Mémoire de stage

DEA Photonique, Image et Cybernétique

10 septembre 2004

Odile BOURQUARDEZ

Asservissement visuel et localisation d'un hélicoptère miniature

Maître de stage : François Chaumette, directeur de recherche INRIA et responsable de l'équipe Lagadic de l'IRISA

ENSPS/ULP
Parc d'innovation
Bd. S. Brant-BP10413
67412 ILLKIRCH

IRISA/INRIA
Campus universitaire de Beaulieu
35042 RENNES Cedex

Remerciements

Un grand merci à François Chaumette qui m'a permis d'effectuer mon stage au sein de l'équipe Lagadic de l'IRISA, et qui a encadré mon travail avec attention.

Je tiens également à remercier Fabien Spindler et Eric Marchand pour leur disponibilité et leur aide précieuse.

Merci à Nicolas Guénard (CEA/LIST) pour avoir consacré du temps à mes questions concernant le simulateur.

Enfin, la convivialité, le dynamisme et la sympathie qui émanent de l'équipe Lagadic ont constitué un cadre de travail agréable et motivant tout au long de mon stage. Merci donc également à Nicolas, Cédric, Anne-Sophie, Muriel, Andrew, Anthony, Omar, Anasse, Pei-Hua et Stéphanie pour leurs conseils, leurs remarques, leur amitié ou tout simplement leur bonne humeur.

Résumé

Si de nombreux travaux de recherche concernant les robots mobiles ont été menés ces dernières années, les robots volants semblent avoir été moins étudiés, particulièrement en France. Le projet national Robea Robvolint, qui a pour objectif de mettre au point un petit hélicoptère capable d'évoluer de manière autonome dans un environnement intérieur, permettra également de faire progresser la recherche dans ce domaine.

Ainsi, mon stage a pour but d'étudier des méthodes de localistaion par vision et d'asservissement visuel qui pourront être utilisées par l'hélicoptère. Des expérimentations ont été réalisées sur le robot cartésien de l'IRISA, après y avoir implanté le simulateur d'hélicoptère développé par le laboratoire concepteur de l'hélicoptère. Des résultats intéressants ont été obtenus avec la méthode développée pour la localisation, et les expérimentations d'asservissement visuel ont permis de tirer des conclusions qui pourront éclairer de futurs travaux.

Abstract

I have undertook my training period in the Lagadic team at IRISA. The work of this research group mainly deals with visual servoing, robotics, computer vision, and augmented reality. This team participates in the national project Robea Robvolint ; this project is realized through a collaboration between several french laboratories. Its aim is to progress knowledge in the field of aerial robotics with an application in indoor environments. This project has the objective to realize a small helicopter capable of exploring an indoor environment autonomously.

The aim of my work is to study methods for vision-based localization and visual servoing which could be used by the helicopter. I have tested experimentedly the algorithms on the cartesian robot of IRISA, which can simulate the behaviour of the helicopter. Interesting results were obtained with the developed method for localization, and the experiments with visual servoing allow us to draw some conclusions which will be used in the continuation of the project.

Table des matières

Table des figures

Introduction

Le projet national Robea (RObotique et Entités Artificielles) Robvolint (RObot VOLant d'INTérieur) a pour but de développer les équipements électroniques, informatiques et sensoriels d'un petit hélicoptère électrique à quatre rotors (X4-flyer, figure 1) et également de développer différentes fonctionnalités lui permettant d'évoluer de manière autonome dans un environnement d'intérieur. Mon stage s'inscrit dans ce cadre et a pour but de permettre à l'hélicoptère, équipé d'une caméra embarquée, de connaître sa position par rapport à son environnement, ainsi que de réaliser des tâches d'asservissement visuel. Les expérimentations sont dans un premier temps menées sur le robot cartésien de l'IRISA, simulant le comportement de l'hélicoptère. Plus tard, et en dehors du cadre de mon stage, des expérimentations réelles seront réalisées au CEA/LIST (Fontenay Aux Roses, 92), qui travaille sur la réalisation du X4-flyer.

FIG. 1 – Hélicoptère de type X4-flyer

1 Contexte

1.1 Le projet Robvolint

Ces dernières années, les projets de recherche portant sur les machines volantes n'ont cessé de se multiplier, tant en Europe qu'aux Etats-Unis. Ainsi, le projet Robea Robvolint fait intervenir quatre laboratoires (I3S : Informatique Signaux et Systèmes de Sophia Antipolis, IRISA à Rennes, CEA/LIST : Laboratoire d'Intégration des Systèmes et des Technologies, à Fontenay Aux Roses et IRCCyN : Institut de Recherche en Communications et en Cybernétique de Nantes), et a pour objectif principal de faire progresser les connaissances dans le domaine de la robotique aérienne en environnement d'intérieur. Le scénario envisagé est le suivant :
Il s'agit d'explorer un immeuble ou un tunnel contaminé ou accidenté. Un drône miniature est utilisé afin de réaliser une première observation avant l'entrée dans le bâtiment. Mis en œuvre par une seule personne, de maniement simple, il retransmet une image de l'intérieur des différentes pièces ouvertes permettant de détecter des éléments importants pour la mission (sauvetage, dégât, structure interne du bâtiment). Le comportement du drône est sûr grâce à l'évitement automatique d'obstacles présents et simplifié par un retour automatique vers l'opérateur à la fin de la mission.

Le choix de s'intéresser au vol en intérieur n'a rien de réducteur. En effet, contrairement à ce qu'on pourrait penser au premier abord, les contraintes dynamiques imposées par le vol en intérieur (effets de sol, plafond, ...) sont certes modélisables, mais difficiles à résoudre. De plus, les essais de vol en extérieur ne sont pas autorisés à proximité des laboratoires, pour des raisons de sécurité.

1.2 Le robot volant

Le type d'hélicoptère choisi pour mener à bien ce projet est un X4-flyer, drône principalement destiné au vol en intérieur, et dont la manipulation n'est pas dangeureuse. Il s'agit d'un petit hélicoptère électrique à quatre rotors ([16], figure 1).

L'hélicoptère sera équipé d'une centrale inertielle (gyroscope, accéléromètre), d'un proximètre, et d'une caméra. L'hélicoptère se devant d'être léger et de petite taille, il n'est pas possible de traiter les images de la caméra directement sur l'hélicoptère. Ainsi, il a été choisi de transmettre les images acquises par liaison HF (haute fréquence) vers un calculateur qui numérisera l'image et pourra faire les

traitements nécessaires. Un mode téléopération sera également mis en place, et les commandes issues de la manipulation du joystick seront transmises à l'hélicoptère par liaison HF également. Un DSP (Digital Signal Processing) embarqué assurera la commande du drône, en intégrant les données des capteurs, et les ordres de haut niveau émis par le poste de pilotage.

Le CEA a investi dans la mise en place d'un site expérimental. Il dispose de deux X4-flyer d'origine, de deux mini caméras dotées de leur système de transmission, d'un calculateur (PC) et d'une plate-forme expérimentale : 80 m^2 sur 5 m de hauteur, protégée par des filets.

Des expérimentations sur le robot cartésien de l'IRISA contraint par la dynamique de l'hélicoptère sont envisagées dans un premier temps pour valider les algorithmes de vision. Un simulateur du X4-flyer a été réalisé par le CEA. Par la suite, le CEA mettra à disposition sa plate-forme de test.

Différentes fonctionnalités devront à terme équiper l'hélicoptère, afin de lui permettre de réaliser les tâches envisagées. Ces fonctionnalités sont les suivantes :

- la commande manuelle assistée,
- l'anticollision automatique,
- l'asservissement visuel sur amers (artificiels et naturels),
- la localisation par vision sur amers (artificiels et naturels),
- une interface homme-machine.

1.3 Plan

Mon stage concerne les troisième et quatrième points, et doit apporter les premières solutions aux problèmes liés à l'asservissement visuel de l'hélicoptère, ainsi qu'à sa localisation dans son environnement.

Après un bref état de l'art, nous présenterons les résultats obtenus en implantant le simulateur développé par le CEA sur le robot cartésien de l'IRISA. Ensuite nous développerons la méthode utilisée pour permettre à l'hélicoptère de se localiser par vision, et enfin nous tirerons les conclusions des expérimentations réalisées avec différentes stratégies d'asservissement visuel.

2 Etat de l'art

2.1 Localisation

2.1.1 Principe

Il s'agit d'obtenir la position et l'orientation relative entre un engin et son environnement. Pour les robots mobiles, l'utilisation du GPS (Global Positioning System) peut paraître intéressante pour localiser l'engin. Cependant, l'utilisation d'une caméra permet une meilleure précision, et apporte la possibilité de s'adapter à l'environnement. Dans [19], les deux systèmes sont utilisés pour le contrôle d'un hélicoptère. Il s'agit d'une part de guider l'hélicoptère lors de la phase de vol (avec le GPS), et d'autre part de lui permettre de localiser la piste d'atterissage (avec la caméra).

Dans le cadre du projet Robvolint, on s'intéresse à la localisation par vision de l'hélicoptère : le calcul de pose consiste donc à utiliser l'image fournie par la caméra pour en déduire la position et l'orientation de l'engin par rapport à la scène observée.

2.1.2 Méthodes

Différents algorithmes ont été développés pour le calcul de pose. Les plus courants sont la méthode de Dementhon [5] (approche numérique itérative), les approches linéaires (par moindres carrés), et les approches non linéaires, qui minimisent, par des algorithmes numériques (Levenberg-Marquardt par exemple) la distance entre des éléments caractéristiques extraits de l'image observée et la projection du modèle dans cette image. Si les approches linéaires sont simples et ne nécessitent pas d'initialistion de leurs paramètres, elles restent cependant peu précises. Elles constituent principalement une bonne initialisation pour les méthodes non linéaires, plus stables et plus précises. Dans [4, 15], une méthode non linéaire de calcul de pose basée sur un asservissement visuel virtuel a été proposée. Comme pour toutes les méthodes citées plus haut, la connaissance du modèle CAO de l'objet observé est requise.

2.2 Asservissement visuel

2.2.1 Principe

Il s'agit d'utiliser les informations visuelles issues d'une caméra pour contrôler les mouvements d'un robot.

En fonction de la tâche que l'on souhaite accomplir, il est tout d'abord nécessaire de sélectionner les informations visuelles qui seront aptes à la réaliser de manière optimale. Il faut ensuite exprimer cette tâche sous forme de consigne s^* à atteindre par les informations visuelles courantes s. Une loi de commande appropriée est alors à élaborer pour calculer, à partir de la différence $s - s^*$, la consigne de vitesse à envoyer à l'engin. L'engin réagira à ces commandes selon sa dynamique, et l'image donnée par la caméra sera donc modifiée en conséquence. A partir des informations visuelles extraites de la nouvelle image, on peut alors calculer une nouvelle commande, et ce jusqu'à ce que l'on ait accompli la tâche désirée (figure 2).

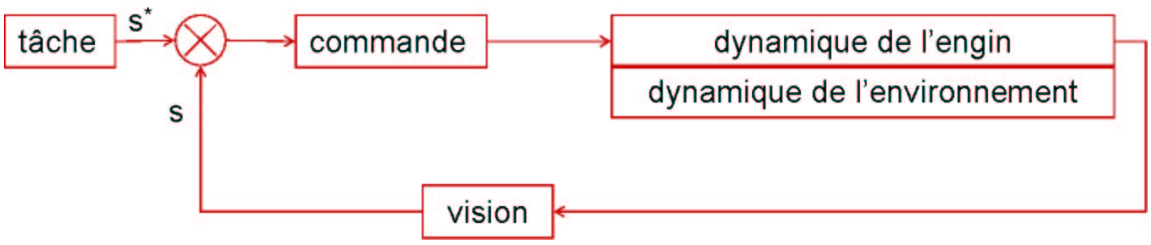


FIG. 2 – Schéma de principe de l'asservisssement visuel

Deux grandes familles d'asservissement visuel ont été mises au point [2, 9, 10] : ceux pour lesquels l'entrée de la commande est constituée d'entités 2D (asservissements visuels 2D), et ceux pour lesquels l'entrée de la commande est constituée d'entités 3D (asservissements visuels 3D). Si un des principaux avantages des asservissements visuels 2D est d'éviter la reconstruction de la pose, contrairement aux asservissements visuels 3D, la forme de la matrice d'interaction ne permet pas, bien souvent, d'obtenir un bon découplage des degrés de liberté (alors que ce découplage peut être obtenu dans les asservissements visuels 3D). De plus, si avec l'asservissement visuel 2D, le comportement des informations visuelles dans l'image est généralement satisfaisant (puisque le contrôle est fait à ce niveau), la trajectoire dans l'espace 3D peut être parfois peu satisfaisante lorsque de grands mouvements de translation et de rotation sont à réaliser.

