en s'intéressant notamment à la détection et la propagation de maladies pendant les différents stades de levée.

Cette analyse a pour but de mettre en lumière la résistance de plusieurs variétés de betterave face aux maladies. De ce fait, ce projet tend vers une diminution des intrants phytosanitaires comme l'orientent les différentes réglementations françaises et européennes.

Dans ce projet supervisé par l'ITB, l'AgroSup Dijon et l'IRSTEA apportent respectivement leur expertise dans la détection de maladie par traitement d'image et dans les opérations de contrôle/commande de système robotisé.

## 2 Objectifs

Le but de cette étude est d'élaborer un outil de phénotypage moyen/haut débit, pour remplacer BECAM, outil actuellement utilisé par l'ITB pour l'acquisition d'image sur des parcelles de betteraves sucrières pour réaliser des mesures colorimétriques et géométriques. Pour ce faire, un cahier des charges préliminaire a été effectué sur l'utilisation d'une plateforme mobile associée à un bras manipulateur portant des capteurs.

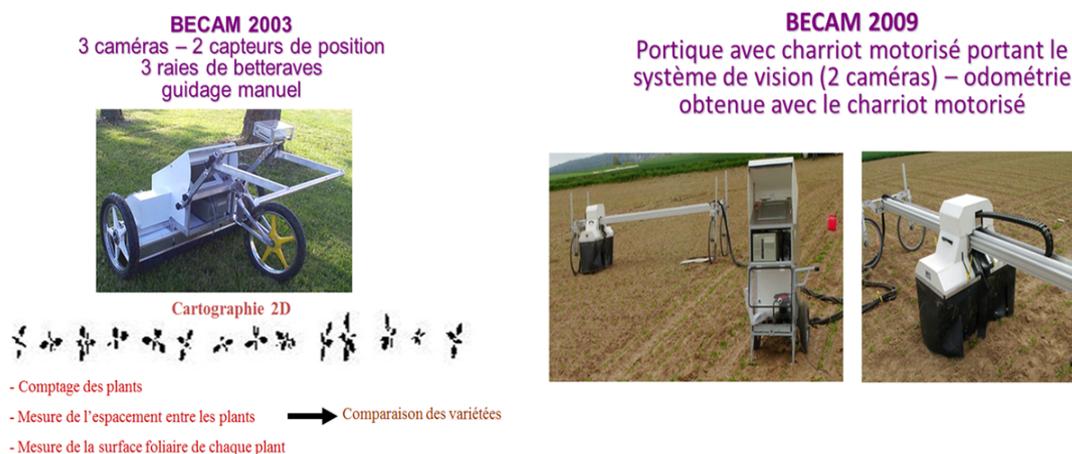


FIGURE II.1 – Dispositifs BECAM

1. PHÉNotypage pour détecter AUTomatiquement les maladies FOLiaires.

Trois objectifs principaux se dégagent :

- Réaliser un suivi autonome des lignes de betteraves par vision.
- Réaliser des mesures sur les plants de betteraves par vision (comptage des plants, surface foliaire, mesures géométriques et colorimétriques, cartographie, mesures individuelles et statistiques).
- Réaliser une commande active de la position de la caméra afin d’optimiser la précision des mesures.

Comme nous le verrons, ce stage peut être décomposé en deux parties distinctes :

- La première portera sur le suivi autonome de lignes de cultures par vision. Cette portion s’intéressera aux cultures en début de croissance (lorsque le taux de recouvrement est faible).
- La seconde portera sur le relevé de mesures sur les plants. Ici, il s’agira d’étudier des betteraves matures (stade de levée très avancé) à l’aide d’une caméra montée sur un bras manipulateur.

### 3 Matériel utilisé

#### 3.1 Systèmes robotisés

##### Jaguar

Dans un premier temps, il a été décidé d’utiliser un robot mobile (appelé Jaguar) pour surplomber un rang. Ceci n’étant possible que lorsque les betteraves ne se recouvrent pas, cette solution s’adresse au stade de levée peu avancé (jusqu’à quatre feuilles par plant).



FIGURE II.2 – Photographies du robot Jaguar

Sur ce robot, on peut apercevoir deux caméras :

- Une webcam sur le dessus permettant d’avoir une vision d’ensemble des rangées, utilisée pour le suivi de lignes.
- Une caméra giga ethernet pointée vers le sol servant à étudier le développement des plants.

## Bras robotisé UR5

Lorsque les betteraves ont atteint un stade avancé, avec un taux de recouvrement important, un robot mobile ne peut pas être utilisé pour réaliser des mesures à l'intérieur des parcelles. Par conséquent, pour palier ce problème et ne pas endommager les cultures, il a été choisi de monter un bras manipulateur (bras UR5 développé par Universal Robot) sur un axe linéaire raccordé à une plateforme mobile de plus grande taille permettant ainsi de s'approcher au plus près des zones d'intérêt sans endommager les cultures.



FIGURE II.3 – Photographie du bras UR5

L'étude du bras se fera dans la partie [Mesures sur les plants](#)

## 3.2 Outils informatiques

### ROS

#### Principe et intérêt

ROS (*Robot Operating System*) est une plateforme de développement logiciel gratuite et *open source* dédiée à la robotique. Il appartient à la catégorie des *middlewares*<sup>2</sup> permettant par conséquent de contrôler des périphériques de bas niveau, faciliter la réutilisation de programmes et la communication entre processus etc.

Un des principes fondamentaux de ROS est de proposer des *plugins* réutilisables permettant donc d'accélérer les recherches en robotique et de favoriser le partage de données/programmes. Il propose notamment une large gamme de paquet (de fiabilité et performance variables) permettant d'utiliser différents types de capteurs facilement. Mais ce n'est pas tout, ROS propose également un grand nombre d'outils, de tutoriels d'utilisation, d'applications de perception, commande et autres algorithmes.

Nous nous intéresserons dans les prochains paragraphes aux principaux concepts utilisés par ROS. Cette liste n'est pas exhaustive, mais donne un aperçu du fonctionnement général de ROS.

#### Comprendre la notion de paquet

Tous les composants de ROS sont répartis dans des *packages*, un dossier pouvant contenir des nœuds, des bibliothèques, des fichiers de configuration etc. La plupart du temps, un *package* regroupe un ensemble de nœuds servant à un même processus. Chaque

---

2. Logiciel tiers permettant la communication notamment entre des applications informatiques disparates et des capteurs.

paquet est contenu dans un environnement de travail isolé appelé *workspace*.

Généralement l'organisation d'un *package* est la suivante :

- Un fichier **CMakeLists.txt** contenant une série d'instructions destinées à la compilation du projet.
- Un fichier **package.xml** contenant des informations sur le *package*.
- Un fichier **README.md** contenant des informations utiles aux futurs utilisateurs du *package*.
- Des dossiers :
  - **/src** regroupant des programmes C++ et des bibliothèques internes au *package*.
  - **/launch** regroupant des fichiers de lancement.
  - **/config** regroupant des fichiers de configuration.
  - **/srv** et **/msg** regroupant la structure des messages spécifiques et services utilisés.
  - **/scripts** regroupant des programmes Python.
  - **/include** regroupant les bibliothèques qui seront partagées avec d'autres paquets.

### Comprendre la notion ROS *Master*

La notion de ROS *Master* est essentielle à comprendre. En effet, le *Master* est un service de déclaration et d'enregistrement des nœuds. Il permet principalement de référencer et de mettre en communication les nœuds via les *topics* correspondants.

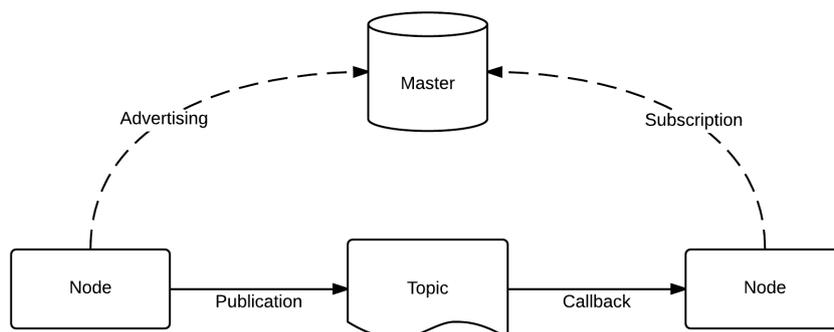


FIGURE II.4 – Principe général de ROS

Source : [www.wikipedia.org](http://www.wikipedia.org)

### Comprendre les notions de nœuds, messages et *topics*

Un nœud est un programme capable d'interagir avec le *Master* afin de communiquer avec d'autres entités.

Un *topic* est l'équivalent d'un fil de discussion. Ils permettent de diffuser un message d'un type précis de manière **asynchrone**.

Un message est une structure de données composée d'une combinaison de type primitif (entiers, booléens, flottants...) ou d'autres messages. Généralement, il est conseillé d'utiliser des messages standards pour que le code soit facilement réutilisable.

Un nœud peut avoir deux rôles, il peut :

- publier un message sur un *topic*, il est alors appelé *publisher*.
- scruter l'activité d'un *topic*, il est alors appelé *subscriber*.

**NB** : Un nœud peut souscrire et publier sur plusieurs *topics* en même temps.

NB : La programmation par nœud permet une grande modularité, chaque nœud étant une petite brique de code facilement remplaçable.

#### Comprendre la notion de service

Un service est une alternative aux *topics*. Elle a pour particularité de permettre à deux nœuds uniquement de travailler de manière synchrone. On peut voir les services comme une fonction avec une entrée (requête émise par un nœud client), un traitement et une sortie (réponse retournée à un nœud).

Ici aussi, la structure des requêtes et des réponses est fixe (définie dans un fichier **\*.srv** situé dans le dossier **/srv**).

#### Comprendre la notion de fichier de lancement

Comme son nom l'indique, un fichier de lancement permet de décrire la composition et la configuration d'un système ROS (nœuds à charger, paramètres utilisés, renommages éventuels ...).

NB : Un fichier de lancement peut en appeler un autre.

#### Comprendre la notion de serveur de paramètres

Afin de permettre une reconfiguration facile du robot, le *master* possède un serveur de paramètres partagés aux différents nœuds. Ainsi, tous les nœuds peuvent utiliser des paramètres communs (tel que le format d'une image, la fréquence de travail etc) sans que l'utilisateur ait besoin de modifier le coeur du code.

#### Quelques outils

Pour aider au développement de système robotisé, l'environnement ROS propose de nombreux outils comme un :

- Simulateur 3D (Gazebo).
- Outil de visualisation (Rviz).
- Outil de développement (Rqt) composé de nombreux *plug in* dont par exemple :
  - Graph : Permet de visualiser l'ensemble des nœuds et des *topics* actifs.
  - PackageGraph : Permet de visualiser le graphique des dépendances.
  - LaunchTree : Permet de parcourir l'arborescence d'un fichier **.launch**.
  - Plot : Trace un graphique avec les données d'un *topic*.
  - ImageView : Affiche l'image publiée par une caméra.
- Système d'enregistrement et de rejeu (Rosbag<sup>3</sup>).
- Outil de planification de mouvement (notamment utilisés pour les bras).

## OpenCV

OpenCV (*Open source Computer Vision*) est une bibliothèque gratuite proposant de nombreux algorithmes de vision par ordinateur. Grâce à sa communauté très active et sa documentation fournie, elle est aujourd'hui (avec la librairie PIL) la bibliothèque de référence pour la vision par ordinateur. L'intérêt de cette bibliothèque réside notamment dans sa simplicité d'utilisation combinée à sa performance.

---

3. Outil permettant d'enregistrer l'activité d'une partie voire de tous les *topics* actifs (au choix) et de les rejouer ultérieurement. Très utilisé pour le débogage.

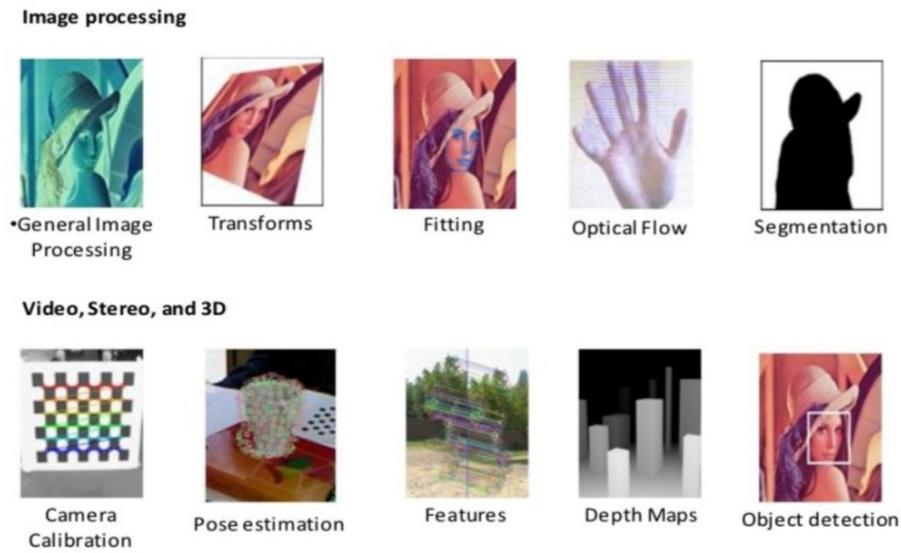


FIGURE II.5 – Aperçu des possibilités d'OpenCV

Source : fr.slideshare.net

Bien que cet outil soit performant, il faut bien comprendre que le temps de traitement dépend à la fois de la complexité du calcul, mais aussi de la taille de l'image.

# III Suivi de lignes par vision

## 1 État de l'art

Avant de choisir l'approche semblant la plus intéressante dans notre cas de figure, intéressons nous aux travaux concernant le suivi de lignes à l'aide d'une caméra. Ce travail a pour but de déterminer l'écart (latéral et angulaire) du véhicule par rapport à une trajectoire donnée. Ce paragraphe n'a pas vocation à être exhaustif, mais permet simplement de se faire une idée sur les approches courantes dans ce domaine.

Dans cette section, nous commencerons par rappeler brièvement quelques travaux réalisés dans le domaine de l'aide à la conduite pour nous intéresser, dans un second temps aux recherches menées sur le suivi de cultures. Enfin, nous positionnerons notre approche vis-à-vis de cette étude. Nous limiterons notre étude au guidage par caméra monoculaire.

Dans la plupart des travaux, l'approche utilisée suit la même architecture. Les auteurs commencent tout d'abord par effectuer une segmentation sur l'image.<sup>1</sup> Généralement, cette opération permet de distinguer deux classes de pixels (souvent noire/blanche) suivant l'intérêt de ceux-ci; on parle alors de binarisation. Cette étape est fondamentale car elle conditionne la suite du processus. Cependant, le traitement d'image étant couteux en temps de calcul, il faut donc trouver un équilibre entre le temps d'exécution et la pertinence du résultat. Le paragraphe [Comparatif des méthodes de segmentation](#) donne une idée intuitive<sup>2</sup> de la performance des principaux algorithmes utilisés.

Une fois l'image correctement segmentée, une méthode de détection de lignes est utilisée pour mesurer leur position.

Enfin, connaissant la position des lignes dans l'image, on peut connaître la position relative du robot par rapport à ces lignes dans l'espace réel.

### 1.1 Aide à la conduite

Afin de réduire le nombre d'accidents sur la route, de nombreux projets ont vu le jour ces vingt dernières années centrés sur la conduite assistée voire même autonome. Lorsque seule une camera est utilisée pour le suivi de route, trois grandes stratégies de recherche se dégagent.

La première se focalise sur la segmentation de l'image afin de faire ressortir la forme de la route par *clustering*<sup>3</sup> pour ensuite l'étudier, c'est le cas par exemple de [1]. Bien que ces méthodes soient intéressantes, la plupart sont mises à mal lorsque la route est discontinue (lorsqu'un objet l'obstrue par exemple) et ne sont donc pas réellement adaptées à notre

---

1. Opération visant à rassembler des pixels suivant des critères pré-établis pour simplifier et/ou changer la représentation de l'image.

2. La performance de l'algorithme dépendant du matériel utilisé ainsi que de l'exemple à traiter, on préfère ici utiliser une approche qualitative que quantitative.

3. Méthode de partitionnement des données permettant la création de plusieurs classes.

cas de figure. En effet, les cultures à suivre étant souvent clairsemées contrairement aux routes, cela rend difficile la création d'une entité unique, rendant l'étude plus complexe.

La seconde approche propose de détecter le point de fuite pour ensuite rechercher la route correspondante. On retrouve cette idée dans [2] ou encore [3]. Or nous verrons par la suite qu'il existe des moyens plus adaptés au suivi de culture.

Enfin la dernière méthode propose d'utiliser un modèle à retrouver dans l'image. L'idée intuitive consiste à faire une supposition sur la position relative entre le véhicule et la route qu'il faudra rectifier en observant l'image. Une très intéressante est notamment celle développée [4]. Celle-ci permet, à l'aide processus de recherche guidé par un modèle statistique des bords, une gestion efficace des occultations ou imperfections du marquage et une adaptation aux routes peu structurées.

## 1.2 Suivi de cultures

Avec le développement de l'agriculture de précision, de nombreux travaux ont été menés afin de pouvoir utiliser des robots d'assistance (aide à la conduite d'engins agricoles) voire même autonomes. Pour cela, il est nécessaire de pouvoir suivre des cultures de façon précise. Les paragraphes suivants présentent les différentes approches utilisées pour le suivi de culture par caméra monoculaire.

Une méthode consiste à seuiller son image initiale et à effectuer ensuite une transformation de Hough<sup>4</sup> sur l'ensemble des points restants (en affectant plus de poids aux plants situés dans le second plan de l'image) comme dans [5]. Une autre approche se base sur un seuillage dynamique<sup>5</sup> Une fois binarisée les auteurs de l'article [6] ne gardent que le centre des rangées complètement visibles sur chacune des lignes (bloc de couleur blanche bordé de noir). Enfin, ils utilisent la méthode de Hough pour déterminer la position du point de fuite des rangées. Dans cet article, il faut tout de même noter un point important, les auteurs commencent par réduire l'étude à la partie centrale de l'image (ne couvrant que 23 % de l'image d'origine) ce qui est discutable.