En outre, l'asservissement visuel 2D présente l'avantage d'être robuste et stable. Ainsi, on s'orientera plutôt vers ce type d'asservissement, en cherchant néanmoins à atténuer ses défauts en sélectionnant au mieux les informations visuelles utilisées.

Dans [20], les auteurs proposent d'élaborer un schéma de commande découplé, et de réduire les non-linéarités dans la matrice d'interaction, afin d'avoir une meilleure trajectoire du robot. Ainsi, l'utilisation de moments invariants comme informations visuelles permet d'obtenir une matrice d'interaction présentant ces propriétés intéressantes. Par exemple les coordonnées normées du centre de gravité et la surface de l'objet permettent de commander les degrés de liberté de translation ; l'utilisation de moments d'ordre supérieur permettent la commande des degrés de liberté de rotation [3, 20].

2.2.2 Application aux systèmes sous-actionnés

L'hélicoptère est un exemple de système sous-actionné, puisqu'il a six degrés de liberté pour seulement quatre entrées de commande. Des recherches pour l'asservissement visuel 2D des systèmes sous-actionnés ont montré que l'exploitation d'informations visuelles bien adaptées aux contraintes du système permet de commander les mouvements de l'engin [8, 18].
Dans [8], la stratégie consiste à modéliser des informations visuelles dont la dynamique conserve la propriété de passivité structurelle de l'hélicoptère. Cette notion de passivité s'apparente au découplage entre degrés de liberté de rotation et de translation, et elle a été obtenue pour des points projetés sur une caméra sphérique [13]. Des expérimentations en simulation ont été menées en utilisant le moment d'ordre 1 (centre de gravité) comme information visuelle, connaissant l'orientation de l'hélicoptère par rapport à son environnement.

Par ailleurs, des travaux récents portants sur l'asservissement visuel d'un dirigeable [18] ont permis d'obtenir un découplage adapté à la dynamique de l'engin en utilisant comme informations visuelles le point de fuite et la ligne d'horizon.

2.3 Conclusion

Pour permettre la localisation par vision de l'hélicoptère, nous avons retenu la technique de calcul de pose par asservissement visuel virtuel [4, 15], que nous avons associée à un filtre de Kalman étendu, afin de filtrer le bruit de mesure et également de prédire l'état en l'absence de mesure. En effet, en l'absence de

caméra rapide, la mesure par vision ne donne un résultat que toutes les 40 ms, ce qui est insuffisant pour notre application.

Nous avons expérimenté les méthodes d'asservissement visuel utilisant les moments 2D [3, 20], puis la projection sur une surface sphérique [13], sur le robot cartésien contraint par la dynamique de l'hélicoptère.

3 Simulateur de l'hélicoptère

Le simulateur utilisé se présente sous la forme d'un logiciel développé par le CEA/LIST. Il permet de simuler le comportement dynamique du X4-flyer. A partir de quatre consignes spécifiées par l'utilisateur, ainsi que des mesures d'orientation et de vitesses instantanées du robot, le simulateur calcule les vitesses à envoyer au robot pour qu'il se déplace à la manière de l'hélicoptère. Les consignes sont les vitesses v_x, v_y, et v_z de l'hélicoptère, exprimées dans le repère qui lui est lié, et l'angle θ de lacet exprimé dans un repère fixe. Les mesures sont quant à elles données par l'odométrie du robot cartésien. Le simulateur actuel fonctionne avec une période d'échantillonage de 10 ms.

Nous avons utilisé ce simulateur de deux manières : d'une part en simulation pure, c'est-à-dire sans l'implémenter sur le robot, et en introduisant des mesures de positions et de vitesses théoriques non bruitées, calculées à partir des vitesses que fournit le simulateur. L'autre manière, plus réaliste, consiste à faire bouger le robot avec la dynamique de l'hélicoptère, les mesures étant fournies par l'odométrie.

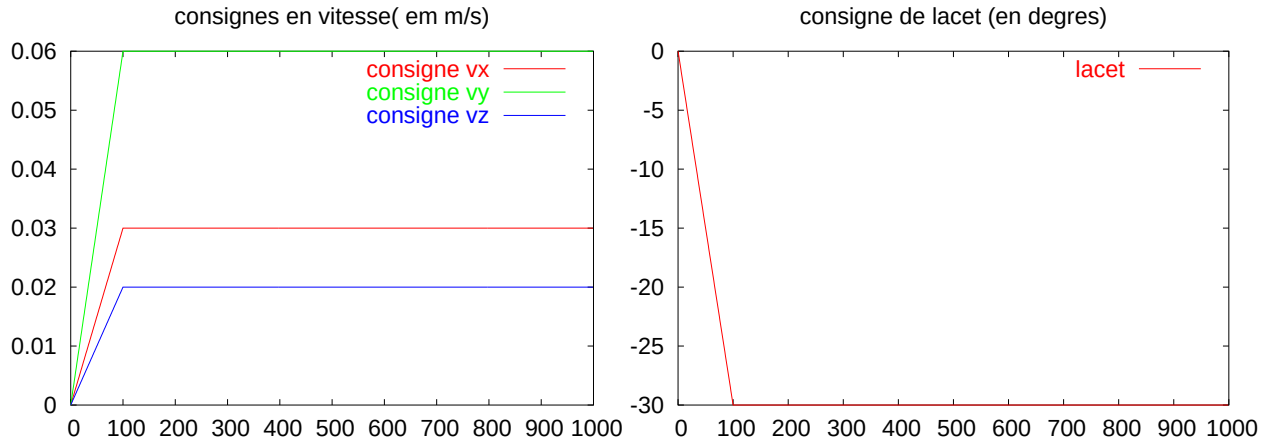
Les figures 3 à 7 présentent les courbes obtenues en simulation pure et en implémentant le simulateur sur le robot cartésien, avec les consignes suivantes :

- $v_x = 0.03$ m/s
- $v_y = 0.06$ m/s
- $v_z = 0.02$ m/s
- $\theta = -30$ degrés
- durée $= 10$ s

La figure 3 montre l'évolution des consignes fournies au simulateur. Nous avons choisi d'utiliser des rampes au lieu d'échelons afin d'éviter les pics trop grands sur les vitesses calculées par le simulateur (celui-ci étant assez réactif).

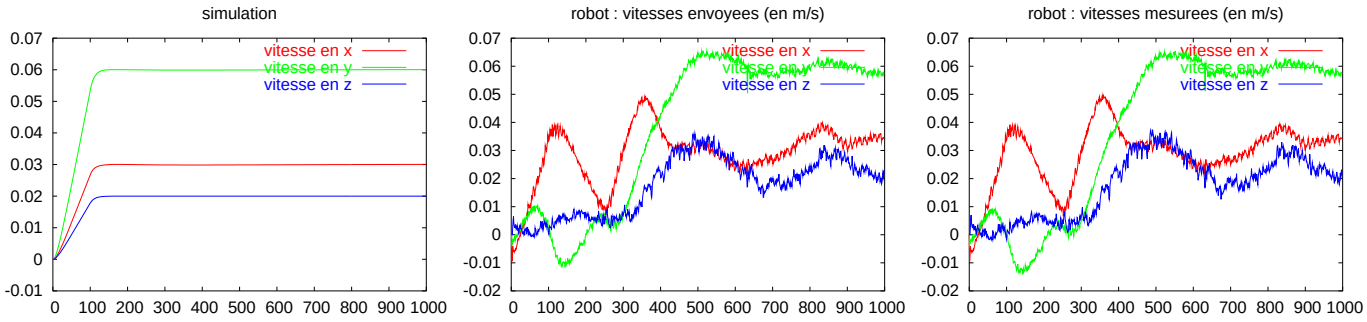
Les figures 4 à 7 permettent de comparer le comportement du simulateur lorsqu'il fonctionne avec des mesures théoriques parfaites et lorsqu'il envoye ses commandes au robot et que les mesures sont issues de l'odométrie de celui-ci. On voit que sur le robot, les transitoires sont allongés et les dépassements sont plus importants par rapport à la simulation pure. Les figures 4 et 5 montrent que les vitesses envoyées au robot et mesurées sont fortement bruitées par rapport à la simulation pure. Notament sur les figures 5.b et 5.c, on remarque également l'effet de lissage dû à la dynamique du robot cartésien : les vitesses envoyées sont plus bruitées que les vitesses effectivement réalisées par le robot. Les figures 6 et 7 représentent quant à elles les positions : le robot ne suit pas exactement les positions théoriques obtenues en simulation ; les courbes de position restent cependant « lisses » et peu

bruitées.



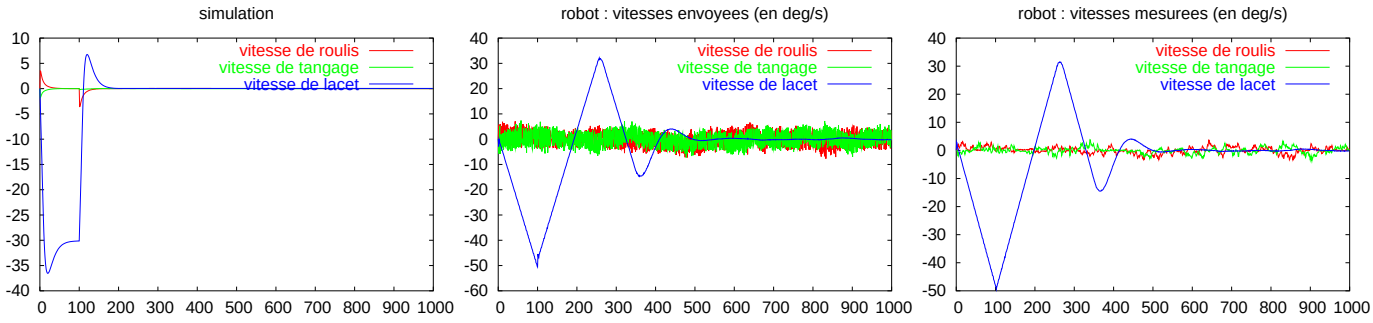
(a) (b)

FIG. 3 – Consignes, pour chaque itération du simulateur : (a) vitesses en x, y, z, (b) angle de lacet



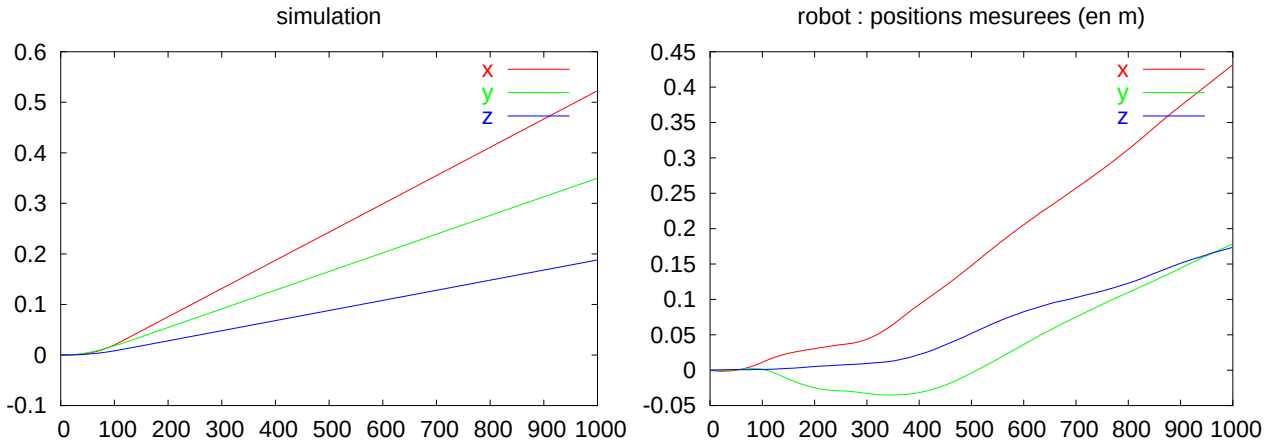
(a) (b) (c)

FIG. 4 – Vitesses selon les axes x, y et z du repère lié à l'hélicoptère : (a) vitesses calculées en simulation pure, (b) vitesses calculées par le simulateur et envoyées au robot, (c) vitesses mesurées sur le robot