Cette méthode présuppose donc de pouvoir distinguer au moins deux rangées et que celles-ci soient pleines. Par conséquent, elle peut tout à fait être appliquée sur des cultures en rangées denses mais souffrira notamment du recouvrement entre les rangées. La figure suivante présente l'implémentation de cet algorithme sur une culture dense (type lavande).

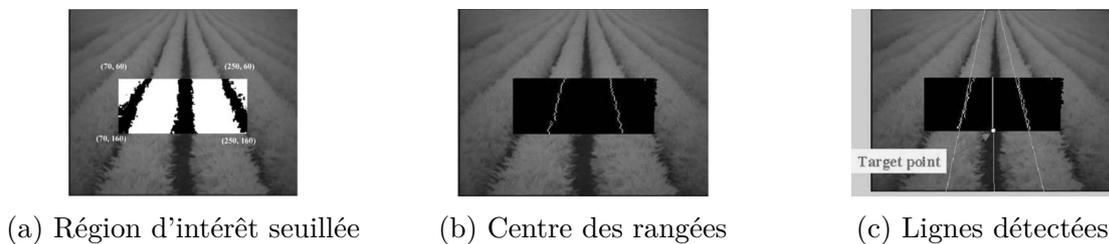


FIGURE III.1 – Détection de ligne par méthode de Hough [6]

Une autre procédure propose de modifier l'image de telle sorte qu'elle soit représentative d'une vue de dessus des cultures. Il suffit ensuite de trouver des lignes parallèles dans la nouvelle image. Dans le document [7] celle-ci est ensuite transformée en nuances

4. Méthode permettant de trouver les lignes qui passent par un maximum de points.

5. La valeur du seuil dépend de la valeur moyenne des pixels de l'image considérée.

de gris puis divisée en trois sections (où sont censées être situées les rangées). Les trois blocs ainsi formés sont ensuite superposés. Enfin, la méthode de Hough est appliquée sur l'imagette puis les lignes trouvées sont replacées sur l'image d'origine.

L'avantage majeur de cette méthode est de pouvoir obtenir un résultat correct même lorsque le sol comporte de nombreux adventices<sup>6</sup>. Cependant, pour arriver à ce résultat, on superpose des sections de l'image de façon arbitraire ce qui suppose une parfaite connaissance de la position du robot. De ce fait, l'algorithme sera impacté si le robot dévie légèrement de sa position souhaitée (patinage par exemple). De plus, le temps d'exécution de l'algorithme est important (entre 0.5 et 1.3 secondes). Les images suivantes présentent les étapes phares de cette méthode, la reprojection dans l'espace monde, la séparation et superposition des différents blocs et la détection de ligne dans l'imagette.

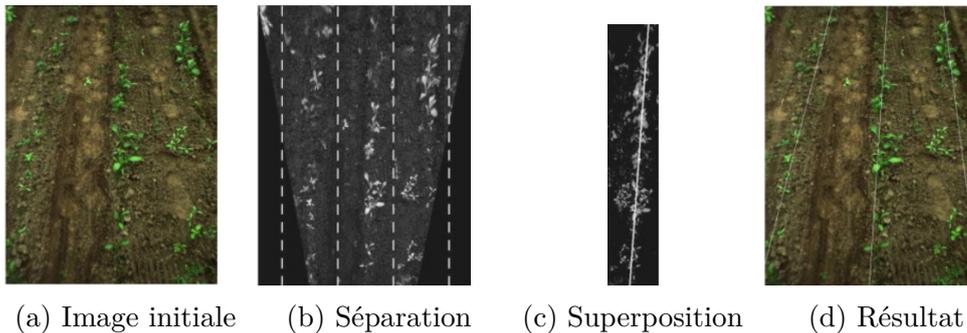
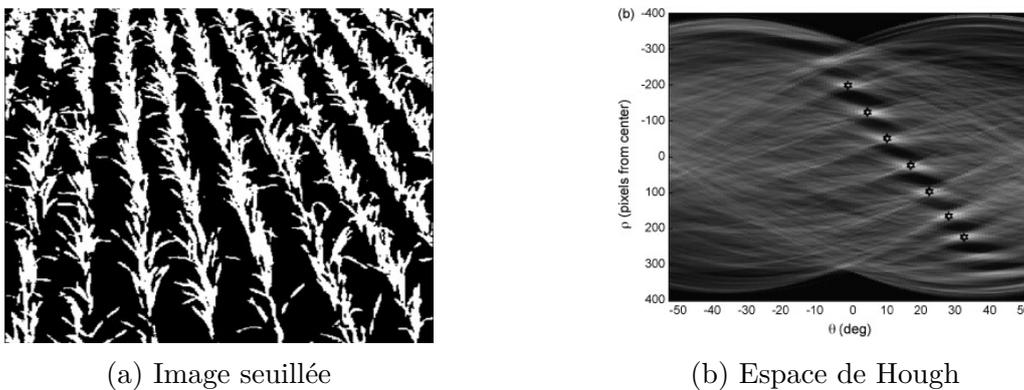


FIGURE III.2 – Détection de ligne par superposition [7]

Un moyen intéressant de détermination de la position et l'orientation des lignes de cultures dans l'image est exposé dans [8]. Pour ce faire, l'auteur part d'une vue large des rangées (au moins 6 visibles) qu'il segmente à partir de la fonction d'excès de vert<sup>7</sup>. Une première transformée de Hough est appliquée sur l'image binarisée révélant les rangées recherchées. Mais l'auteur va plus loin en effectuant une transformée de Hough dans l'espace du même nom ce qui permet d'avoir une information sur le point de fuite et sur la largeur des rangs obtenant ainsi un résultat cohérent. En effet, une propriété fondamentale de l'espace de Hough réside dans le fait que les points situés sur une même ligne dans cet espace se croisent au même point dans l'image initiale.



6. Plante se développant dans un endroit sans y avoir été intentionnellement installée.

7. Fonction permettant d'isoler les pixels tels que  $2V - B - R > 0$  où R,V,B sont les composantes de couleurs rouges, vertes et bleues du pixel considéré.

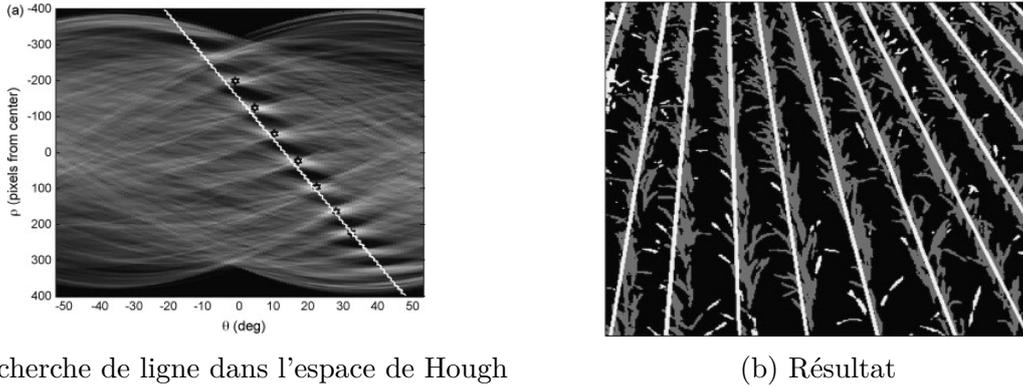


FIGURE III.4 – Détection de ligne par double méthode de Hough [8]

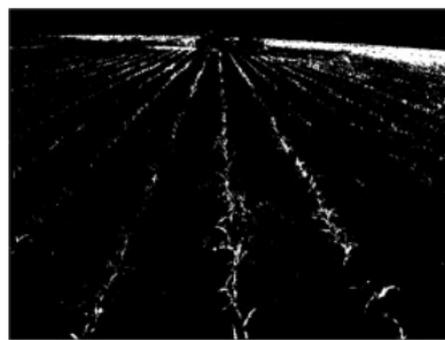
L'auteur de l'article [9] utilise sensiblement la même approche mais n'effectuant pas une seconde transformée de Hough, de nombreuses lignes incohérentes apparaissent dans son résultat final. Pour les filtrer, l'auteur tente de modifier légèrement les droites pour les faire coïncider en un même point (le point de fuite). Lorsque l'écart est trop important la ligne est jugée aberrante. Une approche semblable est détaillée dans [10]. La différence réside dans l'utilisation d'un partitionnement en K-moyennes pour déterminer le point de fuite.

Une méthode de Hough revisitée est expliquée dans [11]. Dans celle-ci (appelée *Random Hough Transform*), plutôt que de compter le nombre de points passant par chaque droite, deux points sont tirés au sort parmi les points d'intérêt. En réitérant l'opération, on obtient un ensemble de paramètres qu'il suffit de filtrer (en utilisant la médiane par exemple). Bien que cette technique semble plus rapide qu'une transformée de Hough classique, celle-ci impose de n'observer qu'une seule rangée à la fois.

Une façon de détecter les lignes de culture lorsque de nombreux adventices sont présents est également exposée dans la littérature. Pour cela, l'image initiale est en premier lieu normalisée puis seuillée deux fois. Le premier seuillage permet d'isoler les végétaux (en se basant sur la méthode d'Otsu<sup>8</sup>), le second permet de distinguer les plants des adventices (en se basant sur la fonction d'excès de vert). Enfin, pour trouver les lignes de culture, une régression linéaire est faite autour de la position moyenne des rangées (modèle fixe créé par apprentissage) [12]. Cette méthode est certes très performante, mais le modèle utilisé impose que les variations de position des lignes dans l'image soit faible. Les images suivantes montrent le fonctionnement général de cet algorithme.



(a) Image initiale



(b) Image seuillée

8. Méthode de binarisation automatique basée sur la forme de l'histogramme de l'image.

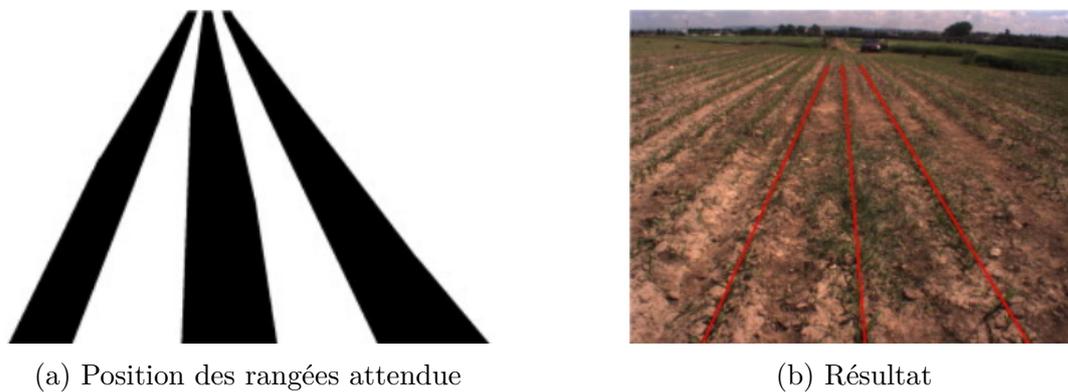


FIGURE III.6 – Détection autour d'un motif [12]

### 1.3 Choix d'une approche

#### Comparatif des méthodes de segmentation

À partir des observations précédentes, il est intéressant de comparer les différentes méthodes de segmentation d'une image. En effet, cette étape est fondamentale et nécessite un compromis entre rapidité et fiabilité des résultats. Le tableau suivant se propose donc de faire une comparaison des différentes approches envisagées pour cette opération afin de faire ressortir la plus adaptée à notre besoin.

	Rapidité	Résultat	Robustesse	Total
Seuillage	+++	-	- - -	-
Seuillage dynamique	++	+	-	++
Methode d'Otsu	-	++	+	++
K-Moyennes	-	Dépend du nombre de classes		Incertain
SVM	-	Dépend de l'apprentissage		+++
		++	++	

TABLE III.1 – Comparatif des méthodes de segmentation

Au vu de ce comparatif, notre choix s'est tourné vers la méthode du séparateur à vaste marge qui sera détaillée ultérieurement.

#### Approche utilisée

L'objectif du processus est de pouvoir identifier la position relative du robot par rapport aux différentes rangées de betteraves grâce à une caméra couleur. Par la suite, on se servira de cette information afin de générer une commande adéquate pour déplacer le robot. Pour pouvoir se localiser dans la culture grâce à une caméra, la solution la plus pertinente semble être de détecter les rangs de betteraves. Dans notre cas, le cahier des charges impose que :

- la localisation doit être la plus précise possible.
- le temps de réalisation doit être compatible avec une contrainte temps-réel.

- la méthode doit être robuste vis-à-vis des variations environnementales (rangée clairsemée, vibrations dues aux irrégularités du sol, etc).
- une collaboration de l'algorithme avec d'autres processus doit être facilement réalisable en vue d'une bonne intégration du système.

Au vu des exigences du cahier des charges et de la connaissance *a priori* de l'environnement, une approche *bottom-up* semble être à privilégier. En effet, il existe deux principes généraux pour une approche procédurale : l'approche dite descendante (*top-down*) et celle dite ascendante (*bottom-up*). Dans le premier cas, on part d'une quantité de données importante qu'il faut par la suite décomposer et trier afin d'obtenir un résultat. Dans le second cas, on cherche à confirmer une hypothèse grâce aux données des capteurs (ce qui nécessite des informations sur l'environnement dans lequel évolue le système). Pour illustrer ce propos, on peut imaginer que l'on cherche à se remémorer la commande de systèmes non linéaires. L'approche *top-down* consisterait à écumer tous les livres d'une bibliothèque en espérant trouver une information utile tandis que l'approche *bottom-up* cantonnerait cette étude uniquement aux documents portant sur l'automatisme. Dans ce cas, il semble évident que la seconde méthode est la plus adaptée, cependant, elle nécessite une connaissance *a priori* (il faut savoir que la commande de systèmes non linéaires est une branche de l'automatisme).

L'approche développée dans [4] ainsi que dans [13] semblent particulièrement intéressantes et seront implémentées par la suite après une légère adaptation. L'idée de ces techniques est de définir des zones de recherche restreintes (à partir des connaissances de l'environnement) dans lesquelles l'objet à détecter devrait se situer ; puis de procéder de manière stratégique afin d'obtenir les meilleurs résultats possibles en évitant une recherche complète dans chaque zone. Un des intérêts majeurs de ces méthodes est de réduire le temps de calcul en ne s'intéressant qu'aux parties intéressantes de l'image tout en assurant un résultat cohérent avec la réalité.

Pour pouvoir retrouver ces lignes, on estime (à partir des connaissances *a priori*) l'état du robot. On cherchera ensuite les rangées autour de leur position estimée afin de trouver l'état réel du système. Pour cela, il nous faut un modèle.

L'avantage majeur de cette méthode est d'avoir un résultat appartenant au modèle souhaité. En effet, comme les différents paramètres sont bridés par leur écart-type (Eq.III.4), on ne peut pas trouver un résultat en dehors de cet intervalle. Il faut donc que ces variations soient quantifiées afin de trouver un compromis entre souplesse et réalisme. Pour une meilleure portabilité et intégration, le développement du projet est fait sous ROS (cf ROS )

## 2 Construction de l'approche

### 2.1 Notations utilisées

Pour pouvoir modéliser notre problème, il nous faut d'abord introduire quelques notations. On notera par la suite :

- $X_0$  l'écart latéral du robot par rapport à la ligne centrale,
- $Y_0$  la distance du centre du robot à la caméra suivant la direction d'avance du robot,
- $Z_0$  la hauteur de la caméra par rapport au centre du robot,
- $\alpha$  l'angle de lacet,

- $\beta$  l'angle de tangage,
- $\delta$  l'angle de roulis,
- $\gamma$  l'angle de la caméra par rapport au plan de déplacement du robot,
- $L$  la distance séparant deux rangs,
- $k_x$  (resp  $k_y$ ) le nombre de pixels par unité de longueur suivant l'axe X (resp Y)
- $c_x$  (resp  $c_y$ ) la coordonnée en pixel du centre de l'image suivant X (intersection de l'axe optique avec le plan image), (resp Y)
- $\vec{U}$  l'axe orienté suivant la largeur du plan image dans le repère caméra,
- $\vec{V}$  l'axe de la caméra,
- $\vec{W}$  l'axe orienté suivant la hauteur du plan image dans le repère caméra,
- $u, v$  le système de coordonnées dans le repère image (origine en haut à gauche de l'image,  $u$  orienté vers la droite,  $v$  vers le bas)

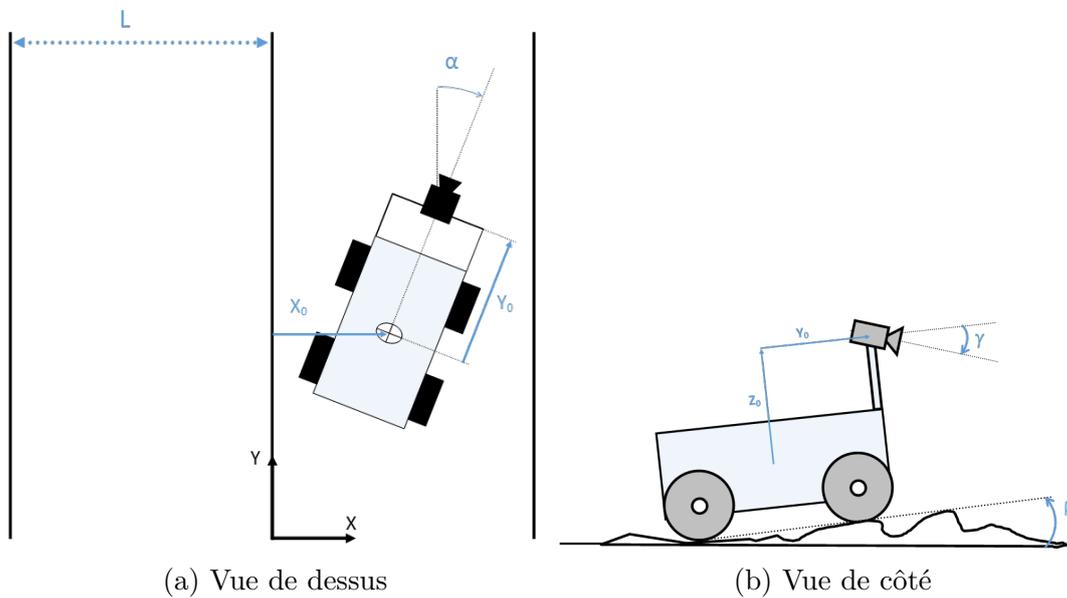


FIGURE III.7 – Notations utilisées

## 2.2 Modélisation du problème

Dans un premier temps, afin de déterminer les zones de recherche restreintes, on souhaite connaître la transformation permettant de passer du repère monde (Fig.III.7) aux coordonnées dans l'image.

Soit un point  $(X, Y, Z)$  défini dans l'espace, celui-ci aura comme coordonnées dans le repère caméra :

$$\begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} = P * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Où  $P = R_\gamma^{x'''} T_{Z_0}^{z'''} T_{Y_0}^{y'''} R_\delta^{y''} R_\beta^{x'} R_\alpha^z T_{X_0}^x$  (cf ) avec  $R_\theta^i$  la matrice de rotation d'angle  $\theta$  selon l'axe  $i$  et  $T_l^i$  la matrice de translation longueur  $l$  suivant l'axe  $i$ .

**NB** : Pour être générique, les valeurs des différents coefficients  $P_{i,j}$  ne sont pas détaillées

ici. Pour connaître leurs définitions, se reporter à la section [Définition de  \$P\_{i,j}\$](#) . Dans celle-ci, on remplacera  $\cos(\theta)$  par  $c_\theta$ ,  $\sin(\theta)$  par  $s_\theta$ ,  $\tan(\theta)$  par  $t_\theta$ .

D'après le modèle sténopé, le point  $(X, Y, Z)$  aura donc comme coordonnées  $(u, v)$  dans l'image :

$$\begin{aligned} u &= k_x \frac{U}{V} + c_x = k_x \frac{P_{11}X + P_{12}Y + P_{13}Z + P_{14}}{P_{21}X + P_{22}Y + P_{23}Z + P_{24}} + c_x \\ v &= k_y \frac{W}{V} + c_y = k_y \frac{P_{31}X + P_{32}Y + P_{33}Z + P_{34}}{P_{21}X + P_{22}Y + P_{23}Z + P_{24}} + c_y \end{aligned} \quad (\text{III.1})$$

Connaissant *a priori* l'environnement dans lequel évolue le robot, on peut dire que les rangées de betterave sont à la fois parallèles et régulièrement espacées (d'une longueur  $L$ ). D'après la modélisation, on peut donc dire qu'un point appartenant à une rangée de betterave vérifie  $Z = C^{te}$  et  $X = \lambda L$ ,  $\lambda \in N$  (l'origine de l'axe  $X$  étant pris sur la rangée centrale).

Dans notre étude, on se limitera à la détection de points appartenant soit à la rangée centrale ( $\lambda = 0$ ) soit aux rangées latérales ( $\lambda = \pm 1$ ). On souhaiterait donc obtenir dans l'image une droite d'équation  $u = Av + B$  correspondant à chacune des lignes. Pour cela, il faut étudier la variation de  $\frac{U}{V}$  et  $\frac{W}{V}$  (ou de  $u$  et  $v$ ) en fonction de la variable  $Y$ .

Ceci revient à exprimer  $\frac{U}{V}$  en fonction de  $\frac{W}{V}$  en utilisant la variable  $Y$  comme pivot.

D'après [Note de calcul pour la modélisation](#), il vient :

$$\begin{aligned} A &= \frac{k_x C}{k_y}, \\ B &= c_x + k_x \left( D - C \frac{c_y}{k_y} \right). \end{aligned} \quad (\text{III.2})$$

Avec

$$\begin{cases} M = P_{11}X + P_{13}Z + P_{14}, \\ N = P_{21}X + P_{23}Z + P_{24}, \\ O = P_{31}X + P_{33}Z + P_{34}, \\ C = \frac{MP_{22} - NP_{12}}{OP_{22} - NP_{32}}, \\ D = \frac{P_{12}}{P_{22}} - \left( \frac{MP_{22} - NP_{12}}{OP_{22} - NP_{32}} \right) \frac{P_{32}}{P_{22}}, \end{cases}$$

Il s'agira donc par la suite de retrouver les droites pour  $X \in \{-L; 0; L\}$ .

### 2.3 Architecture de l'approche

Comme évoqué précédemment, l'approche consistera à faire une hypothèse sur la position du robot que l'on corrigera à l'aide d'observations. Pour cela, on utilisera un filtre de Kalman. Une fois la position retrouvée, on estimera la position suivante autour de laquelle on fera une nouvelle recherche et ainsi de suite.

L'algorithme de recherche peut donc se résumer en quelques étapes :

- La supposition faite sur la position du robot (dans le repère monde),
- Le passage de cette hypothèse dans le repère image,
- La recherche dans les zones d'intérêt et correction,
- L'extraction de la position corrigée,
- La prédiction sur l'état suivant,

Le schéma suivant (Fig.III.8) représente l'algorithme utilisé pour la détection. Celui-ci sera détaillé dans les parties suivantes.

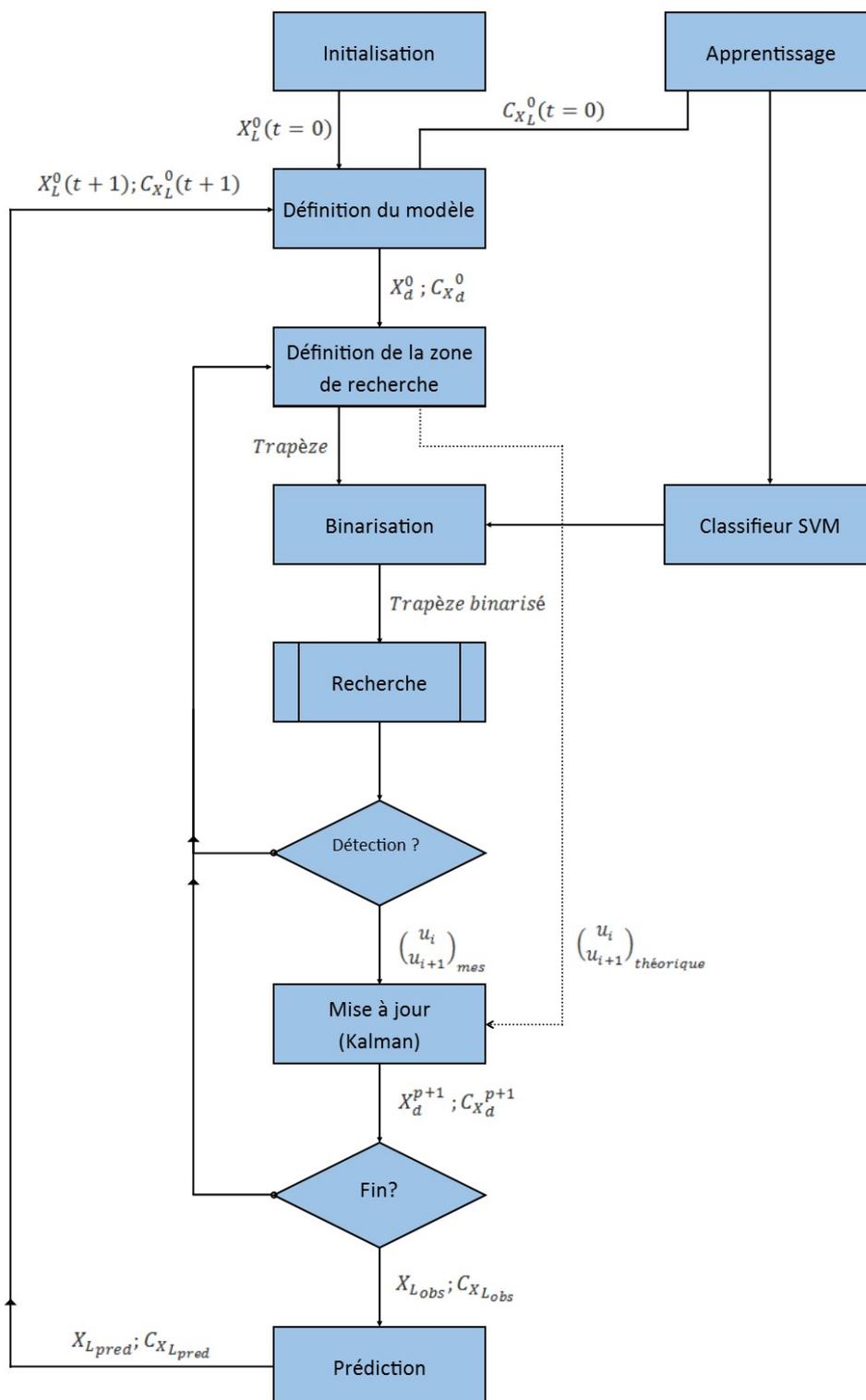


FIGURE III.8 – Schéma de l'algorithme utilisé

Pour pouvoir effectuer ces étapes, on a donc besoin de créer un modèle. Ce modèle est constitué d'un vecteur composé d'un ensemble de paramètres ainsi que d'une matrice de covariance associée caractérisant la dispersion possible de ces paramètres. C'est grâce à cette dispersion qu'il sera possible de restreindre notre espace de recherche. Comme la plupart du temps les rangées de betteraves sont semées à l'aide d'une machine, on suppose que l'inter-rang est constant. De plus, l'angle formé entre l'axe d'avance du robot et l'axe de la caméra (noté  $\gamma$ ) varie très peu. En revanche, les paramètres  $(X_0; \alpha; \beta; \delta)$  sont variables :

- $(X_0; \alpha)$  correspondent au déplacement normal du robot ;
- $(\beta; \delta)$  reflètent les défauts de forme du terrain ;

On construit donc les éléments suivants :

- $\underline{X}_l = (X_0; \alpha; \beta; \delta)^t$  le vecteur composé des variables définissant la position du véhicule mais aussi la géométrie du champ,
- $C_{X_l}$  la matrice de covariance associée au vecteur  $\underline{X}_l$ ,
- $\underline{X}_d = (u_{1g}; \dots; u_{ng}; u_{1c}; \dots; u_{nc}; u_{1d}; \dots; u_{nd})^t$  le vecteur représentant le modèle,
- $C_{X_d}$  la matrice de covariance associée au vecteur  $\underline{X}_d$ ,
- $(u_{1g}; \dots; u_{ng})$ ,  $(u_{1c}; \dots; u_{nc})$  et  $(u_{1d}; \dots; u_{nd})$  les abscisses respectives des rangées de gauche, centrale et de droite dans l'image pour différentes hauteurs  $(v_1; \dots; v_n)$ ,

NB : Pour pouvoir créer la matrice  $C_{X_l}$ , il faut prendre en compte les variations des différents paramètres entre deux instants. En effet, il est probable que le robot ait patiné entre deux images successives ce qui nous force à prendre en compte ces variations hypothétiques sur  $X_0$  et  $\alpha$ . De plus, les irrégularités du terrain reflétées par  $\beta$  et  $\delta$  sont bornées grâce à une phase d'apprentissage.

NB : On crée un vecteur  $\underline{V} = (v_1; v_2; v_3, \dots, v_n)^t$  afin de séparer l'image en plusieurs zones, par conséquent, l'image peut être *découpée* en zones d'intérêt. On verra par la suite que ces zones se réduiront drastiquement au cours de la phase de recherche permettant un gain de temps conséquent.

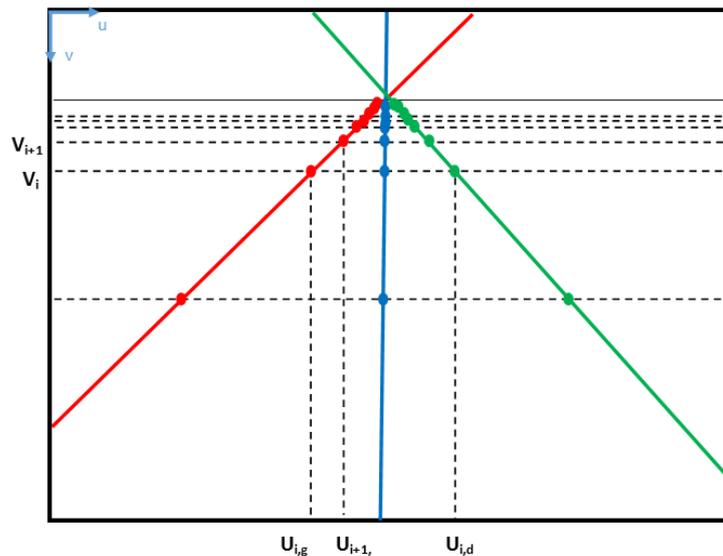


FIGURE III.9 – Notations dans l'espace image

### 3 Initialisation et création des zones de recherche

#### 3.1 Hypothèse initiale sur la position / construction de $X_l^0$

L'étape initiale de l'algorithme consiste à faire une hypothèse sur l'état du robot. Pour cela on initialise le vecteur  $\underline{X}_l$  à une valeur  $\underline{X}_l^0 = (X_0^0; \alpha^0; \beta^0; \delta^0)^t$  proche de l'état réel du robot.

#### 3.2 Passage dans le repère image / construction de $X_d^0$

Pour accélérer les recherches, on découpe tout d'abord l'image suivant des lignes horizontales représentant des zones de même étendue (Fig.III.9 ; Fig.III.10).

Comme  $v = k_y \frac{P_{31}X + P_{32}Y + P_{33}Z + P_{34}}{P_{21}X + P_{22}Y + P_{23}Z + P_{24}} + c_y$  on peut créer un vecteur  $V = (v_1; v_2; v_3, \dots, v_n)^t$  pour  $X = 0$ ,  $Z = C^{te}$  et différentes valeurs de  $Y$ .

Connaissant les paramètres estimés de position du robot, de la caméra et la géométrie de la route,  $(X_l^0)$  on est à même de définir le vecteur moyen  $X_d^0$  des positions des bette-raves dans l'image (Fig.III.11). Une fois le vecteur  $V = (v_1; v_2; v_3, \dots, v_n)^t$  fixé pour une image donnée, on crée le vecteur  $\underline{X}_d^0 = (u_{1g}; \dots; u_{ng}; u_{1c}; \dots; u_{nc}; u_{1d}; \dots; u_{nd})^t$  à partir des paramètres estimés  $\underline{X}_l^0$ .

Or, d'après Eq.III.2 on sait que  $u = Av + B$ , il vient donc :

$$\begin{cases} u_{i,g} = A_{x=-L} v_i + B_{x=-L} \\ u_{i,c} = A_{x=0} v_i + B_{x=0} \\ u_{i,d} = A_{x=+L} v_i + B_{x=+L} \end{cases} \quad (\text{III.3})$$



FIGURE III.10 – Droites horizontales



FIGURE III.11 – Lignes estimées

#### 3.3 Construction des zones de recherche

La position de chacun des points  $(u_{1g}; \dots; u_{ng}; u_{1c}; \dots; u_{nc}; u_{1d}; \dots; u_{nd})$  n'étant pas connue précisément (c'est ce que l'on cherche à déterminer), on construit un intervalle de recherche à partir de la matrice de covariance  $C_{X_d}$  (Fig.III.12). Comme dans [4], on considère que les paramètres constituant  $\underline{X}_l$  sont indépendants.

$C_{X_d}$  est donc obtenue à partir de la matrice Jacobienne  $J_C$  :  $C_{X_d} = J_C C_{X_l} J_C^t$

$$J_C = \begin{pmatrix} \frac{\partial u_{1g}}{\partial X_0} & \frac{\partial u_{1g}}{\partial \alpha} & \frac{\partial u_{1g}}{\partial \beta} & \frac{\partial u_{1g}}{\partial \delta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_{ng}}{\partial X_0} & \frac{\partial u_{ng}}{\partial \alpha} & \frac{\partial u_{ng}}{\partial \beta} & \frac{\partial u_{ng}}{\partial \delta} \\ \frac{\partial u_{1c}}{\partial X_0} & \frac{\partial u_{1c}}{\partial \alpha} & \frac{\partial u_{1c}}{\partial \beta} & \frac{\partial u_{1c}}{\partial \delta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_{nc}}{\partial X_0} & \frac{\partial u_{nc}}{\partial \alpha} & \frac{\partial u_{nc}}{\partial \beta} & \frac{\partial u_{nc}}{\partial \delta} \\ \frac{\partial u_{1d}}{\partial X_0} & \frac{\partial u_{1d}}{\partial \alpha} & \frac{\partial u_{1d}}{\partial \beta} & \frac{\partial u_{1d}}{\partial \delta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_{nd}}{\partial X_0} & \frac{\partial u_{nd}}{\partial \alpha} & \frac{\partial u_{nd}}{\partial \beta} & \frac{\partial u_{nd}}{\partial \delta} \end{pmatrix} C_{X_i} = \begin{pmatrix} \sigma_{X_0}^2 & 0 & 0 & 0 \\ 0 & \sigma_\alpha^2 & 0 & 0 \\ 0 & 0 & \sigma_\beta^2 & 0 \\ 0 & 0 & 0 & \sigma_\delta^2 \end{pmatrix} \quad (\text{III.4})$$

NB : Les calculs sont détaillés en annexe ([Calcul de la matrice Jacobienne  \$J\_C\$](#) ).

Cette matrice  $C_{X_d}$  permet de prendre en compte la cohérence entre les mesures des rangées grâce aux corrélations entre les composantes du vecteur  $\underline{X_d}$ ; elle servira notamment lors de la phase de mise à jour.