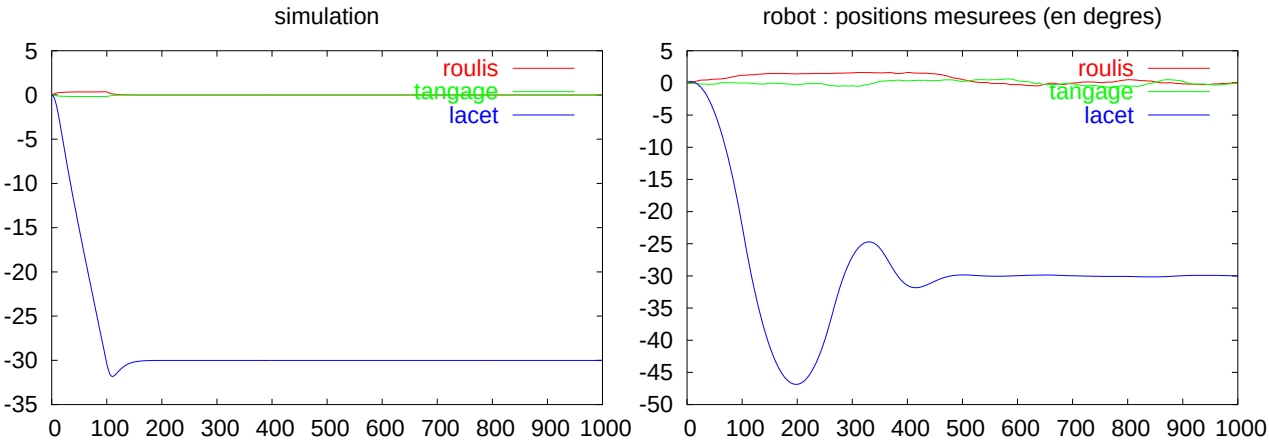
(a) (b) (c)

FIG. 5 – Vitesses angulaires dans le repère lié à l'hélicoptère : (a) vitesses calculées en simulation pure, (b) vitesses calculées par le simulateur et envoyées au robot, (c) vitesses mesurées sur le robot



(a) (b)

FIG. 6 – Positions selon les axes X, Y et Z du repère fixe : (a) positions calculées en simulation pure, (b) positions mesurées sur le robot



(a) (b)

FIG. 7 – Positions angulaires dans le repère fixe: (a) positions calculées en simulation pure, (b) positions mesurées sur le robot

4 Localisation de l'hélicoptère

Sur un bras manipulateur, l'odométrie peut permettre de connaître la position et l'orientation de l'organe terminal du robot, que l'on souhaite contrôler. Il n'en est pas de même dans le cas d'un hélicoptère, où un capteur extéroceptif est nécessaire pour localiser l'engin. Ainsi, une caméra, rigidement liée à l'hélicoptère, fournit une image, dont l'exploitation des caractéristiques visuelles va permettre de calculer la position et l'orientation de la cible observée par rapport à la caméra (calcul de la pose). La mesure de la pose, introduite dans un filtre de Kalman, nous a permis de localiser l'hélicoptère dans son environnement. Nous présenterons donc tout d'abord la méthode de calcul de pose utilisée, puis l'utilisation du filtre de Kalman, qui a permis de filtrer le bruit de mesure et d'estimer la pose en l'absence de mesure. Enfin, les résultats obtenus sur le robot cartésien contraint par la dynamique de l'hélicoptère sont présentés.

4.1 Calcul de pose

4.1.1 Principe

Nous avons choisi d'utiliser une méthode non linéaire : l'asservissement visuel virtuel [4, 15].

Une caméra virtuelle est déplacée par asservissement visuel jusqu'à ce que la projection du modèle CAO de l'objet, connaissant la pose courante de la caméra virtuelle, coïncide avec l'image observée par la caméra réelle. Les deux caméras (réelle et virtuelle) sont alors superposées, et la position finale de la caméra virtuelle correspond donc finalement à une estimation très fiable de la position de la caméra réelle. Il s'agit en fait de minimiser la fonction $\boldsymbol{e} = {}^{C}\boldsymbol{p} - {}^{C}\boldsymbol{M}_{O}{}^{O}\boldsymbol{P}$, en déplacant la caméra virtuelle par un asservissement visuel 2D classique, où :

- ${}^{C}\boldsymbol{p}$ représente l'information visuelle extraite de l'image réelle ;
- ${}^{C}\boldsymbol{M}_{O}$ la matrice homogène représentant la position de l'objet dans le repère de la caméra virtuelle (donc ${}^{C}\boldsymbol{M}_{O}$ est connu puisque la position de la caméra virtuelle est connue) ;
- ${}^{O}\boldsymbol{P}$ les paramètres du modèle CAO de l'objet.

${}^{C}\boldsymbol{M}_{O}{}^{O}\boldsymbol{P}$ est donc l'information visuelle extraite de la caméra virtuelle (image virtuelle obtenue par projection du modèle CAO), que l'on souhaite faire correspondre à l'information issue de l'image réelle ${}^{C}\boldsymbol{p}$.

Lorsqu'on atteint la position telle que $\boldsymbol{e} \approx \boldsymbol{0}$, les deux caméras (réelle et virtuelle) sont alors superposées, et la position finale de la caméra virtuelle cor-

respond à la position de la caméra réelle, que l'on cherchait à estimer.

4.1.2 Mise en œuvre

Pour notre application, nous avons utilisé la méthode de calcul de pose par asservissement visuel virtuel, qui donne des résultats précis, mais nécessite une initialisation afin que le déplacement à réaliser par la caméra virtuelle ne soit pas trop important (condition classique pour les asservissements visuels 2D [2, 10]). Pour chaque nouveau calcul, on prend donc la position finale précédente comme position initiale de la caméra virtuelle. Pour la première image, on initialise cette position par un premier résultat de calcul de pose obtenu avec la méthode de Dementhon [5].

La cible utilisée est constituée de points blancs sur fond noir, dont la position sur la cible est connue ; les informations visuelles considérées sont les coordonnées des points.

Toutes les 40 ms (c'est-à-dire à la cadence vidéo), une nouvelle image est traitée, et la matrice ${}^{C}\boldsymbol{M}_{O}$, représentant la position et l'orientation de la cible dans le repère de la caméra, est estimée.

4.2 Filtrage de Kalman

Le simulateur du comportement de l'hélicoptère calcule une nouvelle consigne toutes les 10 ms, en utilisant les mesures d'orientation et de vitesse de l'hélicoptère. Le calcul de pose, par traitement d'image, ne fournit quant à lui qu'une mesure toutes les 40 ms. De plus, cette mesure est bruitée. Le filtre de Kalman est un outil couramment utilisé dans ce type de situation : il permet de filtrer efficacement les bruits de mesure, en utilisant un modèle d'état du système considéré. Nous avons donc, à partir d'un modèle simple du comportement de l'hélicoptère, en modélisant les bruits d'état, et en mesurant les bruits de mesure, mis en œuvre un filtre de Kalman. Nous avons ainsi pu non seulement filtrer le bruit présent dans la mesure de la pose, mais aussi prédire une estimation de la position toutes les 10 ms, et ce même en l'absence de mesure (c'est-à-dire trois fois sur quatre en pratique).

4.2.1 Définitions

Pour la modélisation de notre système, nous avons supposé que la vitesse de rotation de l'hélicoptère est constante, et que son accélération en translation est

constante. Les termes d'ordre supérieur sont considerés comme du bruit.

On note $\mathcal{R}_{\mathcal{O}}$ le repère lié à l'objet, et $\mathcal{R}_{\mathcal{C}(t)}$ le repère lié à la caméra à l'instant t.

On définit les variables constituant le vecteur d'état :

– $\boldsymbol{p}$ représente la position de la cible, exprimée dans $\mathcal{R}_{\mathcal{C}(t)}$,

– $\boldsymbol{\theta}$ représente l'orientation de la cible, exprimée dans $\mathcal{R}_{\mathcal{C}(t)}$,

– $\boldsymbol{v}$ représente la vitesse de translation de la caméra, exprimée dans $\mathcal{R}_{\mathcal{C}(t)}$,

– $\boldsymbol{\omega}$ représente la vitesse de rotation de la caméra, exprimée dans $\mathcal{R}_{\mathcal{C}(t)}$,

– $\boldsymbol{a}$ représente l'accélération de translation de la caméra, exprimée dans $\mathcal{R}_{\mathcal{C}(t)}$.

On note

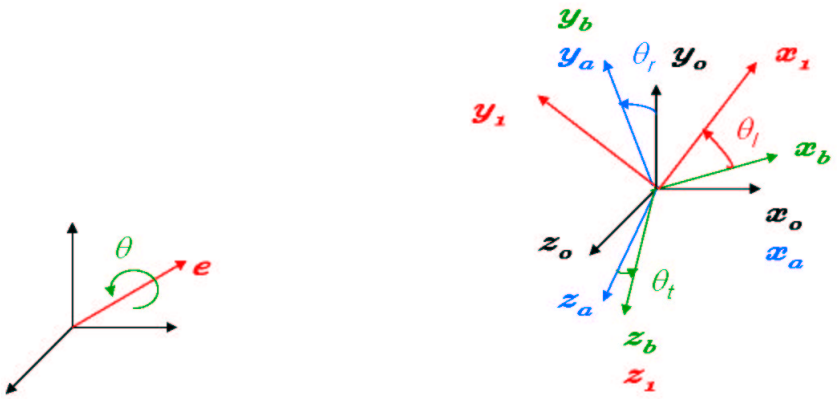
$$\boldsymbol{u}(t) = \begin{pmatrix} \boldsymbol{p}(t) \\ \boldsymbol{\theta}(t) \\ \boldsymbol{v}(t) \\ \boldsymbol{\omega}(t) \\ \boldsymbol{a}(t) \end{pmatrix} \tag{1}$$

Plusieurs paramétrisations peuvent être utilisées pour représenter l'orientation de la cible $\boldsymbol{\theta}$: représentation axe-angle, roulis-tangage-lacet, angles d'Euler, matrice de rotation, etc. Pour les calculs et la mise en place des équations d'état, nous avons utilisé la représentation axe-angle (figure 8.a) : rotation d'un angle θ autour d'un axe unitaire $\boldsymbol{e}$. On a donc $\boldsymbol{e} = \frac{\boldsymbol{\theta}}{\theta}$ et $\theta = \|\boldsymbol{\theta}\|$, où $\boldsymbol{\theta}$ est le vecteur représentant la rotation. Les résultats obtenus lors de la mise en œuvre du filtre sont par contre exprimés avec les angles d'Euler XYZ (figure 8.b) : on décompose la rotation entre les repères R_0 et R_1 en trois rotations successives d'angles θ_r, θ_t, θ_l autour des axes successifs x_0, y_a et z_b. Cette représentation est non redondante (de dimension trois) ; elle est utilisée par le simulateur, ainsi que par certains modules de calcul de pose et de commande du robot cartésien.

4.2.2 Equations d'état du modèle

En considérant un modèle à vitesse de rotation et accélération en translation constantes, nous allons établir les équations d'état du système correspondant. Ces hypothèses ne sont dans la pratique pas forcément vérifiées, et on va donc par la suite modéliser les variations par rapport au modèle par l'introduction de bruit d'état.

Dans [12], les équations du mouvement d'une cible mobile dans le repère de la caméra sont données, sous les hypothèses de vitesse de rotation de la cible constante, et d'accélération en translation constante. En adaptant ces équations au



(a) (b)

FIG. 8 – Représentations de la rotation : (a) représentation axe-angle, (b) angles d'Euler XYZ

cas qui nous intéresse (cible fixe et caméra en mouvement), on obtient :

$$\dot{\boldsymbol{p}}(t) = -\boldsymbol{v} - \boldsymbol{\omega} \times \boldsymbol{p} - \boldsymbol{a}t \tag{2}$$

L'intégration de cette équation conduit à l'équation d'état pour la partie translation [12, 23] :

$$\boldsymbol{p}(t+dt) = \boldsymbol{R}(-\boldsymbol{\omega}(t)dt)\boldsymbol{p}(t) - \boldsymbol{S}(-\boldsymbol{\omega}(t)dt)\boldsymbol{v}(t)dt - \boldsymbol{T}(-\boldsymbol{\omega}(t)dt)\boldsymbol{a}(t)\frac{dt^2}{2} \tag{3}$$

avec

$$\boldsymbol{R}(\boldsymbol{\theta}) = \boldsymbol{I_3} + \frac{\sin\theta}{\theta}\mathrm{sk}(\boldsymbol{\theta}) + \frac{1-\cos\theta}{\theta^2}\mathrm{sk}^2(\boldsymbol{\theta}) \tag{4}$$

$$\boldsymbol{S}(\boldsymbol{\theta}) = \boldsymbol{I_3} + \frac{1-\cos\theta}{\theta^2}\mathrm{sk}(\boldsymbol{\theta}) + \frac{\theta-\sin\theta}{\theta^3}\mathrm{sk}^2(\boldsymbol{\theta}) \tag{5}$$

$$\boldsymbol{T}(\boldsymbol{\theta}) = \boldsymbol{I_3} + 2\frac{\theta-\sin\theta}{\theta^3}\mathrm{sk}(\boldsymbol{\theta}) + \frac{\theta^2-2(1-\cos\theta)}{\theta^4}\mathrm{sk}^2(\boldsymbol{\theta}) \tag{6}$$

où $\boldsymbol{I_3}$ est la matrice identité de rang 3, $\theta = \|\boldsymbol{\theta}\|$ et $\mathrm{sk}(\boldsymbol{\theta})$ est la matrice antisymétrique associée à $\boldsymbol{\theta}$:

$$\mathrm{sk}(\boldsymbol{\theta}) = \begin{pmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{pmatrix} \tag{7}$$

Notons le couplage entre les termes de rotation et de translation : $\boldsymbol{p}$ est fonction des termes de translation et de $\boldsymbol{\omega}$. On peut également remarquer que $\boldsymbol{R}(\boldsymbol{\theta})$ n'est rien d'autre que la formule de Rodrigues, permettant de passer de la représentation d'une rotation sous la forme vecteur de rotation à la matrice de rotation associée.

Pour la partie rotation, on obtient (voir l'annexe C pour le détail des calculs) :

$$\boldsymbol{\theta}(t+dt) = \Phi(\boldsymbol{R}^T(\boldsymbol{\omega}(t)dt)\boldsymbol{R}(\boldsymbol{\theta}(t))) \tag{8}$$

Les dernières équations d'état sont évidentes :

$$\begin{aligned} \boldsymbol{v}(t+dt) &= \boldsymbol{v}(t) + \boldsymbol{a}(t)dt & (9)\\ \boldsymbol{\omega}(t+dt) &= \boldsymbol{\omega}(t) & (10)\\ \boldsymbol{a}(t+dt) &= \boldsymbol{a}(t) & (11) \end{aligned}$$

L'équation (10) est donnée par l'hypothèse « rotation à vitesse constante », et les équations (9) et (11) découlent de l'hypothèse « accélération de translation constante ».

On peut finalement écrire, à partir de (3), (8), (9), (10), et (11), le modèle d'état sous la forme

$$\boldsymbol{u}(t+dt) = f(\boldsymbol{u}(t)) = \begin{pmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \\ \boldsymbol{f}_4 \\ \boldsymbol{f}_5 \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{R}(-\boldsymbol{\omega}(t)dt)\boldsymbol{p}(t) - \boldsymbol{S}(-\boldsymbol{\omega}(t)dt)\boldsymbol{v}(t)dt - \boldsymbol{T}(-\boldsymbol{\omega}(t)dt)\boldsymbol{a}(t)\frac{dt^2}{2} \\ \Phi(\boldsymbol{R}(\boldsymbol{\omega}(t)dt)^T\boldsymbol{R}(\boldsymbol{\theta}(t))) \\ \boldsymbol{v}(t) + \boldsymbol{a}(t)dt \\ \boldsymbol{\omega}(t) \\ \boldsymbol{a}(t) \end{pmatrix} \tag{12}$$

L'équation (12) permet donc de décrire complètement l'évolution du système, si son mouvement respecte les hypothèses de vitesse de rotation et accélération en translation constantes. Dans la pratique, ces hypothèses ne seront pas forcément vérifiées. Afin d'avoir un modèle plus réaliste, nous allons donc modéliser les écarts par rapport à ce modèle en lui ajoutant du bruit.

4.2.3 Bruit d'état et de mesure

Pour modéliser le bruit d'état, nous avons négligé le couplage entre le bruit sur la rotation et sur la translation. En d'autres termes, nous avons considéré que la matrice de covariance contient des zéros au niveau des termes « croisés ».

En identifiant le bruit des termes d'ordre supérieur négligés dans le modèle (vitesse de rotation constante, accélération de translation constante), on obtient pour la matrice $\boldsymbol{Q}$ représentant la covariance du bruit d'état (voir l'annexe D pour le détail des calculs) :

$$\boldsymbol{Q} = \begin{pmatrix} \sigma_t^2 \frac{dt^6}{36}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_t^2 \frac{dt^5}{12}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_t^2 \frac{dt^4}{6}\boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \sigma_r^2 \frac{dt^4}{4}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_r^2 \frac{dt^3}{2}\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \sigma_t^2 \frac{dt^5}{12}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_t^2 \frac{dt^4}{4}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_t^2 \frac{dt^3}{2}\boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \sigma_r^2 \frac{dt^3}{2}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_r^2 dt^2 \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \sigma_t^2 \frac{dt^4}{6}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_t^2 \frac{dt^3}{2}\boldsymbol{I}_3 & \boldsymbol{0}_3 & \sigma_t^2 dt^2 \boldsymbol{I}_3 \end{pmatrix} \tag{13}$$

où

$$\boldsymbol{I}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{0}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{14}$$

Seuls σ_t^2 et σ_r^2 sont alors à choisir pour fixer le bruit d'état.

L'équation d'état du filtre de Kalman est constituée du modèle considéré (12), auquel on ajoute le bruit d'état (13) :

$$\boldsymbol{u}(t+dt) = f(\boldsymbol{u}(t)) + \boldsymbol{q} \tag{15}$$

où $\boldsymbol{Q}$ est la matrice de covariance de $\boldsymbol{q}$.

Pour l'équation de mesure, on a :

$$\boldsymbol{z}(t) \;\; = \;\; \boldsymbol{H}\boldsymbol{u}(t) + \boldsymbol{m} \tag{16}$$

avec

$$\boldsymbol{z}(t) = \begin{pmatrix} \boldsymbol{p}(t) \\ \boldsymbol{\theta}(t) \end{pmatrix}, \quad \boldsymbol{u}(t) = \begin{pmatrix} \boldsymbol{p}(t) \\ \boldsymbol{\theta}(t) \\ \boldsymbol{v}(t) \\ \boldsymbol{\omega}(t) \\ \boldsymbol{a}(t) \end{pmatrix}, \quad \boldsymbol{H} = \begin{pmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{pmatrix}$$

On notera $\boldsymbol{M}$ la matrice de covariance du bruit de mesure $\boldsymbol{m}$. Dans la pratique, le bruit de mesure peut être mesuré. Pour ce faire, nous avons supposé le bruit constant, et nous avons calculé la variance des mesures obtenues lorsque les consignes sont à zéro.

4.2.4 Equations du filtre

Le système considéré est décrit par le système d'équations suivant (équations (15) et (16)) :

$$\left\{ \begin{array}{lcl} \boldsymbol{u}(t+dt) & = & f(\boldsymbol{u}(t)) + \boldsymbol{q} \\ \boldsymbol{z}(t) & = & \boldsymbol{H}\boldsymbol{u}(t) + \boldsymbol{m} \end{array} \right. \tag{17}$$

où f est la fonction non linéaire définie par (12). f étant non linéaire, il est nécessaire d'utiliser un filtre de Kalman étendu [22]. L'étape de prédiction (20) demande de linéariser f autour du vecteur d'état courant à chaque itération. On définit donc la matrice $\boldsymbol{A}$ telle que

$$\boldsymbol{A}_{[i,j]}(t) = \frac{\partial \boldsymbol{f}_{[i]}}{\partial \boldsymbol{u}_{[j]}}(\boldsymbol{u}(t)) \qquad \forall (i,j) \in [1, \ldots 5] \tag{18}$$

Ainsi, les équations classiques du filtre de Kalman deviennent :

– étape de prédiction :

$$\boldsymbol{u}^{-}(t+dt) = f(\boldsymbol{u}(t)) \tag{19}$$

$$\boldsymbol{P}^{-}(t+dt) = \boldsymbol{A}(t)\boldsymbol{P}(t)\boldsymbol{A}^{T}(t) + \boldsymbol{Q} \tag{20}$$

– étape de filtrage :

$$\boldsymbol{K}(t+dt) = \boldsymbol{P}^{-}(t+dt)\boldsymbol{H}^{T}(\boldsymbol{S}(t+dt))^{-1} \tag{21}$$

$$\boldsymbol{u}(t+dt) = \boldsymbol{u}^{-}(t+dt) + \boldsymbol{K}(t+dt)[\boldsymbol{z}(t+dt) - \boldsymbol{H}\boldsymbol{u}^{-}(t+dt)] \tag{22}$$

$$\boldsymbol{P}(t+dt) = \boldsymbol{P}^{-}(t+dt) - \boldsymbol{K}(t+dt)\boldsymbol{S}(t+dt)\boldsymbol{K}^{T}(t+dt) \tag{23}$$

$$\text{avec} \qquad \boldsymbol{S}(t+dt) = \boldsymbol{H}\boldsymbol{P}^{-}(t+dt)\boldsymbol{H}^{T} + \boldsymbol{M} \tag{24}$$

En pratique, nous avons utilisé une autre manière de calculer $\boldsymbol{P}(t+dt)$, numériquement plus stable [1, 12] :

$$\boldsymbol{P}(t+dt) = \left[(\boldsymbol{P}^{-}(t+dt))^{-1} + \boldsymbol{H}^{T}\boldsymbol{M}^{-1}\boldsymbol{H}\right]^{-1} \tag{25}$$

$\boldsymbol{P}^-$ est la variance de l'erreur d'estimation avant mesure, et $\boldsymbol{P}$ après mesure ; $\boldsymbol{K}$ est le gain du filtre de Kalman ; $\boldsymbol{s}^-$ représente l'état avant mesure et $\boldsymbol{s}$ l'état après mesure, pour une itération donnée.

On peut remarquer que le gain $\boldsymbol{K}$, qui dépend directement des matrices de covariance $\boldsymbol{Q}$ et $\boldsymbol{M}$ va permettre de donner un poids plus ou moins important au terme d'innovation $\boldsymbol{z}(t+dt) - \boldsymbol{H}\,\boldsymbol{s}^-(t+dt)$ face au terme de prédiction $\boldsymbol{s}^-(t+dt)$. En effet, on peut écrire $\boldsymbol{K}$ sous la forme

$$\boldsymbol{K} = \frac{\boldsymbol{P}^- \boldsymbol{H}^T}{\boldsymbol{H}\,\boldsymbol{P}^- \boldsymbol{H}^T + \boldsymbol{M}} \tag{26}$$

On voit alors que

- si $\boldsymbol{M}$ est petit et $\boldsymbol{P}^-$ grand (peu de bruit sur la mesure, et fortes perturbations sur l'état) :

$$\boldsymbol{K} \simeq \boldsymbol{H}^{-1} \qquad \text{d'où} \qquad \boldsymbol{u} \simeq \boldsymbol{H}^{-1}\boldsymbol{z} \tag{27}$$

  Dans ce cas, on fait confiance à la mesure plutôt qu'au modèle pour estimer l'état $\boldsymbol{s}$.
- si $\boldsymbol{M}$ est grand et $\boldsymbol{P}^-$ petit (beaucoup de bruit sur la mesure, et peu de perturbations sur l'état) :

$$\boldsymbol{K} \simeq 0 \qquad \text{d'où} \qquad \boldsymbol{u} \simeq \boldsymbol{u}^- \tag{28}$$

  Dans ce cas, on fait confiance au modèle plutôt qu'à la mesure pour estimer l'état $\boldsymbol{s}$.

On voit bien l'importance et l'influence des choix de $\boldsymbol{Q}$ et $\boldsymbol{M}$ sur le filtre.

Afin de mettre en œuvre les équations du filtre, nous avons cherché l'expression analytique de la matrice $\boldsymbol{A}$ définie par (18).