Dans l'image suivante, on représentera les intervalles  $[u_i - \sigma_{u_i}; u_i + \sigma_{u_i}]$  pour chaque hauteur  $v_i$ . Pour éviter les problèmes de chevauchement, seuls les intervalles de la rangée du milieu sont représentés

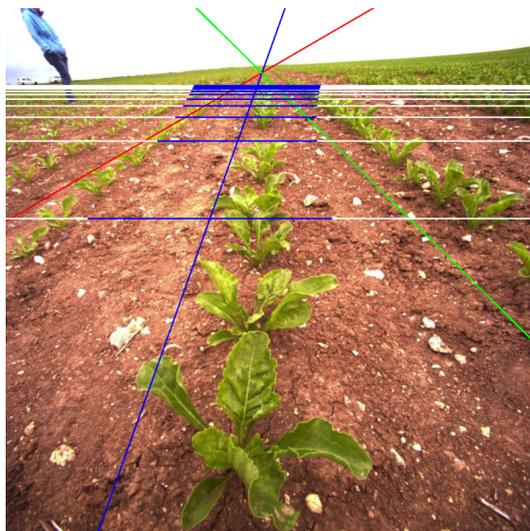


FIGURE III.12 – Intervalle de confiance pour la rangée centrale

## 4 Phase de recherche

### 4.1 Principe

Comme évoqué précédemment, il s'agit ici de retrouver les rangées de betteraves autour de leur position supposée.

NB : On distinguera par la suite deux trapèzes pour la même zone.

- Le premier sera celui dans lequel on cherchera les plants; il prendra en compte les variations autorisées de chaque paramètre mais également la taille des betteraves.

- Le second (inclus dans le premier) sera celui dans lequel devra se trouver le résultat, il ne dépendra que des variations autorisées de chaque paramètre.

Dans un premier temps, pour un soucis de simplicité et de lisibilité, on ne représentera pour l'instant que le trapèze de résultat. Celui-ci sera formé par deux intervalles de recherche ( $[u_i - \sigma_{u_i}; u_i + \sigma_{u_i}]$ ) consécutifs. Les  $\sigma_{u_i}$  ne sont autres que les composantes de la diagonale de la matrice  $C_{X_d}$  (Eq.III.4). Pour éviter le chevauchement, seuls les intervalles de recherche de la rangée centrale sont représentés sur l'image Fig.III.13.

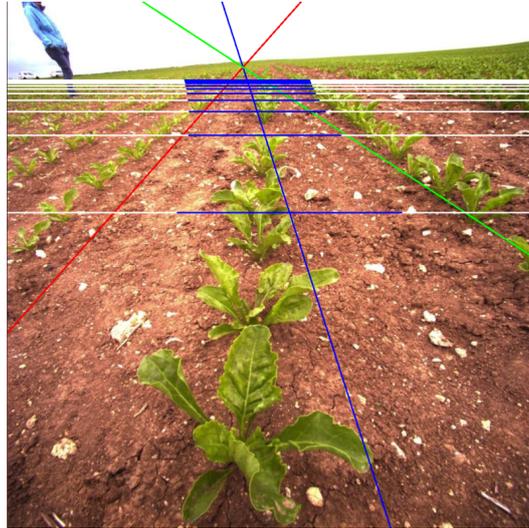


FIGURE III.13 – État initial avec hypothèse

Pour déterminer la position de la rangée, on effectue tout d'abord un filtre de couleur verte sur l'image (Fig.III.14). Comme évoqué précédemment, bien que coûteuse, cette étape est cruciale dans la future étape de détection. L'approche retenue est d'utiliser un seuillage sur chaque pixel dans l'espace RVB<sup>9</sup> grâce à la méthode de discrimination par séparateur à vaste marge (cf [Principe de discrimination par SVM](#)).

Un des intérêts majeurs de cette méthode est de ne binariser que les zones d'intérêt (les trapèze ci-dessus) ce qui permet de réduire énormément le temps de calcul permettant donc de garantir l'aspect temps réel. Une fois l'image partiellement binarisée, on cherche des informations utiles (point, segment..) dans la zone d'intérêt. Une fois obtenue, l'information permet de mettre à jour le modèle  $X_d$  et  $C_{X_d}$ . Par la suite nous noterons  $p$  le nombre de détections effectuées. La mise à jour permet de passer du modèle  $(X_d^p, C_{X_d}^p)$  à  $(X_d^{p+1}, C_{X_d}^{p+1})$  par l'intermédiaire d'une mesure  $\hat{x}$  et d'une covariance  $C_x^p$  grâce à la formule suivante :

$$\begin{cases} X_d^p = X_d^{p-1} + K_d[\hat{x} - x] \\ C_{X_d}^p = (I - K_d H_d) C_{X_d}^{p-1} \end{cases} \quad (\text{III.5})$$

9. Espace dans lequel les couleurs sont décomposées en trois composantes : le Rouge, le Vert et le Bleu. Il existe aussi d'autres espace semblables tels que BVR, YUV, HSV...

Avec

$$\begin{cases} K_d & = C_{X_d}^{p-1} H_d^t [H_d C_{X_d}^{p-1} H_d^t + C_x^p] \\ x & = (u_{i,j}; u_{i+1,j})^t, j \in \{g, c, d\} \\ H_d & \text{tel que } x = H_d X_d \end{cases}$$

Ce système n'est autre qu'une adaptation du filtre de Kalman pour lequel l'étape de prédiction<sup>10</sup> n'est volontairement pas pris en compte.

Afin de réduire rapidement nos zones de recherche, on effectue les deux premières détections sur des rangées différentes. De ce fait, comme on peut le remarquer qu'après deux détections et évolutions (Fig III.17) les zones de recherche sont très faibles ce qui permet de réduire considérablement le temps d'exécution. En effet, le modèle étant très contraint (les rangées sont droites, parallèles et séparées d'une même distance), lorsque l'on connaît la position et l'orientation de deux portions de lignes, on peut déterminer le croisement de celle-ci et donc obtenir des informations sur les autres rangées.

NB : Ce phénomène de contrainte est reflété dans la matrice de covariance  $C_{X_d}$ . Les termes hors diagonale correspondent à l'influence d'un paramètre sur un autre. Ainsi, un terme aura un impact fort sur les termes de la même rangée que lui mais également ceux spatialement proches. (eg : le terme  $u_{3,c}$  aura une influence forte sur tous les  $u_{i,c}$  mais aussi sur  $u_{3,g}$  et  $u_{3,d}$ ).

Dans les paragraphes suivants, nous allons comparer deux méthodes distinctes d'extraction d'information afin de mettre en lumière les forces et les limitations de chacune d'entre-elles.

## 4.2 Détection par régression linéaire

La première approche consiste simplement à trouver la droite passant *au mieux* par l'ensemble des nuages de points verts trouvés dans chaque zone. Ces nuages étant denses, les méthodes de régression robustes ne permettent pas de filtrer les points aberrants et sont donc d'une efficacité moindre.

On peut donc opter pour une régression linéaire simple qui allie simplicité et vitesse d'exécution.

Pour des raisons de clarté, l'image de cet exemple sera complètement binarisée, les bette-raves contenues dans le trapèze d'intérêt seront légèrement grisées, la droite obtenue sera bleutée.

---

10. Pour la phase de recherche uniquement.



FIGURE III.14 – Détection dans l'image binarisée

A partir des segments ainsi obtenus, on met à jour le vecteur de modèle à l'aide de l'équation III.5.

L'image Fig.III.15 donne un aperçu l'état initial. Celle-ci comporte :

- les trois lignes qui devraient se superposer avec les rangs de betteraves si le robot était dans la configuration estimée,
- des lignes horizontales délimitant les zones dans laquelle le résultat est attendu pour chaque ligne.

NB : Par soucis de clareté, seules les lignes bleues sont représentées sur la figure Fig.III.15.

Après une première détection sur la rangée centrale (Fig.III.16) on constate que l'étendue des zones a nettement diminuée, ce qui signifie que la variation autorisée dans l'espace de résultat rapetisse.

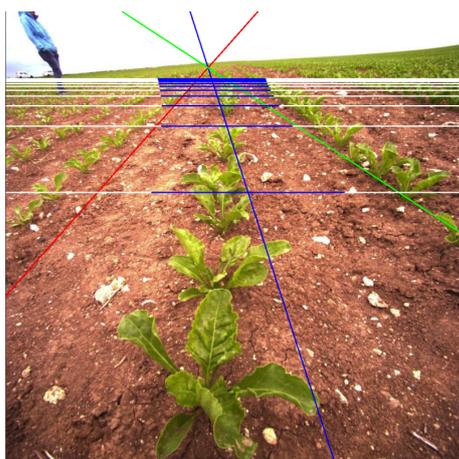


FIGURE III.15 – État initial



FIGURE III.16 – État après première détection

En itérant le processus sur la rangée de droite (Fig.III.17), on constate que le trapèze de résultat est devenu minuscule, il ne variera presque plus lors des autres recherches, comme on peut le constater sur l'image Fig.III.18 qui présente le résultat obtenu après les 27 recherches.



FIGURE III.17 – État après seconde détection



FIGURE III.18 – Résultat après 27 détections

A partir de l'ensemble des recherches, on est en mesure de tracer les lignes recalées à l'aide du filtre de Kalman spatial (Fig. III.19).

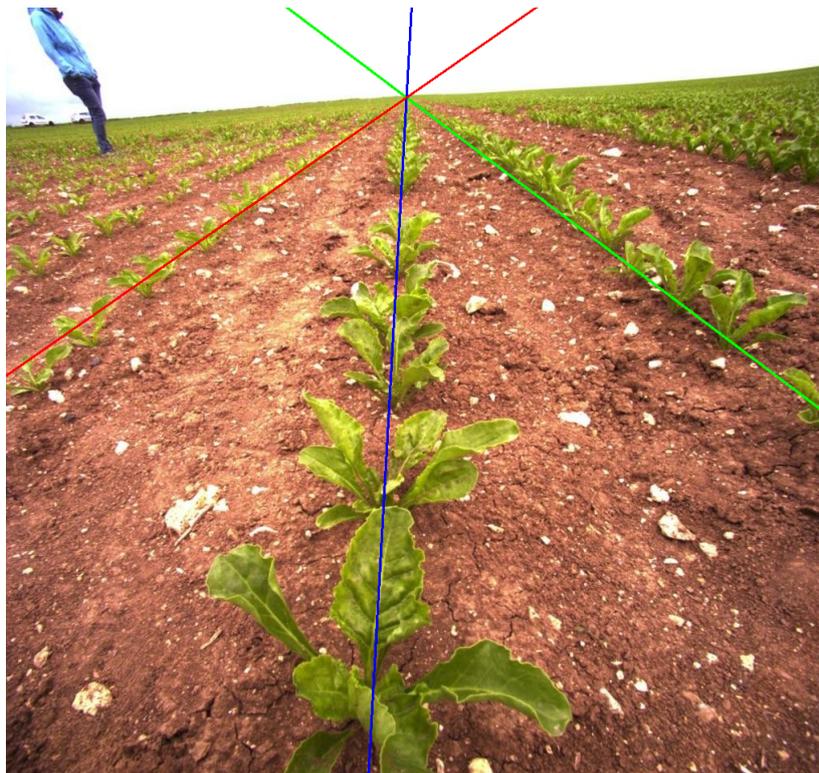


FIGURE III.19 – Lignes retrouvées par régressions linéaires

En regardant les images de l'évolution de la recherche, on voit que les deux premières détections sont fondamentales pour la suite de la localisation. En effet, l'intervalle de recherche devient très faible lorsque deux détections ont été faites sur des rangées différentes. Ce phénomène est dû à la forte redondance du modèle (les lignes sont équidistantes et parallèles dans le monde réel ; connaître la position et l'orientation de deux lignes permet de connaître celle de toutes les autres). Ainsi, si l'une d'entre elle est erronée (mais qu'elle reste cohérente avec l'intervalle de résultat), celle-ci aura tendance à dévier le modèle

de la réalité. Or comme évoqué, la méthode de régression linéaire est certes très rapide mais n'est pas robuste. De ce fait, si les premières betteraves ne sont pas symétriques, la régression aura tendance à biaiser le modèle entraînant donc la perte d'intégrité de la localisation. Pour palier ce problème, on peut utiliser une méthode plus robuste.

### 4.3 Détection par méthode mixte

En étudiant de plus près l'image d'exemple ci-dessus (Fig. III.20), on peut s'apercevoir que les rangées latérales possèdent une bordure "intérieure" presque rectiligne. C'est pourquoi, il peut sembler intéressant de détecter les lignes sur les points les plus à l'intérieur du trapèze à l'aide d'un gradient horizontal. Pour la rangée centrale en revanche, cela est plus délicat. La stratégie utilisée consiste comme dans l'article [13] à calculer le centroïde de chaque zone puis de faire une régression sur l'ensemble. De cette façon, il est plus simple de supprimer les points aberrants tout en ayant une information globale.

**NB** : Il est important de noter que la mise à jour du vecteur de modèle n'intervient donc qu'en fin de recherche, c'est à dire, après avoir étudié les neuf trapèzes de la ligne. On gagne donc en précision au détriment de la vitesse d'exécution.

On peut ainsi modifier légèrement notre vecteur de modèle en remplaçant  $\underline{X}_d = (u_{1g}; \dots; u_{ng}; u_{1c}; \dots; u_{nc}; u_{1d}; \dots; u_{nd})^t$  par  $\underline{X}_d = (u_{1g}; \dots; u_{ng}; A_c; B_c; u_{1d}; \dots; u_{nd})^t$ , où  $(A_c; B_c)$  sont les coefficients de la rangée centrale dans l'image (Eq. III.2). L'exemple suivant est issu d'une image prise en conditions réelles (mouvement, résolution 640x480 pixels).

Comme dans l'exemple précédent, l'image Fig. III.20 présente l'hypothèse faite sur l'état initial ; la figure Fig. III.21 montre le résultat après avoir étudié toute la rangée centrale ; la représentation Fig. III.22 montre l'évolution des zones de recherche après avoir analysé le premier trapèze dans la rangée latérale ; enfin l'image Fig. III.23 présente l'état du système après avoir étudié le premier trapèze de l'autre rangée latérale.

**NB** : Sur les images, les points jaunes représentent les barycentres pour la rangée centrale ; les trapèzes rouges sont les trapèzes de recherche et les bleus ceux de résultats admissibles.



FIGURE III.20 – État initial



FIGURE III.21 – État après première détection



FIGURE III.22 – État après seconde détection



FIGURE III.23 – État après troisième détection

Une fois encore, en examinant l'image Fig.III.24 représentant le résultat final, on constate que la convergence de l'algorithme est rapide puisque l'état définitif est presque identique à celui après 3 détections.



FIGURE III.24 – Résultats

NB :Il faut tout de même noter qu'ici, l'incertitude sur le paramètre  $A_c$  induit un étirement des trapèzes de la rangée centrale. Ce phénomène est d'autant plus fort que l'on s'éloigne du point de fuite.

## 5 Résultat de la localisation

Une fois la phase de recherche terminée, deux stratégies s'offrent à nous. On peut utiliser une commande :

- classique auquel cas, on doit repasser le modèle dans l'espace monde.
- référencée capteur et par conséquent ne travailler que dans l'espace image.

## 5.1 Commande classique

Comme évoqué précédemment, pour pouvoir contrôler le robot, il faut classiquement connaître son écart latéral et angulaire par rapport à sa trajectoire de référence (ici la position désirée correspond à l'alignement et au centrage du robot sur une rangée de culture). Ainsi, on a besoin d'extraire ces paramètres de l'image.

### Extraction des paramètres dans l'espace monde

Une fois la recherche aboutie, on obtient un vecteur de modèle modifié  $\underline{\hat{X}}_d$ . Or on cherche à connaître le vecteur  $\underline{\hat{X}}_l$  correspondant.

Bien que le passage de  $\underline{X}_l$  vers  $\underline{X}_d$  soit relativement facile, la transformation inverse est elle beaucoup plus complexe à calculer (car faisant intervenir des fonctions non linéaires). Cependant, on peut noter que :

$$\begin{cases} \underline{\hat{X}}_d \approx \underline{X}_d^0 + J_{C_{x_l^0}} (\underline{\hat{X}}_l - \underline{X}_l^0) \\ C_{\underline{\hat{X}}_d} = J_C C_{\underline{\hat{X}}_l} J_C^t \end{cases}$$

Cette équation correspond à l'approximation linéaire d'une fonction de plusieurs variables ( $J_{C_{x_l^0}}$  correspond à la matrice Jacobienne de la fonction calculée au point  $X_l^0$  que l'on a aperçu dans Eq.III.4 et calculée en B.1). De ce fait, en calculant la pseudo-inverse de la matrice Jacobienne ( $[J_C^t J_C]^{-1} J_C^t$  notée  $J_C^+$ ), il nous est possible de ramener les paramètres du modèle dans le repère monde (Eq.III.6). Cependant, cette opération est coûteuse en temps de calcul et suppose que  $\|\underline{\hat{X}}_l - \underline{X}_l^0\|$  soit faible. Pour éviter ces contraintes, il pourrait être intéressant de rester dans l'espace image. Cela nécessiterait donc un changement d'approche afin d'utiliser une commande référencée vision.

$$\boxed{\begin{cases} \underline{\hat{X}}_l \approx \underline{X}_l^0 + J_{C_{x_l^0}}^+ (\underline{\hat{X}}_d - \underline{X}_d^0) \\ C_{\underline{\hat{X}}_l} = J_C^+ C_{\underline{\hat{X}}_d} J_C^+ \end{cases}} \quad (\text{III.6})$$

## 5.2 Commande référencée vision

La commande référencée vision consiste à exprimer les tâches robotiques à réaliser, non plus dans l'espace des configurations, mais directement dans l'espace image sous la forme d'une relation locale entre le robot et son environnement. On cherche donc à définir des commandes en boucle fermée sur les informations provenant des capteurs et non plus sur la configuration du robot. Ainsi, toute tâche pouvant s'exprimer sous la forme d'une relation entre le robot et l'environnement sera réalisable par le biais de la commande référencée capteur. Toutefois, cette approche requiert l'utilisation de capteurs fiables et capables de fournir les mesures à une fréquence compatible avec la fréquence d'échantillonnage de l'asservissement.

Par manque de temps, cette approche n'a malheureusement pas pu voir le jour lors de ce stage.

## 6 Prédiction

Une fois la position du robot correctement déterminée, on réitère l'opération. Or plusieurs stratégies s'offrent à nous pour déterminer un point de fonctionnement  $X_l^0$  autour duquel les nouvelles recherches vont être effectuées.