4.2.5 Calcul de $\boldsymbol{A}$

On cherche

$$\boldsymbol{A} = \begin{pmatrix} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{a}} \\ \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{a}} \\ \frac{\partial \boldsymbol{f}_3}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{f}_3}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{f}_3}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{f}_3}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{f}_3}{\partial \boldsymbol{a}} \\ \frac{\partial \boldsymbol{f}_4}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{f}_4}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{f}_4}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{f}_4}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{f}_4}{\partial \boldsymbol{a}} \\ \frac{\partial \boldsymbol{f}_5}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{f}_5}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{f}_5}{\partial \boldsymbol{v}} & \frac{\partial \boldsymbol{f}_5}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{f}_5}{\partial \boldsymbol{a}} \end{pmatrix} \tag{29}$$

où les $\boldsymbol{f}_i$ sont définis en (12).

Les trois dernières lignes de la matrice sont évidentes compte tenu de l'expression de $\boldsymbol{f}_3$, $\boldsymbol{f}_4$ et $\boldsymbol{f}_5$ donnée par (12). On peut voir également que $\boldsymbol{f}_2$ est indépendant de $\boldsymbol{p}$, $\boldsymbol{v}$ et $\boldsymbol{a}$, et que $\boldsymbol{f}_1$ est indépendant de $\boldsymbol{\theta}$. De plus, les dérivées partielles de $\boldsymbol{f}_1$ par rapport à $\boldsymbol{p}$, $\boldsymbol{v}$, et $\boldsymbol{a}$ sont simples. On a donc

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R}(-\boldsymbol{\omega}(t)dt) & \boldsymbol{0}_3 & -\boldsymbol{S}(-\boldsymbol{\omega}(t)dt)dt & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\omega}} & -\boldsymbol{T}(-\boldsymbol{\omega}(t)dt)\frac{dt^2}{2} \\ \boldsymbol{0}_3 & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\theta}} & \boldsymbol{0}_3 & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\omega}} & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 dt \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{pmatrix} \quad (30)$$

Nous avons alors envisagé deux manières de calculer les termes restants : un calcul exact et un calcul approché.

Pour le calcul exact, nous avons utilisé le logiciel de calcul formel Maple. Pour le calcul approché, il s'agit de raisonner en vitesses au lieu de raisonner sur les positions, ce qui revient à approcher les dérivées partielles au premier ordre. (Voir en annexe E les résultats pour les deux cas de figure considérés).

Comparaison des deux méthodes Nous avons testé les performances du filtre avec les deux méthodes de calcul de $\boldsymbol{A}$. Les résultats sont très similaires (figures 9 et 10) et le filtrage est bien réalisé dans les deux cas. La méthode approchée est donc intéressante, puisqu'elle donne de bons résultats, tout en réduisant significativement la quantité de calculs nécessaire.

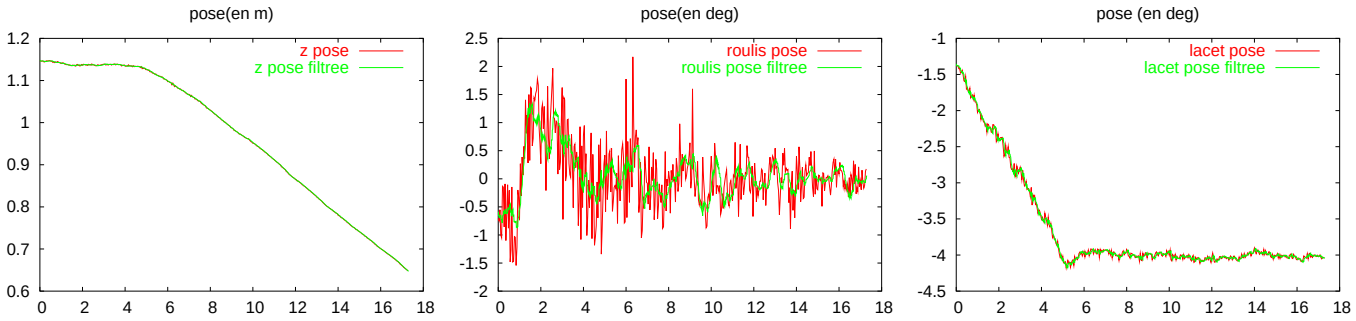


FIG. 9 – Filtrage en utilisant le calcul exact de $\boldsymbol{A}$

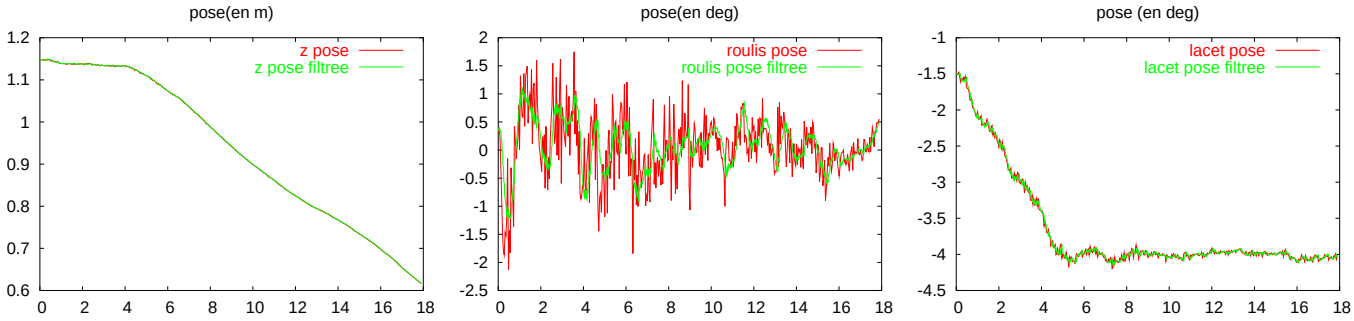


FIG. 10 – Filtrage en utilisant le calcul approché de $\boldsymbol{A}$

4.3 Résultats

L'utilisation classique d'un filtre de Kalman recquiert une nouvelle mesure à chaque itération, pour calculer le nouvel état. Or, nous n'avons à notre disposition qu'une mesure toutes les 40 ms, et nous souhaitons connaître l'état toutes les 10 ms. Nous avons envisagé plusieurs stratégies pour résoudre ce problème :

- utiliser le filtre de Kalman à une période d'échantillonnage de 10 ms, en lui donnant, en l'absence de véritable mesure (trois fois sur quatre), la valeur prédite $\boldsymbol{H}\boldsymbol{u}^{-}(t+dt)$ en guise de mesure $\boldsymbol{z}(t+dt)$;
- utiliser le filtre de Kalman à une période d'échantillonnage de 10 ms, en donnant un poids quasi nul à la mesure lorsque celle-ci n'est pas une véritable mesure (donc trois fois sur quatre) ; pour ce faire, on modifie les covariances $\boldsymbol{Q}$ et $\boldsymbol{M}$ des bruits.

Ces deux méthodes s'avèrent perturber fortement le fonctionnement du filtre de Kalman (temps de réponse allongé, retard important), puisque l'on modifie ses paramètres lorsqu'on n'a pas de mesure. De ce fait, les résultats ne sont pas satisfaisants.

Nous avons donc finalement retenu la stratégie suivante pour la mise en œuvre du filtre de Kalman avec les mesures de pose :

- les covariances des bruits du filtre de Kalman sont fixées à l'initialisation (13) ;
- la période du filtre de Kalman est donnée par la période à laquelle une nouvelle mesure est disponible (40 ms) ;
- chaque fois qu'une nouvelle mesure $\boldsymbol{z}(t+dt)$ est disponible (toutes les 40 ms), on passe par les étapes de prédiction puis filtrage (19, 20, 21, 22, 23), pour calculer le nouveau vecteur d'état $\boldsymbol{u}(t+dt)$;

– toutes les 10 ms, soit l'état est donné par le filtre de Kalman (une fois sur quatre), soit on se contente d'utiliser l'équation d'état non bruitée (12) uniquement pour prédire le nouvel état, sans influencer les covariances ni même l'état pris en compte dans le filtre de Kalman.

Ainsi, toutes les 10 ms, le vecteur d'état nous permet d'accéder à la position de l'objet dans le repère de la caméra (la pose). Donc, en l'inversant, on peut également accéder à la position de la caméra dans le repère fixe (localisation de l'hélicoptère).

Les figures 12 à 16 ont été obtenues de la manière précédemment décrite, sur le robot cartésien contraint par la dynamique de l'hélicoptère. Les consignes envoyées sont :

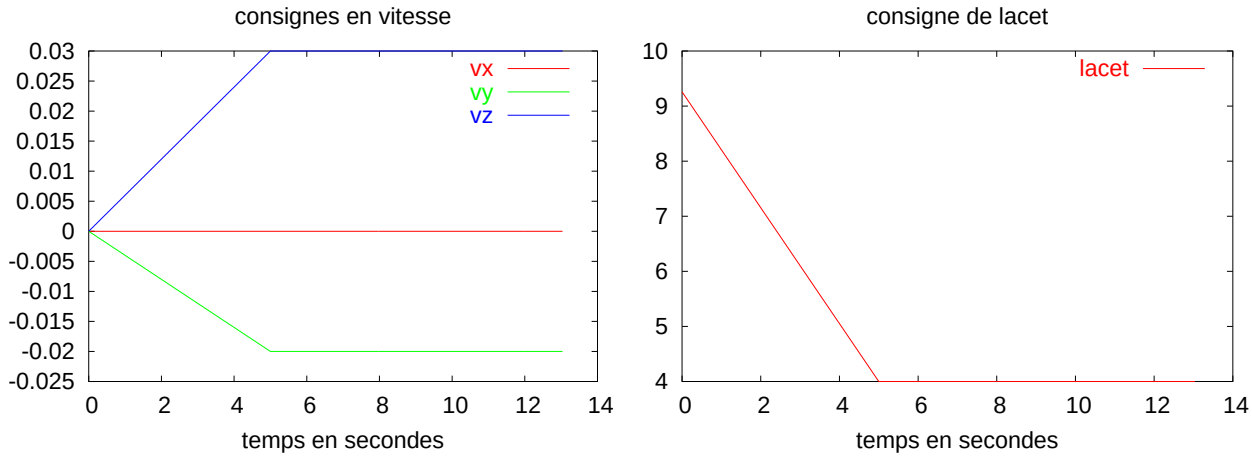
– $v_x = 0$ m/s
– $v_y = -0.02$ m/s
– $v_z = 0.03$ m/s
– $\theta = 4$ degrés

Ces consignes sont envoyées sous la forme de rampes (figure 11), le palier étant atteint en 5 secondes.

Les figures 12 et 13 illustrent bien les effets du filtre de Kalman sur la pose. On remarque que les prédictions en l'absence de mesure donnent des valeurs cohérentes sur la figure 12, et le filtrage du bruit de mesure est bien visible sur la figure 13.

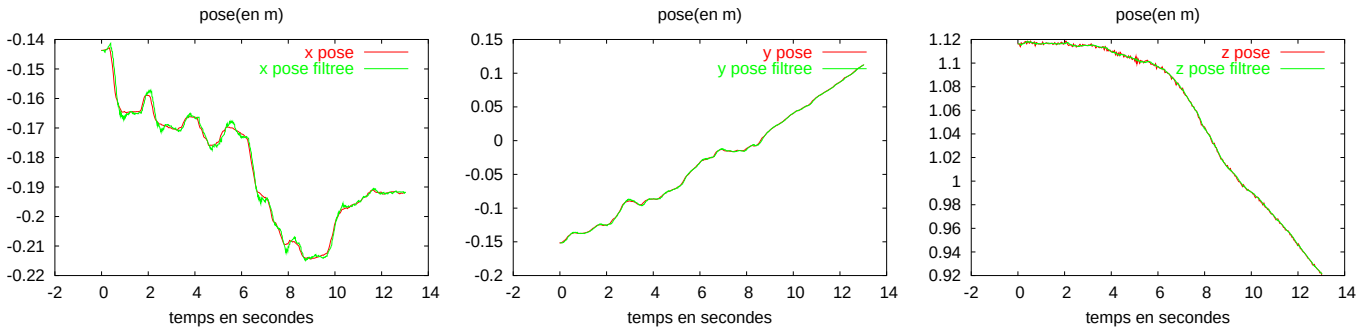
On peut voir sur la figure 16 que les résultats obtenus pour la position et l'orientation de l'hélicoptère sont tout à fait valables et exploitables. Les courbes présentées ici sont d'ailleurs obtenues en injectant ces valeurs comme mesure de l'orientation de l'hélicoptère dans le simulateur.

Enfin, la figure 14 montre que les valeurs de vitesses obtenues grâce au calcul de pose sont fortement bruitées sur la partie translation et on peut voir sur la figure 15 qu'elles présentent un retard important (200 ms environ) sur la partie rotation. Ces problèmes ne nous ont pas permis d'exploiter ces mesures de vitesse dans le simulateur. Concrètement, nous cherchions surtout à pouvoir localiser l'hélicoptère par la vision. Les mesures de vitesses pourront être réalisées par d'autres capteurs (accéléromètre et gyromètre) dont l'hélicoptère sera équipé.



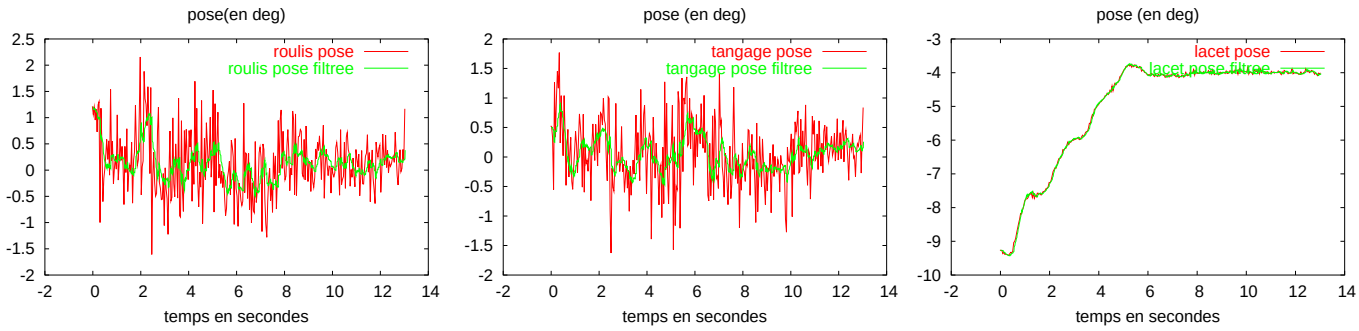
(a) (b)

FIG. 11 – Consignes sous formes de rampes, données en entrée du simulateur : (a) vitesses selon les axes x, y, z du repère lié à l'hélicoptère, (b) angle de lacet



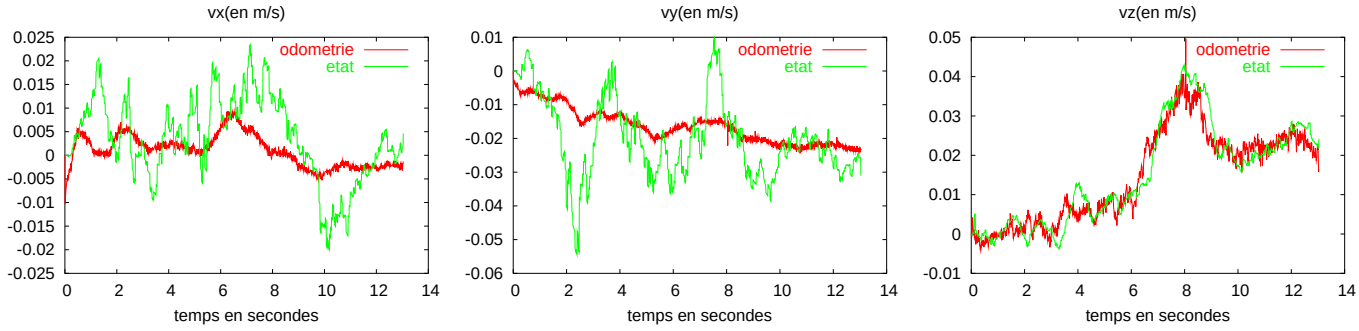
(a) (b) (c)

FIG. 12 – Partie translation du vecteur d'état : en rouge, les valeurs mesurées toutes les 40 ms, et en vert les valeurs calculées toutes les 10 ms



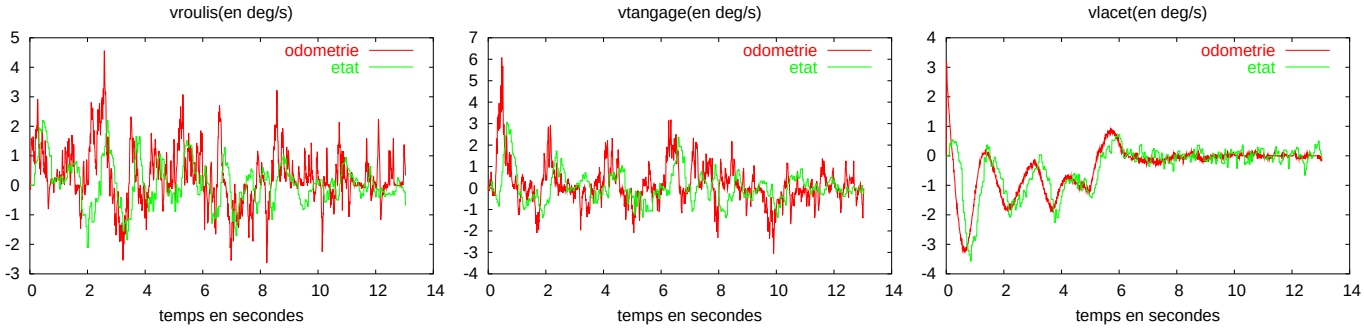
(a) (b) (c)

FIG. 13 – Partie rotation du vecteur d'état (représentation par les angles d'Euler) : en rouge, les valeurs mesurées toutes les 40 ms, et en vert les valeurs calculées toutes les 10 ms



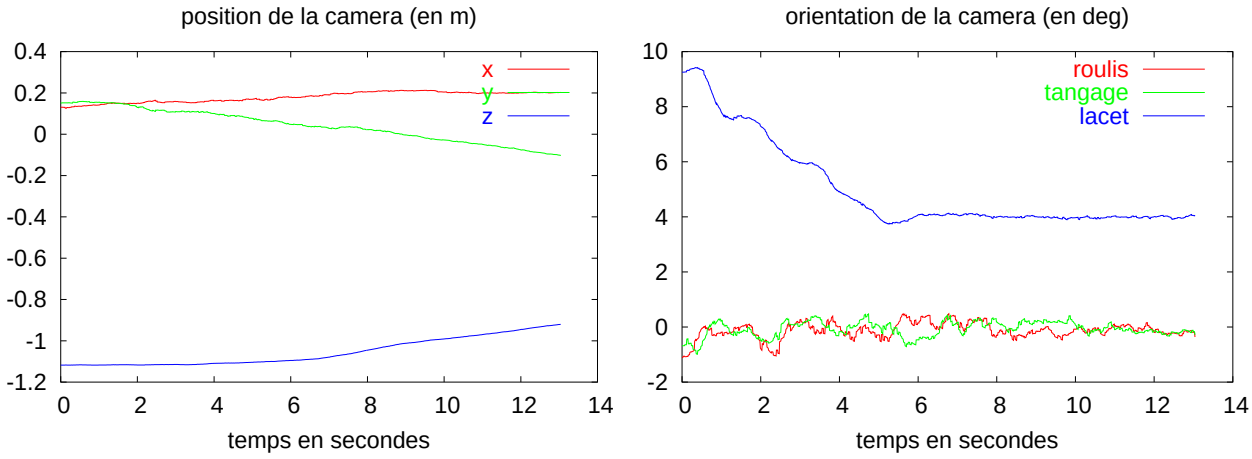
(a) (b) (c)

FIG. 14 – Vitesses de translation de l'hélicoptère, dans le repère qui lui est lié : en rouge, les valeurs données par l'odométrie du robot, et en vert les valeurs obtenues dans le vecteur d'état



(a) (b) (c)

FIG. 15 – Vitesses de rotation de l'hélicoptère, dans le repère qui lui est lié (représentation par les angles d'Euler) : en rouge, les valeurs données par l'odométrie du robot, et en vert les valeurs obtenues dans le vecteur d'état



(a) (b)

FIG. 16 – Position et orientation de l'hélicoptère dans le repère fixe (lié à la cible), obtenues à partir du vecteur d'état : (a) position selon les axes x, y, z, (b) orientation en roulis-tangage-lacet

4.4 Conclusion et perspectives

Un calcul de pose par asservissement visuel virtuel, associé à un filtre de Kalman étendu, nous a permis d'obtenir une mesure fiable de la position et de l'orientation de l'hélicoptère par rapport à une cible connue.

Nous nous sommes ici attachés essentiellement aux problèmes liés au filtrage des mesures et à l'estimation en l'absence de mesure. De ce fait, les développements réalisés pouraient être utilisés par la suite sur des mesures obtenues par un calcul de pose robuste aux erreurs éventuelles de traitement d'image, aux occultations et aux variations d'éclairement [4]. Une autre approche pourra consister à considérer des scènes dont le modèle n'est pas connu, pour lesquelles des techniques de calcul de pose robustes sont également développées [17].

5 Asservissement visuel de l'hélicoptère

Contrairement à un bras manipulateur à six degrés de liberté et six entrées de commande (classiquement utilisé pour les applications d'asservissemenet visuel), l'hélicoptère possède certes six degrés de mobilité, mais seulement quatre entrées de commande. De ce fait, le choix des informations visuelles est particulièrement important : idéalement, il faudrait utiliser des informations visuelles qui ne soient pas modifiées par les mouvements de l'engin selon les degrés de liberté non contrôlés. Ainsi, les mouvements de roulis et de tangage de l'hélicoptère n'affecteraient pas le comportement de la boucle d'asservissement. Il serait également intéressant de découpler les degrés de liberté que l'on contrôle. Hélas, trouver des informations visuelles possédant ces bonnes propriétés est loin d'être évident.

5.1 Utilisation des moments 2D

5.1.1 Principe

Une première approche a consisté a utiliser les moments de l'image [3, 20] comme informations visuelles. Ayant à notre disposition quatre entrées de commande (v_x, v_y, v_z et ω_l), nous avons utilisé quatre informations visuelles [20] :

$$a_n = Z^* \sqrt{\frac{a^*}{a}}, \quad x_n = a_n x_g, \quad y_n = a_n y_g \quad \text{et} \quad \theta = \frac{1}{2}\arctan\Big(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\Big) \quad (31)$$

où

- x_g et y_g représentent les coordonnées du centre de gravité de l'image ;
- a représente la surface de l'objet dans l'image ;
- a^* représente la surface désirée de l'objet dans l'image ;
- Z^* représente la profondeur désirée entre l'objet et la caméra ;
- μ_{ij} représentent les moments centrés de l'image.

L'avantage de ces informations visuelles réside essentiellement dans la forme des matrices d'interaction correspondantes (on considère le cas où l'objet est parallèle au plan image) [20] :

$$\begin{array}{rcl}
\boldsymbol{L}_{x_n} & = & \begin{pmatrix} -1 & 0 & 0 & a_n\epsilon_{11} & -a_n(1+\epsilon_{12}) & y_n \end{pmatrix} \\
\boldsymbol{L}_{y_n} & = & \begin{pmatrix} 0 & -1 & 0 & a_n(1+\epsilon_{21}) & -a_n\epsilon_{11} & -x_n \end{pmatrix} \\
\boldsymbol{L}_{a_n} & = & \begin{pmatrix} 0 & 0 & -1 & -3y_n/2 & -3x_n/2 & 0 \end{pmatrix} \\
\boldsymbol{L}_{\theta} & = & \begin{pmatrix} 0 & 0 & 0 & \theta_{wx} & \theta_{wy} & -1 \end{pmatrix}
\end{array} \quad (32)$$

On se reportera à [20] pour les expressions de ϵ_{11}, ϵ_{12}, ϵ_{21}, θ_{wx} et θ_{wy}. La forme « creuse » de la matrice d'interaction permet de découpler partiellement les degrés de liberté. La translation selon X est principalement liée à x_n, selon Y à y_n, selon Z à a_n, et l'orientation de lacet à θ.