Les prochains paragraphes présentent différentes méthodes allant de la plus simple à la plus précise. Les figures Fig.III.25, III.26, III.27, III.28 représentent ces différentes approches. Sur ces images, on peut distinguer deux robots, l'un en traits pleins représentant la position du robot mesurée, l'autre en pointillés désigne la prédiction utilisée par l'approche.

### 6.1 Cas $X_l^0$ fixe

Comme l'évoque le titre, on peut dans un premier temps choisir d'utiliser un point de fonctionnement fixe autour duquel on fera toutes nos recherches (le plus pertinent serait  $X_l^0 = (0; 0; 0; 0)^t$  c'est-à-dire lorsque le robot est parfaitement aligné avec la rangée à suivre). Le robot pouvant être dans des positions éloignées de ce point de référence, il est nécessaire de prendre en compte ces variations dans la matrice de covariance  $C_{X_l^0}$ . Ainsi les zones de recherche seraient fixes mais très étendues ce qui pose un problème. En effet, l'intérêt majeur de la méthode réside dans la rapidité d'exécution liée à des zones de recherche peu étendues. De plus, lorsque les variations autorisées par  $C_{X_l^0}$  sont trop importantes, il est possible que l'algorithme détecte de mauvaises lignes perturbant par conséquent tout le processus. Il faut donc trouver une méthode plus pertinente.

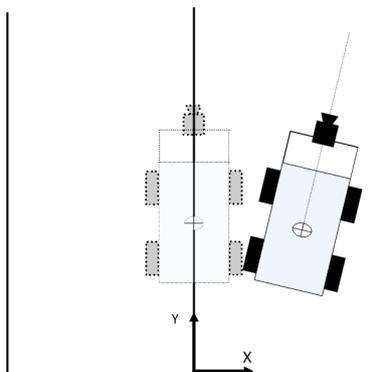


FIGURE III.25 – Représentation de la prédiction dans le cas où  $X_l^0$  est fixe

### 6.2 Cas $X_l^0 = \hat{X}_l$

Une autre méthode plus pertinente consiste à repartir de la mesure effectuée précédemment. En effet, comme la position précédente du robot est connue, on peut supposer que le robot est encore proche de celle-ci. Cela suppose donc que la mesure faite soit bonne mais également que le temps d'exécution de la localisation soit compatible avec la vitesse du système. Le dessin Fig.III.26 représente cette prédiction dans laquelle les deux robots sont superposés. Dans ce cas, on peut remplir la matrice  $C_{X_l^0}$  à l'aide d'une phase d'apprentissage, permettant de relever la variation maximale de chaque paramètre entre deux images. Bien que nettement plus efficace, une méthode similaire peut-être plus performante à moindre coût.

**NB** : On peut également envisager de regarder autour de  $(\hat{X}_0, \hat{\alpha})$  en supposant que  $(\beta, \delta)$  sont de moyenne nulle.

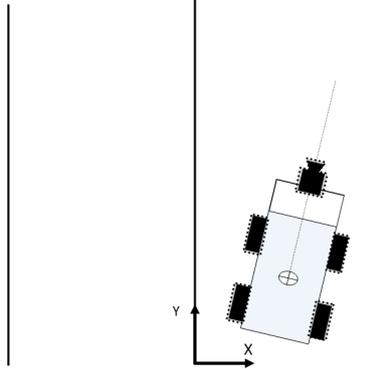


FIGURE III.26 – Représentation de la prédiction dans le cas où  $X_l^0 = \hat{X}_l$

### 6.3 Cas $X_l^0 = \hat{X}_l + vdt$

Dans le cas où la vitesse d'avance n'est pas compatible avec la fréquence de rafraîchissement, on peut cette fois-ci supposer que le robot avance en ligne droite depuis la dernière position mesurée. Cela revient à approximer la trajectoire suivie par le robot par la méthode d'Euler. De ce fait, le paramètre  $X_0$  varie de manière cohérente avec la mesure précédente comme le montre le dessin suivant.

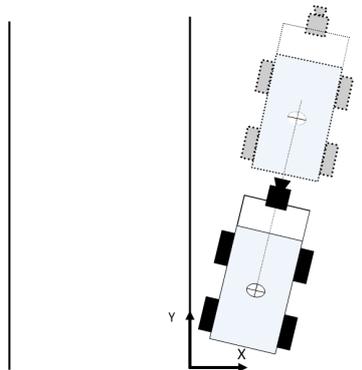


FIGURE III.27 – Représentation de la prédiction dans le cas où  $X_l^0 = \hat{X}_l + vdt$

### 6.4 Utilisation d'un filtre de Kalman

Bien que les méthodes précédentes soient faciles à mettre en place, elles souffrent toutes du glissement. En effet, les variations autorisées prenant en compte les variations mesurées entre deux images, notamment dues à la commande mais aussi au glissement qui est important dans le domaine agricole, celles-ci doivent soit être sur-estimées (pour garantir l'intégrité) soit être estimées régulièrement. C'est pourquoi les paramètres de la matrice de covariance restent importants ce qui induit une perte de temps importante voire même de mauvaises détections (toujours dues à des zones de recherche trop étendues). Pour palier ce problème, on peut envisager d'utiliser un filtre de Kalman.

En effet, le filtre de Kalman permet, à partir de l'odométrie du robot, de prédire sa position *a priori* permettant ainsi de réduire drastiquement l'incertitude sur la position de celui-ci (ie les coefficients de la matrice  $C_{X_l^0}$ ).

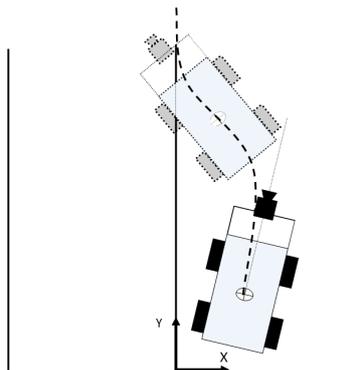


FIGURE III.28 – Représentation de la prédiction avec utilisation du filtre de Kalman

Bien que cette méthode semble la plus adaptée à notre problème, elle pas pû être mise en place lors de ce stage.

## 7 Synthèse sur l'approche développée

Au vu des premiers résultats obtenus, on peut en conclure que la démarche utilisée est prometteuse. En effet, la mise en place d'une approche à l'aide d'un modèle lié à un filtre de Kalman spatial (lors de la phase de recherche) a permis d'améliorer les performances du système de localisation.

Un premier algorithme avait été conçu mais restait très sensible aux vibrations ce qui induisait régulièrement des mesures aberrantes et très bruitées. Or un robot autonome nécessite une localisation parfaitement fiable avant de pouvoir être mis en service.

La méthode réalisée permet quant à elle une gestion des imperfections (rangées clairsemées, irrégularité du terrain, variation de luminosité, développement des betteraves hétérogène...) à l'aide des redondances du modèle. Effectivement, l'approche développée se base sur l'étude de différentes informations fortement corrélées ; on analyse trois lignes parallèles et équidistantes, donc, connaissant la position de deux d'entre elles, on peut déterminer celle de la troisième. Pour un traitement plus rapide de l'image, chaque ligne est elle-même morcelée en régions d'intérêt. Or due à la redondance du système, l'étude de l'une de ces régions donne également des informations sur les autres permettant une convergence rapide de l'approche.

Le point crucial de cette approche réside donc dans la conception et l'étude des régions d'intérêt. En effet, il faut trouver un équilibre entre étendue de la zone d'intérêt et représentativité de celle-ci. Or, la taille de cette dernière dépend notamment de la phase de prédiction. Une gestion astucieuse de cette étape permet donc de réduire la taille des blocs, accélérant par conséquent le processus.

Actuellement, on suppose que le robot ne s'est pas beaucoup déplacé depuis la dernière localisation ce qui est assez réducteur. C'est pourquoi, il est envisagé de mettre en place un second filtre de Kalman, temporel cette fois-ci, pour une amélioration non négligeable de l'ensemble du processus. Néanmoins, par manque de temps, il n'a malheureusement pas été possible de mettre en place une telle approche.

Aussi évident que cela puisse paraître, il est bon de rappeler que la méthode de détection doit être adaptée à la région d'intérêt, mais aussi à l'information recherchée. Lors du stage, il s'est avéré que la détection des bordures de la méthode mixte ([Détection par méthode mixte](#)) n'était pas adaptée à tous les cas de figure ce qui a retardé le projet.

## IV Mesures sur les plants

Pour vérifier le bon développement des betteraves, des séries de mesures peuvent être effectuées tout au long de leur croissance. Lorsque les betteraves sont à un stade peu avancé, le robot Jaguar est généralement utilisé. Cependant, quand les plants de betterave ont atteint un stade avancé, le recouvrement est tel que l'on ne peut plus manœuvrer un robot sans les endommager. Durant cette même période, ils peuvent contracter des maladies pouvant altérer la qualité du sucre. C'est pourquoi, il a été décidé de détecter les plants contaminés afin de pouvoir traiter la parcelle au mieux. Pour cela, on place un capteur au bout d'un bras robotisé afin que celui-ci puisse aller au plus près de la plante. Ce chapitre porte sur l'état actuel des moyens de mesure par caméra monoculaire ainsi que de la modélisation du bras en vue d'une commande active (commande faite en fonction de la perception). Celui-ci a été réalisé en étroite collaboration avec Camille DUBOS dont le sujet de thèse porte sur la définition d'une stratégie de déplacement du capteur s'appuyant sur la perception de la végétation.

Avant de pouvoir commander le bras robotisé, il est nécessaire de le modéliser. La section suivante s'intéressera à la modélisation du bras par la méthode de Denavit-Hartenberg.

### 1 Modélisation du bras UR5

Pour pouvoir commander le bras, on a besoin de connaître les équations régissant le mouvement. Avant toute chose, on définit deux espaces de travail distincts :

- l'espace articulaire qui a pour référence le repère lié à chaque articulation motorisée du robot.
- l'espace opérationnel qui a pour référence le repère lié à l'organe terminal du robot.

Cette section se divise en trois parties. La première permet d'établir une convention dans les notations servant de base pour la suite. La seconde et la troisième établissent respectivement le modèle géométrique direct et inverse, outils permettant de passer des coordonnées articulaires aux coordonnées opérationnelles et *vice versa*.

## 1.1 Description du bras

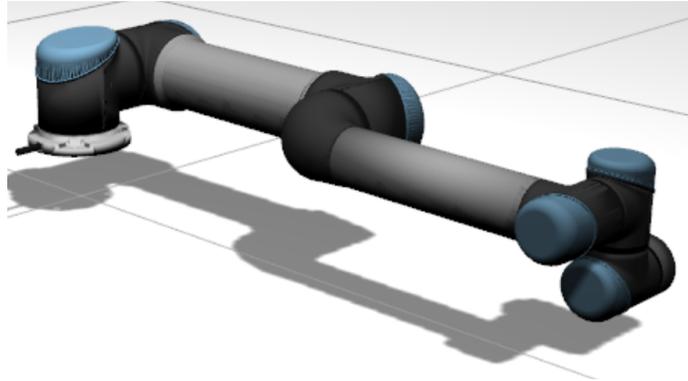


FIGURE IV.1 – Représentation du bras UR5

Pour pouvoir modéliser le bras, il est préférable d'utiliser une convention dans la notation de la position relative entre deux solides. Une des plus répandues est certainement celle de Denavit-Hartenberg dont un rappel est établi en annexe ([Méthode de Denavit-Hartenberg](#)).

D'après la modélisation précédente, le bras UR5 peut être représenté comme suit Fig.IV.2 avec les paramètres détaillés dans le tableau Tab.IV.1.

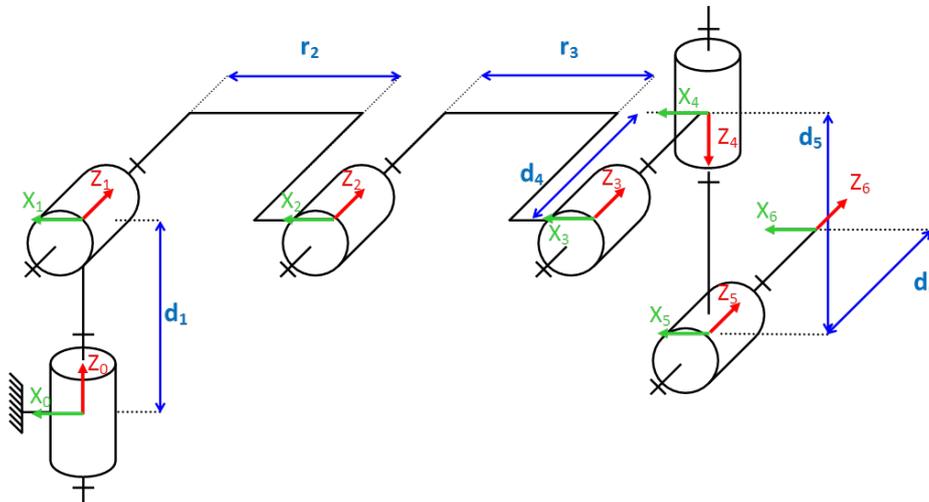


FIGURE IV.2 – Modélisation du bras UR5

i	$d_i$	$\theta_i$	$r_i$	$\alpha_i$
1	$d_1=0.0895$	$\theta_1$	0	$\frac{\pi}{2}$
2	0	$\theta_2$	$r_2=-0.425$	0
3	0	$\theta_3$	$r_3=-0.392$	0
4	$d_4=0.1092$	$\theta_4$	0	$\frac{\pi}{2}$
5	$d_5=0.0947$	$\theta_5$	0	$-\frac{\pi}{2}$
6	$d_6=0.0823$	$\theta_6$	0	0

TABLE IV.1 – Paramètres géométriques du robot UR5

## 1.2 Modèle géométrique direct

Le modèle géométrique direct est un outil qui permet de calculer la position et l'orientation de l'effecteur du robot en fonction des différents angles des liaisons. Ce modèle revient à calculer la matrice de passage entre le repère  $R_0$  et le repère de l'effecteur. Par la suite on notera  $T_{ij}$  la matrice de passage exprimant la position et l'orientation d'un repère  $R_j$  par rapport au repère  $R_i$ . D'après la convention de Denavit-Hartenberg, on a :

$$T_{i-1;i} = Rot_{\theta_i}^z \quad Trans_{d_i}^z \quad Trans_{r_i}^x \quad Rot_{\alpha_i}^x \quad (IV.1)$$

**NB** : Pour éviter de surcharger le rapport le détail des calculs sera établi en annexe ([Note de calcul du bras UR5](#)).

Dans notre cas, cette matrice est obtenue par la relation suivante :

$$T_{06} = T_{01}T_{12}T_{23}T_{34}T_{45}T_{56} = \begin{pmatrix} R_{06} & P_{06} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (IV.2)$$

## 1.3 Modèle géométrique inverse

Lorsqu'il est calculable, il existe plusieurs méthodes pour trouver le modèle géométrique inverse plus ou moins adaptées suivant la géométrie du robot :

- la méthode de Pieper s'adresse aux architectures simples (robots à 6 degrés de liberté, possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques).
- la méthode de Paul traite séparément chaque cas particulier et convient à la majorité des robots industriels.
- les méthodes géométriques consistent à trouver directement chaque variable articulaire à partir de considérations géométriques.

Dans notre cas, nous utiliserons la méthode de Paul. Par la suite, on notera  $U_i^d$  la position et l'orientation de l'effecteur dans le repère  $R_i$ . On définit la pose de l'effecteur

désiré par :

$$U_0^d = \begin{pmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Pour calculer les angles  $\theta_i$ , on compare généralement :

- $U_0^d$  avec  $T_{06}$ .
- $T_{10}U_0^d = U_1^d$  avec  $T_{16}$ .
- $T_{20}U_0^d = U_2^d$  avec  $T_{26}$ .
- $T_{30}U_0^d = U_3^d$  avec  $T_{36}$ .
- $T_{40}U_0^d = U_4^d$  avec  $T_{46}$ .
- $T_{50}U_0^d = U_5^d$  avec  $T_{56}$ .

Dans notre cas, il semble plus pertinent d'étudier :

- $T_{10}U_0^d$  avec  $T_{12}T_{23}T_{34}T_{45}T_{56}$
- $U_1T_6^5T_5^4$  avec  $T_1^2T_2^3T_3^4$

L'idée sous-jacente est de séparer le robot plan composé des liaisons 2,3,4 (cf Fig.IV.2) du reste du bras. C'est pour cela qu'on utilisera la matrice  $T_{14}$ .

NB : Par manque de temps, l'étude de la matrice Jacobienne (et des singularités du bras) n'a pas été faite.

## 2 Détection des plants

Avant de pouvoir s'approcher au plus près de chaque plant, il est fondamental de pouvoir les détecter convenablement. Or des travaux préliminaires ont été effectués. Le but de cette activité est d'arriver à dissocier les plants de betterave du sol mais également des adventices. Pour cela, on utilise ici aussi une discrimination l'aide d'un séparateur à vaste marge (noté SVM).

Une fois isolés, les plants de betterave peuvent être étudiés à l'aide de leurs propriétés géométriques et colorimétriques. Une illustration de ce processus est exposée dans les images suivantes. L'image IV.3 montre tout d'abord l'image initiale puis le masque<sup>1</sup> obtenu à l'aide du séparateur à vaste marge. Enfin on obtient l'image finale en appliquant le masque sur l'image initiale.

---

1. Résultat d'une opération de logique utilisée en informatique et en électronique pour sélectionner un sous-ensemble de bits à conserver ou au contraire à écraser.



(a) Image initiale



(b) Image binarisée par SVM



(c) Contour extrait

FIGURE IV.3 – Illustration de la détection des plants

### 3 Synthèse

A l'aide de la modélisation par la méthode de Denavit-Hartenberg et, plus particulièrement, du modèle géométrique inverse, on est en mesure de déplacer le bras pour atteindre une position désirée. Néanmoins, il reste à déterminer la stratégie de déplacement du bras. En effet, plusieurs tactiques sont envisageables pour passer d'une position à une autre. On peut, par exemple, définir la trajectoire la plus courte pour le déplacement, celle permettant d'éviter dans la mesure du possible les singularités, etc. Cette partie sera traitée dans la thèse de Camille DUBOS.