Notons $\boldsymbol{s}$ le vecteur d'informations visuelles à quatre composantes

$$\boldsymbol{s} = (x_n, y_n, a_n, \theta) \tag{33}$$

Soit $\boldsymbol{q_1}$ la partie du torseur cinématique que l'on commande, et $\boldsymbol{q_2}$ la partie que l'on ne commande pas :

$$\boldsymbol{q_1} = (v_x, v_y, v_z, \omega_l) \tag{34}$$

$$\boldsymbol{q_2} = (\omega_r, \omega_t) \tag{35}$$

On a alors

$$\dot{\boldsymbol{s}} = \boldsymbol{L_1 q_1} + \boldsymbol{L_2 q_2} \tag{36}$$

avec

$$\boldsymbol{L_1} = \begin{pmatrix} -1 & 0 & 0 & y_n \\ 0 & -1 & 0 & -x_n \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{37}$$

$$\boldsymbol{L_2} = \begin{pmatrix} a_n\epsilon_{11} & -a_n(1+\epsilon_{12}) \\ a_n(1+\epsilon_{21}) & -a_n\epsilon_{11} \\ -3y_n/2 & -3x_n/2 \\ \theta_{wx} & \theta_{wy} \end{pmatrix} \tag{38}$$

Soit l'erreur $\boldsymbol{e}$ à minimiser :

$$\boldsymbol{e} = -\lambda(\boldsymbol{s} - \boldsymbol{s}^*) \tag{39}$$

où λ représente un gain proportionnel.
Comme on cherche à faire décroître exponentiellement l'erreur, on veut $\dot{\boldsymbol{e}} = -\lambda \boldsymbol{e}$, c'est-à-dire $\dot{\boldsymbol{s}} = -\lambda(\boldsymbol{s} - \boldsymbol{s}^*)$.

On en déduit, avec (36)

$$\boldsymbol{q_1} = \boldsymbol{L_1}^{-1}(-\lambda(\boldsymbol{s} - \boldsymbol{s}^*) - \boldsymbol{L_2 q_2}) \tag{40}$$

La loi de commande qui permettra de faire décoître exponentiellement l'erreur $\boldsymbol{e}$ est donnée par (40). On remarque qu'elle contient le terme $\boldsymbol{L_2 q_2}$ qui agit un peu à la manière d'une perturbation. On peut alors soit estimer $\boldsymbol{q_2}$ et en tenir compte dans la loi de commande, soit considérer ce terme négligeable. Nous avons opté pour la deuxième solution, et avons donc utilisé la loi de commande

$$\boldsymbol{q_1} = -\lambda {\boldsymbol{L_1}}^{-1}(\boldsymbol{s} - \boldsymbol{s}^*) \tag{41}$$

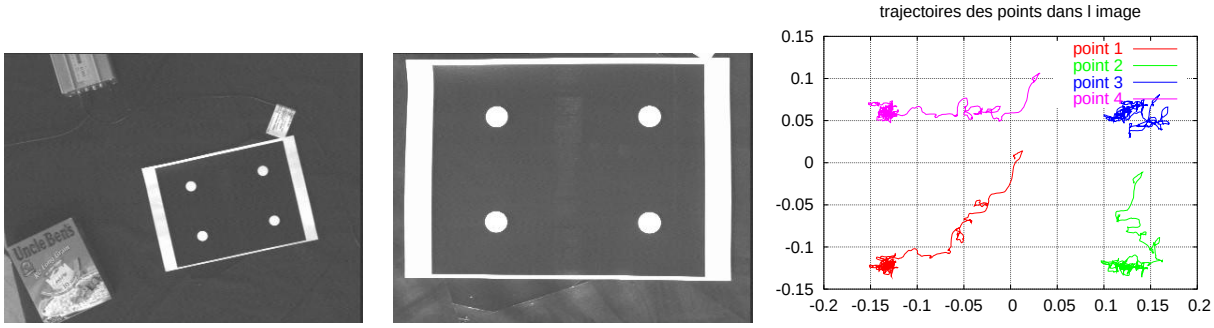
5.1.2 Mise en œuvre et résultats

La loi de commande de l'asservissement visuel nous fournit v_x, v_y, v_z, et ω_l, toutes les 40 ms. Chaque fois qu'une nouvelle consigne est ainsi calculée, on transmet directement v_x, v_y et v_z au simulateur, et on calcule l'angle de lacet à atteindre en 40 ms à partir de ω_l. Ainsi, pendant 40 ms, les consignes sont maintenues constantes.

La mire utilisée est constituée de quatre points blancs sur fond noir, formant un rectangle. Le traitement d'image permet de connaître les coordonnées des points à chaque itération. La consigne spécifiée consiste à placer la caméra à 50 cm au dessus de la cible, celle-ci étant centrée et orientée dans le même sens que la caméra. On souhaite donc que les coordonnées des points dans l'image soient : $(-0.14, -0.1), (0.14, -0.1), (0.14, 0.1), (-0.14, 0.1)$ (voir figure 17.b). Le gain λ de la loi de commande a été fixé à 0.15.

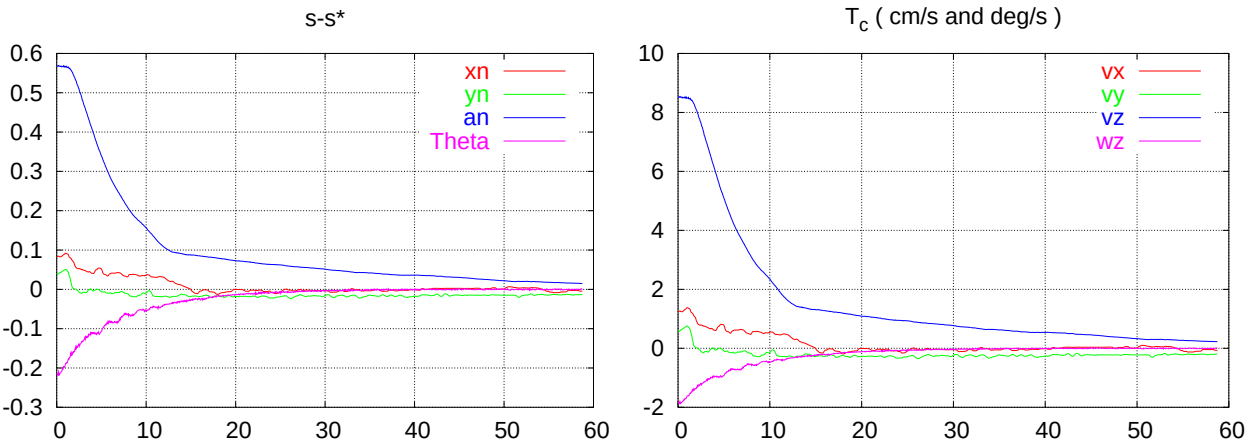
Les figures 17 à 19 présentent les résultats obtenus. On observe que l'erreur sur les informations visuelles (figure 18.a) décroit exponentiellement vers zéro, tout comme les vitesses calculées par l'assservissement visuel (figure 18.b). Cependant, du fait de la dynamique de l'hélicoptère, celui-ci continue à bouger légèrement même une fois la convergence atteinte. On observe bien ce phénomène sur la figure 17.c, où l'on voit que une fois l'état final atteint, la position des points dans l'image continue à évoluer autour de la position finale.

De plus, du fait de la dynamique de l'hélicoptère, on observe une erreur entre la position finale atteinte dans l'image et la position souhaitée (figures 17.b et 17.c). Par exemple, le point 1 oscille autour de $(-0.14, -0.14)$, alors qu'il devrait se trouver en $(-0.14, -0.1)$, le point 3 oscille autour de $(0.14, 0.07)$, alors qu'il devrait se trouver en $(0.14, 0.1)$, et il en est de même pour les autres points. On voit également sur la figure 19 que les vitesses du robot ne restent pas nulles, particulièrement au niveau des angles de roulis et de tangage.



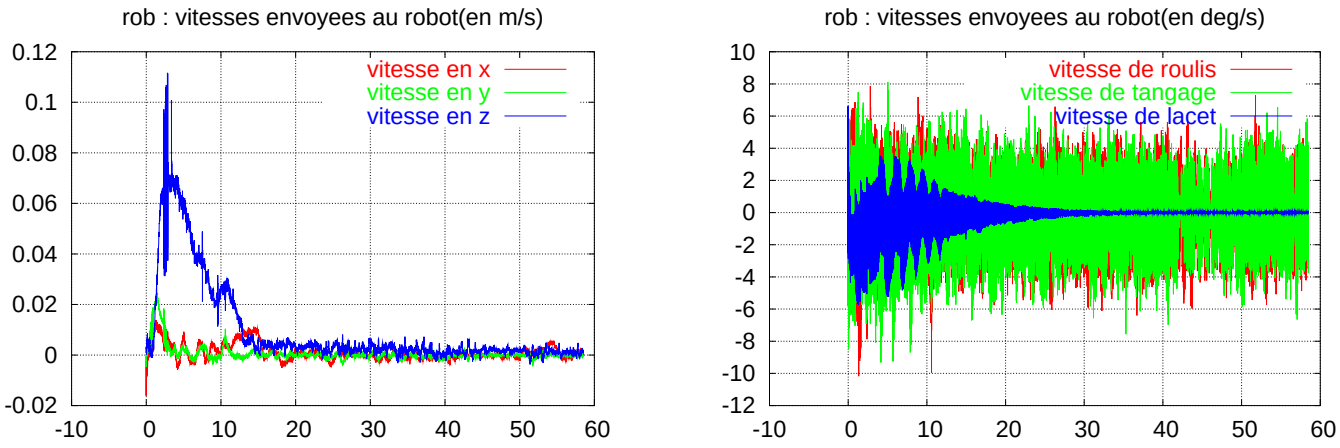
(a) (b) (c)

FIG. 17 – Asservissement visuel : (a) image initiale, (b) image finale, (c) trajectoires des points dans l'image



(a) (b)

FIG. 18 – Asservissement visuel : (a) erreur sur les informations visuelles, (b) vitesses calculées par l'asservissement visuel



(a) (b)

FIG. 19 – Vitesses envoyées au robot par le simulateur : (a) vitesses de translation, (b) vitesses de rotation

5.2 Utilisation de la projection sphérique

5.2.1 Equations de la projection sphérique

Une approche intéressante pour modéliser des informations visuelles consiste à considérer l'image obtenue avec une caméra sphérique.

Concrètement, on ne va pas utiliser une caméra sphérique, mais calculer, à partir d'une caméra classique et de son modèle de projection perspective, les coordonnées de la projection des points sur une surface sphérique.

Afin d'alléger les calculs, on considère le cas d'une sphère de rayon unité. On a (voir figure 20) :

$$\tan\alpha = \frac{x_s}{z_s} \quad \text{et} \quad \tan\alpha = \frac{x}{1} \tag{42}$$

d'où

$$x = \frac{x_s}{z_s} \tag{43}$$

Or,

$$\cos\alpha = \frac{z_s}{1} \quad \text{et} \quad \cos\alpha = \frac{1}{\|\boldsymbol{p}_p\|} \tag{44}$$

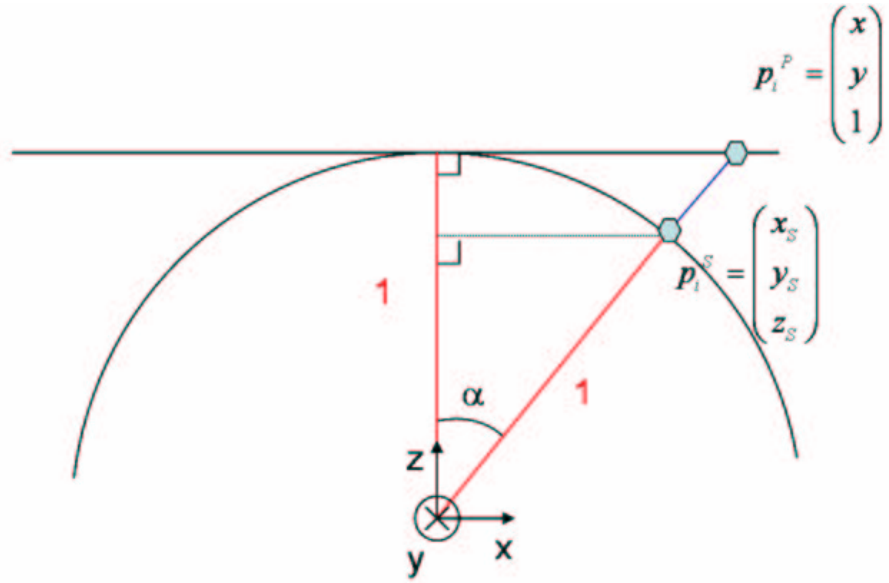


FIG. 20 – Modèle de projection sphérique

D'où

$$x_s = \frac{x}{\|\boldsymbol{p}_p\|} \tag{45}$$

De même, on a

$$y_s = \frac{y}{\|\boldsymbol{p}_p\|} \tag{46}$$

Finalement, on peut donc écrire la relation entre les coordonnées d'un point sur le plan image perspective et ses coordonnées une fois projeté sur le plan image sphérique :

$$\boldsymbol{p}_s = \frac{\boldsymbol{p}_p}{\|\boldsymbol{p}_p\|} \tag{47}$$

Si on note $\boldsymbol{P} = (X, Y, Z)$ les coordonnées 3D du point considéré (exprimées dans le repère lié à la caméra), et f la distance focale de la caméra, on a (modèle de projection perspective) :

$$\boldsymbol{p}_p = (u, v, f) = \left(\frac{fX}{Z}, \frac{fY}{Z}, f\right) \tag{48}$$

Avec (47), on en déduit alors la relation entre les coordonnées d'un point 3D et les coordonnées de sa projection sphérique :

$$\boldsymbol{p}_s = \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|} \tag{49}$$

La dynamique des coordonnées des points ainsi projetés sur une surface sphérique est donnée dans [6, 8] :

$$\dot{\boldsymbol{p}}_s = \begin{pmatrix} -\frac{\boldsymbol{\pi}_{\boldsymbol{p}_s}}{\|\boldsymbol{P}\|} & \mathrm{sk}(\boldsymbol{p}_s) \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{pmatrix} \tag{50}$$

où

- $\boldsymbol{v}$ et $\boldsymbol{\omega}$ sont les vitesses linéaires et de rotation de la caméra, exprimées dans le repère qui lui est lié ;
- $\boldsymbol{\pi}_{\boldsymbol{p}_s} = \boldsymbol{I}_3 - \boldsymbol{p}_s \boldsymbol{p}_s^T$ (projection sur le plan tangent à la sphère au point $\boldsymbol{p}_s$).