Grâce à la discrimination par séparateur à vaste marge, on peut également isoler les betteraves dans une image. Il est alors possible d'étudier différentes caractéristiques des plants et notamment leur position dans l'image. En couplant cette information avec le modèle géométrique du bras, on est à même de réaliser un asservissement visuel du bras.

# V Conclusion

Le but de ce stage était de participer au développement d'une plateforme robotisée autonome servant au phénotypage sur des parcelles de betteraves sucrières. Pour cela, différents points ont été abordés, comme le suivi de culture autonome, les mesures effectuées sur les plants ainsi que les prémices de la commande active du bras UR5 à partir de ces mêmes mesures.

Concernant le suivi de culture de façon autonome, les premiers essais semblent indiquer que l'approche utilisée est prometteuse. En effet, la méthode propose de se servir des redondances du modèle pour se localiser de façon précise. De ce fait, le processus est robuste face aux différents aléas rencontrés. Toutefois, certaines pistes restent à parcourir pour rendre la méthode plus performante, telle que l'utilisation d'un filtre de Kalman lors de la phase de prédiction, afin de répondre parfaitement à l'ensemble des exigences requises.

La phase de mesure est, quant à elle, en bonne voie. Effectivement, bien que le choix du capteur ne soit pas définitif, les premières expériences de déplacement du bras à l'aide des algorithmes de perception ont débutées. Celles-ci restent cependant à approfondir afin de définir la stratégie de déplacement optimale.

D'un point de vue plus personnel, ce stage m'a permis d'acquérir des compétences complémentaires liées à la robotique, telle que ROS, mais aussi de découvrir le monde captivant de la recherche, me confortant dans l'idée de poursuivre mes études par une thèse.

# Bibliographie

- [1] Liang Xiao, Bin Dai, Daxue Liu, Dawei Zhao, and Tao Wu. Monocular Road Detection Using Structured Random Forest. International Journal of Advanced Robotic Systems, 13(3) :101, June 2016.
- [2] Weili Ding, Y Li, W Wang, and Y Zou. Vanishing point detection algorithm for urban road image based on the envelope of perpendicular and parallel lines. 34, 10 2014.
- [3] H. Kong, S. E. Sarma, and F. Tang. Generalizing Laplacian of Gaussian Filters for Vanishing-Point Detection. IEEE Transactions on Intelligent Transportation Systems, 14(1) :408–418, March 2013.
- [4] Romuald AUFRERE and Jean GALLICE. Reconnaissance et suivi de route par vision artificielle, application à l'aide à la conduite. PhD thesis, Université de Clermont-Ferrand 2, Clermont-Ferrand, 2001.
- [5] J. A. Marchant. Tracking of row structure in three crops using image analysis. Computers and Electronics in Agriculture, 15(2) :161–179, July 1996.
- [6] F Rovira-Más, Q Zhang, J F Reid, and J D Will. Hough-transform-based vision algorithm for crop row detection of an automated agricultural vehicle. Proceedings of the Institution of Mechanical Engineers, Part D : Journal of Automobile Engineering, 219(8) :999–1010, August 2005.
- [7] Tijmen Bakker, Hendrik Wouters, Kees van Asselt, Jan Bontsema, Lie Tang, Joachim Müller, and Gerrit van Straten. A vision based row detection system for sugar beet. Computers and Electronics in Agriculture, 60(1) :87–95, January 2008.
- [8] Ch. Gée, J. Bossu, G. Jones, and F. Truchetet. Crop/weed discrimination in perspective agronomic images. Computers and Electronics in Agriculture, 60(1) :49–59, January 2008.
- [9] Juan Luis Barreda Sánchez. Seguimiento visual de líneas de cultivo. September 2009.
- [10] Guoquan Jiang, Xiaojie Wang, Zhiheng Wang, and Hongmin Liu. Wheat rows detection at the early growth stage based on Hough transform and vanishing point. Computers and Electronics in Agriculture, 123 :211–223, April 2016.
- [11] Ronghua Ji and Lijun Qi. Crop-row detection algorithm based on Random Hough Transformation. Mathematical and Computer Modelling, 54(3) :1016–1020, August 2011.
- [12] M. Montalvo, G. Pajares, J. M. Guerrero, J. Romeo, M. Guijarro, A. Ribeiro, J. J. Ruz, and J. M. Cruz. Automatic detection of crop rows in maize fields with high weeds pressure. Expert Systems with Applications, 39(15) :11889–11897, November 2012.
- [13] Iván D. García-Santillán, Martín Montalvo, José M. Guerrero, and Gonzalo Pajares. Automatic detection of curved and straight crop rows from images in maize fields. Biosystems Engineering, 156 :61–79, April 2017.

- [14] Vladimir Naumovich Vapnik and Vlamimir Vapnik. Statistical learning theory, volume 1. Wiley New York, 1998.

# Annexe A

## Note de calcul pour la modélisation

À l'aide du modèle sténopée, on souhaiterait obtenir dans l'image une droite d'équation  $u = Av + B$  correspondant à chacune des lignes. Pour cela, il faut étudier la variation de  $\frac{U}{V}$  et  $\frac{W}{V}$  (ou de  $u$  et  $v$ ) en fonction de la variable  $Y$ .

Ceci revient à exprimer  $\frac{U}{V}$  en fonction de  $\frac{W}{V}$  en utilisant la variable  $Y$  comme pivot. Pour la lisibilité des calculs on notera :

$$\begin{aligned} M &= P_{11}X + P_{13}Z + P_{14}, \\ N &= P_{21}X + P_{23}Z + P_{24}, \\ O &= P_{31}X + P_{33}Z + P_{34}, \\ C &= \frac{MP_{22} - NP_{12}}{OP_{22} - NP_{32}}, \\ D &= \frac{P_{12}}{P_{22}} - \left( \frac{MP_{22} - NP_{12}}{OP_{22} - NP_{32}} \right) \frac{P_{32}}{P_{22}}, \end{aligned}$$

Il vient :

$$\begin{aligned} \left\{ \begin{array}{l} \frac{U}{V} = \frac{P_{12}Y + M}{P_{22}Y + N} \\ \frac{W}{V} = \frac{P_{32}Y + O}{P_{22}Y + N} \end{array} \right. & \xrightarrow{\substack{\text{si } P_{22} \neq 0: P_{12}Y = (P_{22}Y + N - N) \frac{P_{12}}{P_{22}} \\ \text{si } P_{22} \neq 0: P_{32}Y = (P_{22}Y + N - N) \frac{P_{32}}{P_{22}}}} \left\{ \begin{array}{l} \frac{U}{V} = \frac{P_{12}}{P_{22}} + \frac{M - N \frac{P_{12}}{P_{22}}}{P_{22}Y + N} \\ \frac{W}{V} = \frac{P_{32}}{P_{22}} + \frac{O - N \frac{P_{32}}{P_{22}}}{P_{22}Y + N} \end{array} \right. \\ & \xrightarrow{\text{si } \frac{W}{V} - \frac{P_{32}}{P_{22}} \neq 0} \left\{ \begin{array}{l} \frac{U}{V} = \frac{P_{12}}{P_{22}} + \frac{M - N \frac{P_{12}}{P_{22}}}{O - N \frac{P_{32}}{P_{22}}} \left( \frac{W}{V} - \frac{P_{32}}{P_{22}} \right) \\ P_{22}Y + N = \frac{O - N \frac{P_{32}}{P_{22}}}{\frac{W}{V} - \frac{P_{32}}{P_{22}}} \end{array} \right. \\ & \xrightarrow[\text{En notant } \frac{U}{V} = C \frac{W}{V} + D]{\text{III.1}} \left\{ \begin{array}{l} u = k_x \left( C \frac{v - c_y}{k_y} + D \right) + c_x \\ P_{22}Y + N = \frac{O - N \frac{P_{32}}{P_{22}}}{\frac{W}{V} - \frac{P_{32}}{P_{22}}} \end{array} \right. \end{aligned}$$

On a ainsi une équation de droite dans l'espace image (sous certaines hypothèses) de la forme  $u = Av + B$  avec :

$$\boxed{\begin{aligned} A &= \frac{k_x C}{k_y}, \\ B &= c_x + k_x \left( D - C \frac{c_y}{k_y} \right). \end{aligned}} \quad (\text{A.1})$$

Il s'agira donc par la suite de retrouver les droites pour  $X \in \{-L; 0; L\}$ .

### Validité des hypothèses

Dans les équations, on a besoin que  $P_{22}$  soit non-nul. Or ce paramètre caractérise la relation entre la direction principale du champ (notée  $Y$ , l'axe de profondeur) et l'axe optique de la caméra.

$$P_{22} = 0 \Leftrightarrow c_\alpha(c_\beta c_\gamma - c_\delta s_\beta s_\gamma) - s_\alpha s_\delta s_\gamma = 0 \Leftrightarrow \vec{e}_W \perp \vec{e}_Y.$$

Or dans les conditions réelles, l'axe de la caméra n'est jamais perpendiculaire à l'axe du champ donc cette hypothèse est valide.

Pour que le modèle soit valide, on a aussi besoin que  $\frac{W}{V} - \frac{P_{32}}{P_{22}}$  soit non-nul. Il vient :

$$\begin{aligned} & \left\{ \frac{W}{V} - \frac{P_{32}}{P_{22}} \neq 0 \right. \\ \frac{V=P_{22}Y+N; \quad P_{22} \neq 0}{W=P_{32}Y+O} & \rightarrow \left\{ P_{32}(P_{22}Y + N) \neq P_{22}(P_{32}Y + O) \right. \\ & \rightarrow \left\{ P_{32}N \neq P_{22}O \right. \\ \frac{P_{22} \neq 0}{& \rightarrow \left\{ X(P_{31}P_{22} - P_{21}P_{32}) \neq Z(P_{23}P_{32} - P_{33}P_{22}) + P_{24}P_{32} - P_{34}P_{22} \right.} \\ & \rightarrow \left\{ Xm \neq Zn + o \right. \end{aligned}$$

L'hypothèse est donc justifiée si l'on ne travaille pas dans le plan  $Vect(\frac{n+o}{m}, 0, 1), (0, 1, 0)$  si  $m \neq 0$ .

## Annexe B

### Calcul de la matrice Jacobienne $J_C$

La matrice Jacobienne permet dans notre cas de lier les variations dans l'espace image avec celles dans l'espace réel. En effet, l'approximation d'une fonction à plusieurs variables est donnée par :

$$F(X) \approx F(X_0) + J_{F_{X_0}} (X - X_0)$$

Dans notre cas, on a :

$$\begin{cases} X_d \approx X_{d_0} + J_{C_{X_{l_0}}} (X_l - X_{l_0}) \\ C_{X_d} = J_C C_{X_l} J_C^t \end{cases}$$

Il nous faut donc calculer la matrice Jacobienne  $J_C$

$$J_C = \begin{pmatrix} \frac{\partial u_{1g}}{\partial X_0} & \frac{\partial u_{1g}}{\partial \alpha} & \frac{\partial u_{1g}}{\partial \beta} & \frac{\partial u_{1g}}{\partial \delta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_{ng}}{\partial X_0} & \frac{\partial u_{ng}}{\partial \alpha} & \frac{\partial u_{ng}}{\partial \beta} & \frac{\partial u_{ng}}{\partial \delta} \\ \frac{\partial u_{1c}}{\partial X_0} & \frac{\partial u_{1c}}{\partial \alpha} & \frac{\partial u_{1c}}{\partial \beta} & \frac{\partial u_{1c}}{\partial \delta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_{nc}}{\partial X_0} & \frac{\partial u_{nc}}{\partial \alpha} & \frac{\partial u_{nc}}{\partial \beta} & \frac{\partial u_{nc}}{\partial \delta} \\ \frac{\partial u_{1d}}{\partial X_0} & \frac{\partial u_{1d}}{\partial \alpha} & \frac{\partial u_{1d}}{\partial \beta} & \frac{\partial u_{1d}}{\partial \delta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_{nd}}{\partial X_0} & \frac{\partial u_{nd}}{\partial \alpha} & \frac{\partial u_{nd}}{\partial \beta} & \frac{\partial u_{nd}}{\partial \delta} \end{pmatrix} \quad (\text{B.1})$$

Pour faciliter les calculs de la matrice Jacobienne (B.1), on peut remarquer que :

$$\frac{\partial u_{i,\lambda}}{\partial q_j} = \frac{\partial A_\lambda}{\partial q_j} v_i + A_\lambda \frac{\partial v_i}{\partial q_j} + \frac{\partial B_\lambda}{\partial q_j}. \quad (\text{B.2})$$

Or

$$\begin{cases} A = \frac{k_x C}{k_y}, \\ B = c_x + k_x (D - C \frac{c_y}{k_y}), \\ C = \frac{MP_{22} - NP_{12}}{OP_{22} - NP_{32}} = \frac{C_{Num}}{C_{Denom}}, \\ D = \frac{P_{12}}{P_{22}} - \left( \frac{MP_{22} - NP_{12}}{OP_{22} - NP_{32}} \right) \frac{P_{32}}{P_{22}} = \frac{1}{P_{22}} [P_{12} - CP_{32}], \\ M = P_{11}X + P_{13}Z + P_{14}, \\ N = P_{21}X + P_{23}Z + P_{24}, \\ O = P_{31}X + P_{33}Z + P_{34}, \end{cases}$$

$$\frac{\lambda \in \{-1; 0; 1\}}{q_j \in \{X_0; \alpha; \beta; \delta\}} \rightarrow \begin{cases} \frac{\partial A_\lambda}{\partial q_j} = \frac{k_x}{k_y} \frac{\partial C}{\partial q_j} \\ \frac{\partial B_\lambda}{\partial q_j} = k_x \frac{\partial D}{\partial q_j} - c_y \frac{\partial A_\lambda}{\partial q_j} \\ \frac{\partial C}{\partial q_j} = \frac{1}{(OP_{22} - NP_{32})^2} \left[ \frac{\partial C_{Num}}{\partial q_j} (OP_{22} - NP_{32}) - \frac{\partial C_{Denom}}{\partial q_j} (MP_{22} - NP_{12}) \right] \\ \frac{\partial D}{\partial q_j} = \frac{1}{P_{22}^2} \left[ \left( \frac{\partial P_{12}}{\partial q_j} - \frac{\partial P_{32}}{\partial q_j} C - \frac{\partial C}{\partial q_j} P_{32} \right) P_{22} - \frac{\partial P_{22}}{\partial q_j} (P_{12} - P_{32} C) \right] \end{cases} \quad (\text{B.3})$$

Au vu des équations, il nous faut désormais calculer les différents coefficients  $P_{i,j}$  ainsi que leurs dérivées partielles.

## 1 Définition de $P_{i,j}$

D'après la modélisation on sait que  $P = R_\gamma^{x'''} T_{Z_0}^{y'''} T_{Y_0}^{z'''} R_\delta^{y''} R_\alpha^{z''} R_\beta^{x''} R_\alpha^z T_{X_0}^x$  avec  $R_\theta^i$  la matrice de rotation d'angle  $\theta$  selon l'axe  $i$  et  $T_i^j$  la matrice de translation de longueur  $l$  suivant l'axe  $i$ .

Pour faciliter les futurs calculs, les variables du modèle sont représentées en rouge.

$$\begin{aligned}
P &= \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} c_\delta & 0 & s_\delta & 0 \\ 0 & 1 & 0 & Y_0 \\ -s_\delta & 0 & c_\delta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\beta & -s_\beta & 0 \\ 0 & s_\beta & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\alpha & -s_\alpha & 0 & X_0 c_\alpha \\ s_\alpha & c_\alpha & 0 & X_0 s_\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} c_\delta & 0 & s_\delta & 0 \\ s_\delta s_\gamma & c_\gamma & -c_\delta s_\gamma & Y_0 c_\gamma - Z_0 s_\gamma \\ -s_\delta c_\gamma & s_\gamma & c_\delta c_\gamma & Y_0 s_\gamma + Z_0 c_\gamma \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\alpha & -s_\alpha & 0 & X_0 c_\alpha \\ s_\alpha c_\beta & c_\alpha c_\beta & -s_\beta & X_0 s_\alpha c_\beta \\ s_\alpha s_\beta & c_\alpha s_\beta & c_\beta & X_0 s_\alpha s_\beta \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} c_\alpha c_\delta + s_\alpha s_\beta s_\delta & & & \\ s_\alpha (c_\beta c_\gamma - c_\delta s_\beta s_\gamma) + c_\alpha s_\delta s_\gamma & c_\alpha (c_\beta c_\gamma - c_\delta s_\beta s_\gamma) - s_\alpha s_\delta s_\gamma & -c_\gamma s_\beta - c_\beta c_\delta s_\gamma & X_0 (c_\alpha c_\delta + s_\alpha s_\beta s_\delta) \\ s_\alpha (c_\beta s_\gamma + c_\delta c_\gamma s_\beta) - c_\alpha c_\gamma s_\delta & c_\alpha (c_\beta s_\gamma + c_\delta c_\gamma s_\beta) + c_\gamma s_\alpha s_\delta & c_\beta c_\delta c_\gamma - s_\beta s_\gamma & X_0 (s_\alpha (c_\beta c_\gamma - c_\delta s_\beta s_\gamma) + c_\alpha s_\delta s_\gamma) + Y_0 c_\gamma - Z_0 s_\gamma \\ 0 & 0 & 0 & X_0 (s_\alpha (c_\beta s_\gamma + c_\delta c_\gamma s_\beta) - c_\alpha c_\gamma s_\delta) + Z_0 c_\gamma + Y_0 s_\gamma \end{pmatrix}
\end{aligned}$$

## 2 Calcul de $\frac{\partial P_{i,j}}{\partial q_i}$

$$\left\{ \begin{array}{l}
 \frac{\partial P_{11}}{\partial X_0} = 0 \\
 \frac{\partial P_{12}}{\partial X_0} = 0 \\
 \frac{\partial P_{13}}{\partial X_0} = 0 \\
 \frac{\partial P_{14}}{\partial X_0} = c_\alpha c_\delta + s_\alpha s_\beta s_\delta \\
 \frac{\partial P_{21}}{\partial X_0} = 0 \\
 \frac{\partial P_{22}}{\partial X_0} = 0 \\
 \frac{\partial P_{23}}{\partial X_0} = 0 \\
 \frac{\partial P_{24}}{\partial X_0} = s_\alpha (c_\beta c_\gamma - c_\delta s_\beta s_\gamma) + c_\alpha s_\delta s_\gamma \\
 \frac{\partial P_{31}}{\partial X_0} = 0 \\
 \frac{\partial P_{32}}{\partial X_0} = 0 \\
 \frac{\partial P_{33}}{\partial X_0} = 0 \\
 \frac{\partial P_{34}}{\partial X_0} = s_\alpha (c_\beta s_\gamma + c_\delta c_\gamma s_\beta) - c_\alpha c_\gamma s_\delta
 \end{array} \right\}
 \left\{ \begin{array}{l}
 \frac{\partial P_{11}}{\partial \alpha} = c_\alpha s_\beta s_\delta - c_\delta s_\alpha \\
 \frac{\partial P_{12}}{\partial \alpha} = -c_\alpha c_\delta - s_\alpha s_\beta s_\delta \\
 \frac{\partial P_{13}}{\partial \alpha} = 0 \\
 \frac{\partial P_{14}}{\partial \alpha} = X_0 c_\alpha s_\beta s_\delta - X_0 c_\delta s_\alpha \\
 \frac{\partial P_{21}}{\partial \alpha} = c_\alpha (c_\beta c_\gamma - c_\delta s_\beta s_\gamma) - s_\alpha s_\delta s_\gamma \\
 \frac{\partial P_{22}}{\partial \alpha} = -s_\alpha (c_\beta c_\gamma - c_\delta s_\beta s_\gamma) - c_\alpha s_\delta s_\gamma \\
 \frac{\partial P_{23}}{\partial \alpha} = 0 \\
 \frac{\partial P_{24}}{\partial \alpha} = X_0 (c_\alpha (c_\beta c_\gamma - c_\delta s_\beta s_\gamma) - s_\alpha s_\delta s_\gamma) \\
 \frac{\partial P_{31}}{\partial \alpha} = c_\alpha (c_\beta s_\gamma + c_\delta c_\gamma s_\beta) + c_\gamma s_\alpha s_\delta \\
 \frac{\partial P_{32}}{\partial \alpha} = c_\alpha c_\gamma s_\delta - s_\alpha (c_\beta s_\gamma + c_\delta c_\gamma s_\beta) \\
 \frac{\partial P_{33}}{\partial \alpha} = 0 \\
 \frac{\partial P_{34}}{\partial \alpha} = X_0 (c_\alpha (c_\beta s_\gamma + c_\delta c_\gamma s_\beta) + c_\gamma s_\alpha s_\delta)
 \end{array} \right\}
 \left\{ \begin{array}{l}
 \frac{\partial P_{11}}{\partial \beta} = c_\beta s_\alpha s_\delta \\
 \frac{\partial P_{12}}{\partial \beta} = c_\alpha c_\beta s_\delta \\
 \frac{\partial P_{13}}{\partial \beta} = -s_\beta s_\delta \\
 \frac{\partial P_{14}}{\partial \beta} = X_0 c_\beta s_\alpha s_\delta \\
 \frac{\partial P_{21}}{\partial \beta} = -s_\alpha (c_\gamma s_\beta + c_\beta c_\delta s_\gamma) \\
 \frac{\partial P_{22}}{\partial \beta} = -c_\alpha (c_\gamma s_\beta + c_\beta c_\delta s_\gamma) \\
 \frac{\partial P_{23}}{\partial \beta} = c_\delta s_\beta s_\gamma - c_\beta c_\gamma \\
 \frac{\partial P_{24}}{\partial \beta} = -X_0 s_\alpha (c_\gamma s_\beta + c_\beta c_\delta s_\gamma) \\
 \frac{\partial P_{31}}{\partial \beta} = -s_\alpha (s_\beta s_\gamma - c_\beta c_\delta c_\gamma) \\
 \frac{\partial P_{32}}{\partial \beta} = -c_\alpha (s_\beta s_\gamma - c_\beta c_\delta c_\gamma) \\
 \frac{\partial P_{33}}{\partial \beta} = -c_\beta s_\gamma - c_\delta c_\gamma s_\beta \\
 \frac{\partial P_{34}}{\partial \beta} = -X_0 s_\alpha (s_\beta s_\gamma - c_\beta c_\delta c_\gamma)
 \end{array} \right\}
 \left\{ \begin{array}{l}
 \frac{\partial P_{11}}{\partial \delta} = c_\delta s_\alpha s_\beta - c_\alpha s_\delta \\
 \frac{\partial P_{12}}{\partial \delta} = s_\alpha s_\delta + c_\alpha c_\delta s_\beta \\
 \frac{\partial P_{13}}{\partial \delta} = c_\beta c_\delta \\
 \frac{\partial P_{14}}{\partial \delta} = X_0 c_\delta s_\alpha s_\beta - X_0 c_\alpha s_\delta \\
 \frac{\partial P_{21}}{\partial \delta} = s_\gamma (c_\alpha c_\delta + s_\alpha s_\beta s_\delta) \\
 \frac{\partial P_{22}}{\partial \delta} = -s_\gamma (c_\delta s_\alpha - c_\alpha s_\beta s_\delta) \\
 \frac{\partial P_{23}}{\partial \delta} = c_\beta s_\delta s_\gamma \\
 \frac{\partial P_{24}}{\partial \delta} = X_0 s_\gamma (c_\alpha c_\delta + s_\alpha s_\beta s_\delta) \\
 \frac{\partial P_{31}}{\partial \delta} = -c_\gamma (c_\alpha c_\delta + s_\alpha s_\beta s_\delta) \\
 \frac{\partial P_{32}}{\partial \delta} = c_\gamma (c_\delta s_\alpha - c_\alpha s_\beta s_\delta) \\
 \frac{\partial P_{33}}{\partial \delta} = -c_\beta c_\gamma s_\delta \\
 \frac{\partial P_{34}}{\partial \delta} = -X_0 c_\gamma (c_\alpha c_\delta + s_\alpha s_\beta s_\delta)
 \end{array} \right\}$$

### 3 Définition de $\frac{\partial C}{\partial q_i}$

$$\frac{\partial C}{\partial q_j} = \frac{1}{(OP_{22} - NP_{32})^2} \left[ \frac{\partial C_{Num}}{\partial q_j} (OP_{22} - NP_{32}) - \frac{\partial C_{Denom}}{\partial q_j} (MP_{22} - NP_{12}) \right]$$

$$\left\{ \begin{array}{l} \frac{\partial C_{Num}}{\partial q_i} = \frac{\partial M}{\partial q_i} P_{22} + \frac{\partial P_{22}}{\partial q_i} M - \frac{\partial N}{\partial q_i} P_{12} - \frac{\partial P_{12}}{\partial q_i} N \\ \frac{\partial C_{Denom}}{\partial q_i} = \frac{\partial O}{\partial q_i} P_{22} + \frac{\partial P_{22}}{\partial q_i} O - \frac{\partial N}{\partial q_i} P_{32} - \frac{\partial P_{32}}{\partial q_i} N \\ \frac{\partial M}{\partial q_i} = \frac{\partial P_{11}}{\partial q_i} X + \frac{\partial P_{13}}{\partial q_i} Z + \frac{\partial P_{14}}{\partial q_i} \\ \frac{\partial N}{\partial q_i} = \frac{\partial P_{21}}{\partial q_i} X + \frac{\partial P_{23}}{\partial q_i} Z + \frac{\partial P_{24}}{\partial q_i} \\ \frac{\partial O}{\partial q_i} = \frac{\partial P_{31}}{\partial q_i} X + \frac{\partial P_{33}}{\partial q_i} Z + \frac{\partial P_{34}}{\partial q_i} \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{\partial M}{\partial X_0} = c_\alpha c_\delta + s_\alpha s_\beta s_\delta \\ \frac{\partial M}{\partial \alpha} = (c_\alpha s_\beta s_\delta - c_\delta s_\alpha) X + X_0 c_\alpha s_\beta s_\delta - X_0 c_\delta s_\alpha \\ \frac{\partial M}{\partial \beta} = (c_\beta s_\alpha s_\delta) X - s_\beta s_\delta Z + X_0 c_\beta s_\alpha s_\delta \\ \frac{\partial M}{\partial \delta} = (c_\delta s_\alpha s_\beta - c_\alpha s_\delta) X + c_\beta c_\delta Z + X_0 c_\delta s_\alpha s_\beta - X_0 c_\alpha s_\delta \\ \frac{\partial N}{\partial X_0} = 0 \\ \frac{\partial N}{\partial \alpha} = 0 \\ \frac{\partial N}{\partial \beta} = 0 \\ \frac{\partial N}{\partial \delta} = 0 \\ \frac{\partial O}{\partial X_0} = 0 \\ \frac{\partial O}{\partial \alpha} = 0 \\ \frac{\partial O}{\partial \beta} = 0 \\ \frac{\partial O}{\partial \delta} = 0 \end{array} \right. \rightarrow \left\{ \begin{array}{l} \frac{\partial C_{Num}}{\partial X_0} = c_\alpha c_\delta + s_\alpha s_\beta s_\delta \\ \frac{\partial C_{Num}}{\partial \alpha} = 0 \\ \frac{\partial C_{Num}}{\partial \beta} = 0 \\ \frac{\partial C_{Num}}{\partial \delta} = 0 \\ \frac{\partial C_{Denom}}{\partial X_0} = 0 \\ \frac{\partial C_{Denom}}{\partial \alpha} = 0 \\ \frac{\partial C_{Denom}}{\partial \beta} = 0 \\ \frac{\partial C_{Denom}}{\partial \delta} = 0 \end{array} \right.$$

# Annexe C

## Principe de discrimination par SVM

La discrimination par séparateurs à vaste marge [14] (ou machines à vecteurs de support, souvent noté SVM) est une méthode de classement statistique basée sur un apprentissage supervisé. Le principe est de regrouper des échantillons ayant des propriétés similaires afin de créer des classes séparées de façon linéaire. Dans le cas où le problème de discrimination est non linéairement séparable, la méthode propose une idée pour tenter de se ramener à un problème linéaire.

### 1 Principe général

L'idée intuitive de l'approche est de trouver la frontière séparant "au mieux"<sup>1 2</sup> deux ensembles d'échantillons dont les propriétés sont distinctes. On définit ainsi un hyperplan<sup>3</sup> comme solution d'un problème d'optimisation sous contraintes dont la fonction d'objectif est exprimée uniquement à partir de produits scalaires.

Pour construire un estimateur de la fonction de décision idéal, on possède au départ un ensemble de données d'entraînement  $\{(\vec{x}_1, y_1); (\vec{x}_2, y_2); \dots; (\vec{x}_n, y_n)\}$  où  $\vec{x}_i \in \mathbb{R}$  est l'échantillon  $i$  et  $y_i \in \{-1; +1\}$  indique la classe à laquelle appartient ce point. Ce problème revient en fait à trouver une fonction  $h$  qui, à un vecteur d'entrée  $\vec{x}$ , associe  $y = h(\vec{x})$ .

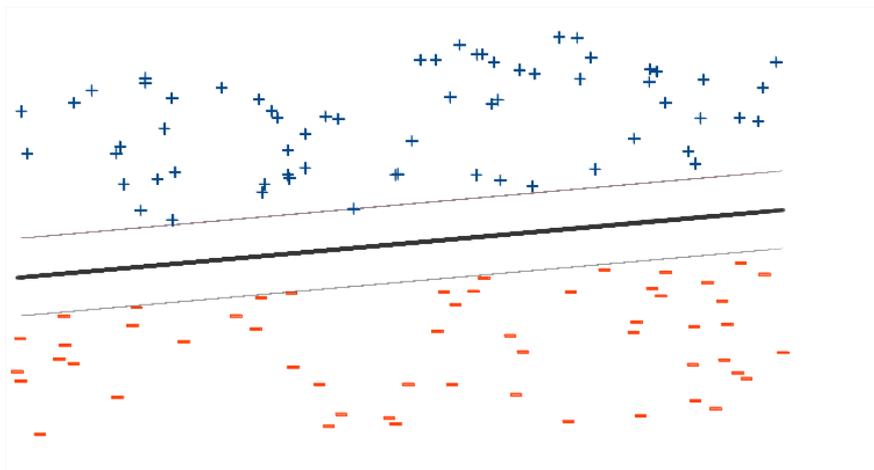


FIGURE C.1 – Exemple de discrimination. Il s'agit de séparer les croix des traits. La frontière est représentée en noir.

1. D'après la théorie statistique de l'apprentissage, la frontière est optimale lorsque la distance entre la frontière et les échantillons les plus proches est maximale. On parle alors de marge maximale.
2. Dans un espace  $E$  complet muni d'un produit scalaire. On dit aussi que l'espace est Hilbertien.
3. Dans un espace de dimension finie  $n$ , un hyperplan est un espace de dimension  $n-1$ . Dans un espace à trois dimensions, un hyperplan sera un plan (dimension 2))

## 2 Cas d'une fonction discriminante linéaire

### 2.1 Formalisation du problème

Un problème de discrimination est dit linéairement séparable lorsqu'il existe une fonction de décision linéaire  $h$  telle que  $h(\vec{x}) = \vec{n}^T \cdot \vec{x} + a$  classe correctement tous les échantillons de l'apprentissage; autrement dit :

$\text{Signe}(h(\vec{x}_i)) = y_i, i \in [1, n]$  ou encore  $y_i h(\vec{x}_i) \geq 0, i \in [1, n]$ .

La frontière de décision est alors obtenue pour l'hyperplans d'équation  $h(\vec{x}) = 0$  noté  $\mathbb{P}$ . Dans ce cas, il existe souvent une infinité de frontières linéaires séparant l'ensemble des échantillons. Pour déterminer la frontière optimale, on cherche à maximiser la confiance dans notre estimateur et ainsi minimiser la probabilité d'erreur associée à notre classifieur. Le problème revient donc à maximiser la marge :

$$\boxed{\begin{cases} \max_{\vec{n}, a} & \min_{i \in [1, n]} & \text{dist}(\vec{x}_i, \mathbb{P}) \end{cases}} \quad (\text{C.1})$$

En posant  $m = \min_{i \in [1, n]} \text{dist}(\vec{x}_i, \mathbb{P}) = \min_{i \in [1, n]} \frac{|\vec{n}^T \cdot \vec{x}_i + a|}{\|\vec{n}\|}$  comme variable, on peut réécrire l'équation précédente comme un problème d'optimisation sous contraintes.

$$\begin{cases} \max_{\vec{n}, a} m \\ \text{avec} & \frac{y_i(\vec{n}^T \cdot \vec{x}_i + a)}{\|\vec{n}\|} \geq m; \quad i \in [1, n] \end{cases}$$

En posant  $\vec{w} = \frac{\vec{n}}{m\|\vec{n}\|}$  et  $b = \frac{a}{m\|\vec{n}\|}$  le problème peut devenir :

$$\boxed{\begin{cases} \min_{\vec{w}, b} \frac{1}{2} \|\vec{w}\|^2 \\ \text{avec} & y_i(\vec{w}^T \cdot \vec{x}_i + b) \geq 1; \quad i \in [1, n] \end{cases}} \quad (\text{C.2})$$

### 2.2 Résolution du problème

Pour résoudre ce problème d'optimisation sous contraintes, une des approches souvent utilisée est d'utiliser la méthode de Kuhn-Tucker.

Rappel : Soit le problème d'optimisation sous contraintes suivant :

$$\begin{cases} \min_x f(x) \\ \text{avec} & h_i(x) \geq 0; \quad i \in [1, m] \end{cases}$$

Avec les fonction  $h_i$  linéaires. On définit le Lagrangien comme la fonction  $\mathbb{L}$  suivante :

$$\mathbb{L}(x, \lambda) = \sum_{i=0}^m \lambda_i h_i - f(x)$$

Dans ce cas, le théorème de Khun-Tucker indique que si  $x^*$  est solution alors il existe un unique vecteur  $\lambda^*$  tel que

$$\begin{cases} \frac{\partial \mathbb{L}(x^*, \lambda^*)}{\partial x_i} = 0 & i \in [1, n] \\ \frac{\partial \mathbb{L}(x^*, \lambda^*)}{\partial \lambda_j} \geq 0 & j \in [i, m] \\ \lambda_j^* \geq 0 & j \in [i, m] \\ \lambda_j^* h_j = 0 & j \in [i, m] \end{cases}$$

En appliquant ce théorème, on obtient

$$\begin{cases} h(\vec{x}) = \left( \sum_{i=0}^n \lambda_i^* y_i \vec{x}_i \right)^T \cdot \vec{x} + b \\ \lambda_i^* = 0 & \text{si } x_i \text{ n'est pas un vecteur de support} \end{cases} \quad (\text{C.3})$$

<sup>4</sup> Une des forces majeures de cette méthode est sa robustesse face à la modification des données d'entraînement. Ainsi comme le modèle ne dépend que des vecteurs de supports, l'ajout d'un point éloigné d'une distance  $\geq m$  de la frontière ne modifiera pas le modèle.

### 3 Cas d'une fonction discriminante non-linéaire

Dans le cas où il n'existe pas de fonction de décision linéaire, une méthode envisagée est de travailler dans un espace de redescription (transformation de l'espace de départ par une fonction  $\psi$ , non linéaire) pour se ramener à un problème linéaire (dans le nouvel espace). La plupart du temps, il s'agit d'augmenter la taille de l'espace de départ dans le but de faire apparaître un séparateur linéaire.

Pour ce faire, on ne considérera plus  $\vec{x}$  mais  $\phi(\vec{x})$  mais aussi une fonction appelée noyau qui traduit l'influence réciproque vérifiant :  $k(\vec{x}_1, \vec{x}_2) = \phi(\vec{x}_1)^T \cdot \phi(\vec{x}_2)$ . Par conséquent, on aura :

$$\begin{aligned} \vec{n} &= \sum_{i=0}^n \lambda_i y_i \phi(\vec{x}_i) \\ \vec{n}^T \cdot \phi(\vec{x}) &= \sum_{i=0}^n \lambda_i y_i k(\vec{x}_i, \vec{x}) \\ h(\vec{x}) &= \vec{n}^T \cdot \phi(\vec{x}) + b \end{aligned}$$

Le calcul de l'hyperplan se fait dans l'espace initial ce qui évite un produit scalaire dans un espace de grande dimension (opération pouvant être coûteuse). De plus, il n'y a pas besoin de connaître explicitement la fonction  $\phi$  mais uniquement la fonction noyau  $k$  que l'on construit directement.

---

4. Un vecteur de support est un vecteur qui vérifie  $m = \text{dist}(\vec{x}_i, \mathbb{P})$

# Annexe D

## Introduction au filtre de Kalman

Le filtre de Kalman est une méthode visant à estimer des paramètres d'un système évoluant dans le temps à partir de mesures bruitées. La force de ce filtre est sa capacité de prédiction des paramètres et de rectification des erreurs, non seulement des capteurs, mais aussi du modèle lui-même! En effet, pour appliquer un filtre de Kalman, il faut avant tout modéliser le système pour lequel on veut estimer les paramètres, de manière linéaire (des variantes du filtre de Kalman existent pour la prise en compte de modèles non linéaires) La force du filtre de Kalman est de contenir un terme d'imprécision sur le modèle lui-même, ce qui lui permet de donner des estimations correctes malgré les erreurs de modélisation.

Le fonctionnement du filtre de Kalman peut se diviser en deux étapes

- La première est la prédiction de l'estimation selon le modèle du système. Pour ce faire, le filtre de Kalman reprend l'estimation précédente des paramètres et de l'erreur et prédit les nouveaux paramètres et la nouvelle erreur en fonction de la modélisation du système.
- La seconde va faire la mise à jour de cette prédiction grâce aux nouvelles mesures. Ces mesures (par définition bruitées) vont permettre d'obtenir une estimation des paramètres et de l'erreur à partir de la prédiction faite. Si jamais le modèle comporte des erreurs, cette étape de mise à jour permettra de les rectifier.

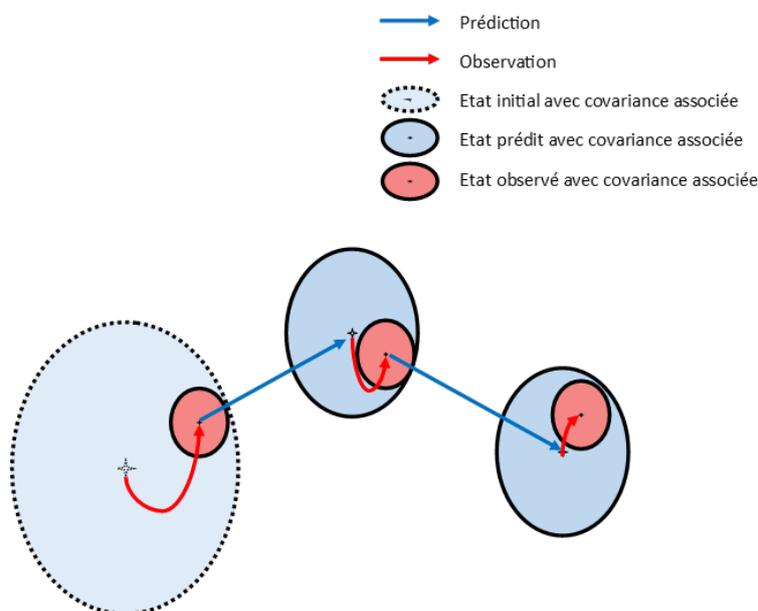


FIGURE D.1 – Principe du filtre de Kalman

# Annexe E

## Méthode de Denavit-Hartenberg

Dans la convention de Denavit-Hartenberg, on commence par fixer les axes des liaisons comme suit :

- $\vec{z}_i$  suivant l'axe de la liaison.
- $\vec{x}_i$  suivant la normale commune entre  $\vec{z}_{i-1}$  et  $\vec{z}_i$  on a donc  $\vec{x}_i = \vec{z}_{i-1} \wedge \vec{z}_i$  (si  $\vec{z}_{i-1}$  et  $\vec{z}_i$  non colinéaires).
- $\vec{y}_i$  tel que le repère  $(A_i, \vec{x}_i, \vec{y}_i, \vec{z}_i)$  soit direct donc  $\vec{y}_i = \vec{z}_i \wedge \vec{x}_i$ .

On note ensuite :

- $\theta_i$  l'angle autour de  $\vec{z}_{i-1}$  permettant de passer de  $\vec{x}_{i-1}$  à  $\vec{x}_i$ .
- $r_i$  la distance selon l'axe  $\vec{x}_i$  entre les axes  $\vec{z}_{i-1}$  et  $\vec{z}_i$ .
- $d_i$  la distance selon l'axe  $\vec{z}_{i-1}$  entre les axes  $\vec{x}_{i-1}$  et  $\vec{x}_i$ .
- $\alpha_i$  l'angle autour de  $\vec{x}_i$  permettant de passer de  $\vec{z}_{i-1}$  à  $\vec{z}_i$ .

NB : Dans le cas d'une liaison glissière la variable articulaire est  $d_i$  tandis que dans le cas d'une liaison pivot, c'est  $\theta_i$  qui joue ce rôle.

# Annexe F

## Note de calcul du bras UR5

### 1 Matrices de passage

D'après la convention de Denavit-Hartenberg, on a :

$$\boxed{T_{i-1;i} = Rot_{\theta_i}^z \quad Trans_{d_i}^z \quad Trans_{r_i}^x \quad Rot_{\alpha_i}^x} \quad (\text{F.1})$$

où  $T_{ij}$  désigne la matrice de passage exprimant la position et l'orientation d'un repère  $R_j$  par rapport au repère  $R_i$ . NB : Par la suite, on remplacera  $\cos(\theta_i)$  par  $c_i$ ,  $\sin(\theta_i)$  par  $s_i$ ,  $\tan(\theta_i)$  par  $t_i$  mais également  $\cos(\theta_i + \theta_j)$  par  $c_{ij}$

Grâce à la formule précédente, on obtient :

$$\begin{array}{l} T_{01} = \begin{pmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{23} = \begin{pmatrix} c_3 & -s_3 & 0 & r_3 c_3 \\ s_3 & c_3 & 0 & r_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{45} = \begin{pmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \quad \left| \quad \begin{array}{l} T_{12} = \begin{pmatrix} c_2 & -s_2 & 0 & r_2 c_2 \\ s_2 & c_2 & 0 & r_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{34} = \begin{pmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{56} = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \right.$$

De la même façon, on peut exprimer les matrices  $T_{i;i-1} = T_{i-1;i}^{-1}$  :

$$\begin{array}{l} T_{10} = \begin{pmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{32} = \begin{pmatrix} c_3 & s_3 & 0 & -r_3 \\ -s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{54} = \begin{pmatrix} c_5 & s_5 & 0 & 0 \\ 0 & 0 & -1 & d_5 \\ -s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \quad \left| \quad \begin{array}{l} T_{21} = \begin{pmatrix} c_2 & s_2 & 0 & -r_2 \\ -s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{43} = \begin{pmatrix} c_4 & s_4 & 0 & 0 \\ 0 & 0 & 1 & -d_4 \\ s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ T_{65} = \begin{pmatrix} c_6 & s_6 & 0 & 0 \\ -s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & -d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \right.$$

## 2 Modèle géométriques direct

A partir des matrices précédentes, on obtient :

$$T_{06} = T_{01}T_{12}T_{23}T_{34}T_{45}T_{56} = \begin{pmatrix} R_{06} & P_{06} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{F.2})$$

Avec

$$R_{06} = \begin{pmatrix} c_6(s_1s_5 + c_{234}c_1c_5) - c_1s_{234}s_6 & -s_6(s_1s_5 + c_{234}c_1c_5) - c_1c_6s_{234} & c_5s_1 - c_{234}c_1s_5 \\ -c_6(c_1s_5 - c_{234}c_5s_1) - s_{234}s_1s_6 & s_6(c_1s_5 - c_{234}c_5s_1) - c_6s_{234}s_1 & -c_1c_5 - c_{234}s_1s_5 \\ c_{234}s_6 + c_5c_6s_{234} & c_{234}c_6 - c_5s_{234}s_6 & -s_{234}s_5 \end{pmatrix},$$

$$P_{06} = \begin{pmatrix} d_6(c_5s_1 - c_{234}c_1s_5) + d_4s_1 + d_5c_1s_{234} + r_3c_1c_{23} + r_2c_1c_2 \\ d_5s_{234}s_1 - d_6(c_1c_5 + c_{234}s_1s_5) - d_4c_1 + r_3s_1c_{23} + r_2s_1c_2 \\ d_1 - d_5c_{234} - d_6s_{234}s_5 + r_3s_{23} + r_2s_2 \end{pmatrix}.$$

## 3 Modèle géométrique inverse

Ici on décompose les système plan du reste pour cela, on s'intéresse à la matrice  $T_{14}$ . Avant cela, on s'intéresse à la matrice  $T_{16}$

**Détermination de  $\theta_1$**

$$\begin{aligned} T_{10}U_0^d &= T_{12}T_{23}T_{34}T_{45}T_{56} \\ \Leftrightarrow U1 &= T_1^6 \\ &= \begin{pmatrix} s_x c_1 + s_y s_1 & n_x c_1 + n_y s_1 & a_x c_1 + a_y s_1 & P_x c_1 + P_y s_1 \\ s_z & n_z & a_z & P_z - d_1 \\ s_x s_1 - s_y c_1 & n_x s_1 - n_y c_1 & a_x s_1 - a_y c_1 & P_x s_1 - P_y c_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} c_{234}c_5c_6 - s_{234}s_6 & -c_6s_{234} - c_{234}c_5s_6 & -c_{234}s_5 & d_5s_{234} + r_3c_{23} + r_2c_2 - d_6s_5c_{234} \\ c_{234}s_6 + c_5c_6s_{234} & c_{234}c_6 - c_5s_{234}s_6 & -s_{234}s_5 & r_3s_{23} - d_5c_{234} + r_2s_2 - d_6s_5s_{234} \\ c_6s_5 & -s_5s_6 & c_5 & d_4 + d_6c_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

De l'équation matricielle ci-dessus, on peut extraire le système suivant :

$$\begin{cases} c_5 & = a_x s_1 - a_y c_1 \\ d_4 + d_6 c_5 & = P_x s_1 - P_y c_1 \end{cases} \longrightarrow \begin{cases} d_4 & = (P_x - d_6 a_x) s_1 - (P_y - d_6 a_y) c_1 \end{cases}$$

On a donc :

$$\boxed{\begin{cases} \theta_1 = \text{Atan}_2(s_1, c_1) \\ s_1 = \frac{(P_x - d_6 a_x) d_4 \pm (d_6 a_y - P_y) \sqrt{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2 - d_4^2}}{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2} \\ c_1 = \frac{(d_6 a_y - P_y) d_4 \mp (P_x - d_6 a_x) \sqrt{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2 - d_4^2}}{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2} \end{cases}} \quad (\text{F.3})$$

On peut voir qu'il y a ici deux solutions pour  $\theta_1$ , ce qui correspond au placement du bras «à gauche» ou «à droite».

### Détermination de $\theta_5$

Grâce à l'équation précédente, on peut exprimer  $\theta_1$  en fonction des paramètres géométriques et désirés. On le considère donc comme connu. Or en remarquant que  $c_5 = a_x s_1 - a_y c_1$  on en déduit :

$$\boxed{\theta_5 = \pm \cos^{-1}(a_x s_1 - a_y c_1)} \quad (\text{F.4})$$

### Détermination de $\theta_6$

On peut désormais exprimer  $\theta_1$  et  $\theta_5$  en fonction des paramètres géométriques et désirés. Comme précédemment comme on a :  $c_6 s_5 = s_x s_1 - s_y c_1$  et  $-s_5 s_6 = n_x s_1 - n_y c_1$  on peut dire que :

$$\boxed{\theta_6 = \text{Atan}_2\left(-\frac{n_x s_1 - n_y c_1}{s_5}, \frac{s_x s_1 - s_y c_1}{s_5}\right)} \quad (\text{F.5})$$

Les liaisons 2, 3 et 4 étant parallèles, on regarde la position du repère 4 dans la base 1.

### Détermination de $\theta_2$

$$\begin{aligned} U1T_6^5 T_5^4 &= T_1^2 T_2^3 T_3^4 \\ &= \begin{pmatrix} c_5(u_{11}c_6 - u_{12}s_6) - u_{13}s_5 & u_{13}c_5 + s_5(u_{11}c_6 - u_{12}s_6) & -u_{12}c_6 - u_{11}s_6 & u_{14} - u_{13}d_6 + d_5(u_{12}c_6 + u_{11}s_6) \\ c_5(u_{21}c_6 - u_{22}s_6) - u_{23}s_5 & u_{23}c_5 + s_5(u_{21}c_6 - u_{22}s_6) & -u_{22}c_6 - u_{21}s_6 & u_{24} - u_{23}d_6 + d_5(u_{22}c_6 + u_{21}s_6) \\ c_5(u_{31}c_6 - u_{32}s_6) - u_{33}s_5 & u_{33}c_5 + s_5(u_{31}c_6 - u_{32}s_6) & -u_{32}c_6 - u_{31}s_6 & u_{34} - u_{33}d_6 + d_5(u_{32}c_6 + u_{31}s_6) \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} c_4 c_{23} - s_4 s_{23} & 0 & c_4 s_{23} + s_4 c_{23} & r_2 c_2 + r_3 c_{23} \\ c_4 s_{23} + s_4 c_{23} & 0 & s_4 s_{23} - c_4 c_{23} & r_2 s_2 + r_3 s_{23} \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &\quad \begin{cases} r_2 c_2 + r_3 c_{23} = u_{14} - u_{13} d_6 + d_5 (u_{12} c_6 + u_{11} s_6) \\ r_2 s_2 + r_3 s_{23} = u_{24} - u_{23} d_6 + d_5 (u_{22} c_6 + u_{21} s_6) \end{cases} \end{aligned}$$

En remarquant que :

$$(r_2 s_2 + r_3 s_{23})^2 + (r_2 c_2 + r_3 c_{23})^2 = r_2^2 + 2r_2 r_3 c_3 + r_3^2$$

On peut dire que :

$$\theta_3 = \pm \text{Arccos} \left( \frac{[u_{14} - u_{13}d_6 + d_5(u_{12}c_6 + u_{11}s_6)]^2 + [u_{24} - u_{23}d_6 + d_5(u_{22}c_6 + u_{21}s_6)]^2 - r_2^2 - r_3^2}{2r_2r_3} \right)$$

$$\begin{cases} (r_2 + r_3c_3)c_2 - r_3s_3s_2 = u_{14} - u_{13}d_6 + d_5(u_{12}c_6 + u_{11}s_6) \\ (r_2 + r_3c_3)s_2 + r_3s_3c_2 = u_{24} - u_{23}d_6 + d_5(u_{22}c_6 + u_{21}s_6) \end{cases} \Leftrightarrow \begin{cases} B_1c_2 - B_2s_2 = Z_1 \\ B_1s_2 + B_2c_2 = Z_2 \end{cases}$$

$$\theta_2 = \text{Atan}_2 \left( \frac{B_1Z_2 - B_2Z_1}{B_1^2 + B_2^2}, \frac{B_1Z_1 + B_2Z_2}{B_1^2 + B_2^2} \right)$$

$$\begin{cases} c_4s_{23} + s_4c_{23} = -u_{12}c_6 - u_{11}s_6 \\ s_4s_{23} - c_4c_{23} = -u_{22}c_6 - u_{21}s_6 \end{cases}$$

De la même façon :

$$\theta_4 = \pi + \text{Atan}_2(s_{23}[u_{22}c_6 + u_{21}s_6] - c_{23}[u_{12}c_6 + u_{11}s_6], s_{23}(u_{12}c_6 + u_{11}s_6) + c_{23}(u_{22}c_6 + u_{21}s_6))$$

$$\left\{ \begin{array}{l}
\theta_1 = + \operatorname{Atan}_2(s_1, c_1) \\
\theta_2 = + \operatorname{Atan}_2\left(\frac{B_1 Z_2 - B_2 Z_1}{B_1^2 + B_2^2}, \frac{B_1 Z_1 + B_2 Z_2}{B_1^2 + B_2^2}\right) \\
\theta_3 = \pm \operatorname{Arccos}\left(\frac{[u_{14} - u_{13} d_6 + d_5 (u_{12} c_6 + u_{11} s_6)]^2 + [u_{24} - u_{23} d_6 + d_5 (u_{22} c_6 + u_{21} s_6)]^2 - r_2^2 - r_3^2}{2r_2 r_3}\right) \\
\theta_4 = \pi + \operatorname{Atan}_2(s_{23} [u_{22} c_6 + u_{21} s_6] - c_{23} [u_{12} c_6 + u_{11} s_6], s_{23} (u_{12} c_6 + u_{11} s_6) + c_{23} (u_{22} c_6 + u_{21} s_6)) \\
\theta_5 = \pm \operatorname{Acos}(a_x s_1 - a_y c_1) \\
\theta_6 = + \operatorname{Atan}_2\left(-\frac{n_x s_1 - n_y c_1}{s_5}, \frac{s_x s_1 - s_y c_1}{s_5}\right)
\end{array} \right.$$

Avec

(F.6)

$$\left\{ \begin{array}{l}
s_1 = \frac{(P_x - d_6 a_x) d_4 \pm (d_6 a_y - P_y) \sqrt{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2 - d_4^2}}{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2} \\
c_1 = \frac{(d_6 a_y - P_y) d_4 \mp (P_x - d_6 a_x) \sqrt{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2 - d_4^2}}{(P_x - d_6 a_x)^2 + (d_6 a_y - P_y)^2} \\
B_1 = r_2 + r_3 c_3 \\
B_2 = r_3 s_3 \\
Z_1 = u_{14} - u_{13} d_6 + d_5 (u_{12} c_6 + u_{11} s_6) \\
Z_2 = u_{24} - u_{23} d_6 + d_5 (u_{22} c_6 + u_{21} s_6)
\end{array} \right.$$