5.2.2 Moments 3D

Intuitivement, il est facile de se rendre compte que la surface correspondant à l'objet considéré, projetée sur une surface sphérique restera invariante aux mouvements de rotation de la caméra. Dans [21], cette propriété est montrée pour la surface ainsi que pour d'autres informations visuelles obtenues à partir des moments 3D de l'image sphérique. On pourrait alors envisager d'utiliser ce type d'information visuelle, invariante aux rotations, pour contrôler les mouvements de translation de la caméra. Cependant, la projection sphérique que nous utilisons est obtenue à partir de l'image d'une caméra classique, et le champ de vision est donc plus restreint que si on utilisait une véritable caméra sphérique. De ce fait les matrices d'interaction correspondantes sont mal conditionnées [21] : seule la translation selon l'axe z pourrait être correctement contrôlée.

5.2.3 Informations visuelles passives

Comme on vient de le voir, trouver des informations visuelles permettant d'obtenir un découplage adapté à notre système sous actionné n'est pas évident. Cependant, dans [13], des informations visuelles dotées de la propriété de passivité structurelle ont été déterminées. Cette notion de passivité s'apparente au découplage entre degrés de liberté de rotation et de translation. Elle a été obtenue pour des points en utilisant le modèle de projection sphérique précédemment décrit. Notons que l'on réalise bien un asservissement 2D, même si la méthode utilisée

nécessite de connaître à chaque instant l'orientation de la caméra par rapport à la cible.

Contrôle de la position Pour contrôler la position, l'information visuelle considérée est le centre de gravité des points projetés sur l'image sphérique.

On définit les coordonnées du centre de gravité des n points projetés sur la surface sphérique :

$$\boldsymbol{q} = \sum_{i=1}^{n} \boldsymbol{p}_i^s \tag{51}$$

Avec (50), on déduit alors l'expression de $\dot{\boldsymbol{q}}$:

$$\dot{\boldsymbol{q}} = -\mathrm{sk}(\boldsymbol{\omega})\boldsymbol{q} - \boldsymbol{Q}\boldsymbol{v} \tag{52}$$

où

$$\boldsymbol{Q} = \sum_{i=1}^{n} \frac{\boldsymbol{\pi}_{\boldsymbol{p}_i^s}}{\|\boldsymbol{P}_i\|} \tag{53}$$

Habituellement, les informations visuelles désirées que l'on considère en asservissement visuel sont exprimées dans le repère lié à la caméra. La particularité du choix fait dans [8, 13] est d'exprimer la consigne à atteindre $\boldsymbol{b}$ dans le repère inertiel fixe. Ainsi, à chaque itération on calculera

$$\boldsymbol{b}^*(t) = {}^f\boldsymbol{R}_{c(t)}^T \boldsymbol{b} \tag{54}$$

où ${}^f\boldsymbol{R}_{c(t)}$ représente l'orientation de la caméra dans le repère fixe.

$\boldsymbol{b}$ étant fixé dans le repère inertiel, la dynamique de $\boldsymbol{b}^*$ est contrainte par celle de la caméra :

$$\dot{\boldsymbol{b}}^*(t) = -\mathrm{sk}(\boldsymbol{\omega})\boldsymbol{b}^*(t) \tag{55}$$

On considère alors l'erreur à minimiser, $\boldsymbol{\delta}$:

$$\boldsymbol{\delta} = \boldsymbol{q} - \boldsymbol{b}^* \tag{56}$$

Et on déduit de (52), (55) et (56) la dynamique de l'erreur :

$$\dot{\boldsymbol{\delta}} = -\mathrm{sk}(\boldsymbol{\omega})\boldsymbol{\delta} - \boldsymbol{Q}\boldsymbol{v} \tag{57}$$

L'objectif de l'asservissement visuel étant de minimiser l'erreur $\boldsymbol{\delta}$, la fonction d'énergie correspondante s'écrit $S = \frac{1}{2}\|\boldsymbol{\delta}\|^2$.
Et, avec (57), on a

$$\dot{S} = -\boldsymbol{\delta}^T \boldsymbol{Q} \boldsymbol{v} \tag{58}$$

Notons que $\dot{\boldsymbol{\delta}}$ est fonction de $\boldsymbol{v}$ et $\boldsymbol{\omega}$ (pas de découplage des degrés de liberté de rotation et de translation), mais que la dérivée $\dot{S}$ de l'énergie ne dépend quant à elle que de $\boldsymbol{v}$ (propriété de passivité structurelle des informations visuelles).

Comme dans [13], on choisit alors

$$\boldsymbol{v} = k_\delta \boldsymbol{Q}^{*-1} \boldsymbol{\delta} \tag{59}$$

où on choisit un gain k_δ strictement positif.

On peut montrer que dans le cas d'un système « parfait » (qui se déplacerait excatement aux vitesses calculées par l'asservissement visuel), ce choix pour la loi de commande de $\boldsymbol{v}$ assure la stabilisation asymptotique de l'erreur $\boldsymbol{\delta}$. En effet, en remplaçant $\boldsymbol{v}$ par son expression dans (58) on obtient

$$\dot{S} = -k_\delta \boldsymbol{\delta}^T \boldsymbol{Q} \boldsymbol{Q}^{*-1} \boldsymbol{\delta} \tag{60}$$

Il est montré dans [13] que $\boldsymbol{Q}\boldsymbol{Q}^{*-1}$ est définie positive. La théorie de Lyapunov sur la stabilité des systèmes garantit alors la convergence de l'erreur $\boldsymbol{\delta}$ vers $\mathbf{0}$. En effet $\dot{S} < 0$, donc l'énergie S de l'erreur diminue dès que la vitesse $\boldsymbol{v}$ de la caméra est non nulle. On n'a pas réellement découplé les degrés de liberté de rotation et de translation du système, mais l'intérêt de la méthode proposée est qu'on a toujours $\dot{S} < 0$, quelle que soit sa vitesse de rotation $\boldsymbol{\omega}$.

On peut remarquer que la matrice $\boldsymbol{Q}$ n'est pas connue à chaque instant (elle est fonction de $\|\boldsymbol{P}_i\|$, et donc de la position 3D des points considérés) : on se contente donc d'utiliser $\boldsymbol{Q}^*$, la matrice $\boldsymbol{Q}$ correspondant à l'état final désiré.

Le centre de gravité des points projetés sur une surface sphérique a permis de contrôler la position de la caméra, et ce sans que les éventuelles rotations de la caméra n'empêchent l'énergie de l'erreur à minimiser de décroître. Pour contrôler l'angle de lacet, il est nécessaire d'utiliser une autre information visuelle.

Contrôle de l'angle de lacet On définit à présent

$${}^a\boldsymbol{q} = \sum_{i=1}^{n} a_i \boldsymbol{p}_i^s \tag{61}$$

où les a_i sont des constantes, fixées par l'orientation que l'on souhaite atteindre.

Notons $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ les vecteurs unitaires du repère lié à la caméra.

L'erreur à minimiser considérée est

$$\boldsymbol{\sigma} = {}^a\boldsymbol{q}_0 - \boldsymbol{e}_1 \tag{62}$$

où

$${}^a\boldsymbol{q}_0 = \frac{{}^a\boldsymbol{q}}{\|{}^a\boldsymbol{q}\|} \tag{63}$$

En considérant comme précédemment l'énergie de l'erreur, la loi de commande

$$\omega_z = k_\sigma \boldsymbol{e}_2^T \boldsymbol{\sigma} \tag{64}$$

permet de contrôler l'orientation en lacet de la caméra, en minimisant l'erreur $\boldsymbol{\sigma}$.

5.2.4 Mise en œuvre et résultats

Initialisations La consigne spécifiée est la même que pour l'expérience précédente (paragraphe 5.1.2) : placer la caméra à 50 cm au dessus de la cible, celle-ci étant centrée et orientée dans le même sens que la caméra. A partir de cet état final souhaité et du modèle CAO de la cible, on calcule les coordonnées des points image de la cible dans l'image perspective (48), puis leurs coordonnées sphériques (47). On en déduit

$$\boldsymbol{b}^*(t_{final}) = \sum_{i=1}^{4} \boldsymbol{p}_i^s(t_{final}) \tag{65}$$

On obtient avec nos spécifications $\boldsymbol{b}^*(t_{final}) = (0, 0, 3.942)$.

Il faut alors calculer le vecteur constant $\boldsymbol{b}$, exprimé dans un repère fixe :

$$\boldsymbol{b} = {}^f\boldsymbol{R}_{c(t_{final})} \boldsymbol{b}^*(t_{final}) \tag{66}$$

Pour contrôler l'angle de lacet, on a choisi

$$a_1 = -1, \quad a_2 = 1, \quad a_3 = 1, \quad a_4 = -1 \tag{67}$$

On peut noter que la matrice $\boldsymbol{Q}^{*-1}$ a une forme diagonale qui permet aux erreurs sur les trois composantes de $\boldsymbol{\delta}$ de décroître à une vitesse du même ordre.

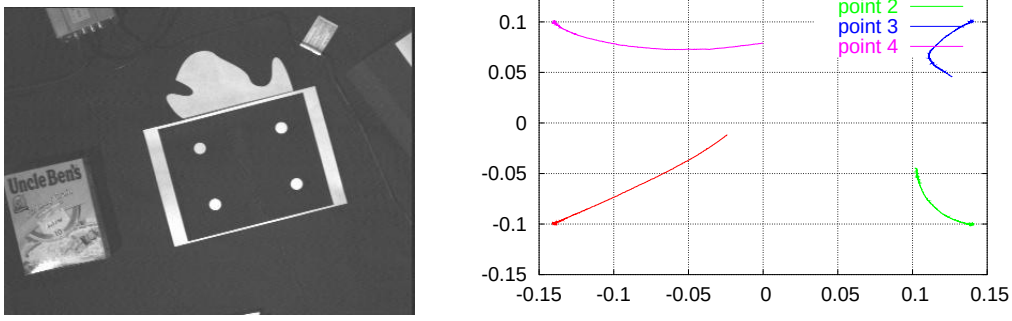
$$\boldsymbol{Q}^{*-1} = \begin{pmatrix} 0.13 & 0 & 0 \\ 0 & 0.13 & 0 \\ 0 & 0 & 4.41 \end{pmatrix} \tag{68}$$

Si on utilise simplement la loi de commande $\boldsymbol{v} = k_\delta \boldsymbol{\delta}$ au lieu de (59), la décroissance selon l'axe Z est très lente par rapport aux autres.

Le problème qui se pose est alors d'une part d'estimer ${}^f\boldsymbol{R}_{c(t_{final})}$ pour pouvoir calculer $\boldsymbol{b}$ avec (66) et ensuite, à chaque itération, d'estimer la matrice ${}^f\boldsymbol{R}_{c(t)}$ pour en déduire $\boldsymbol{b}^*(t)$ avec (54).

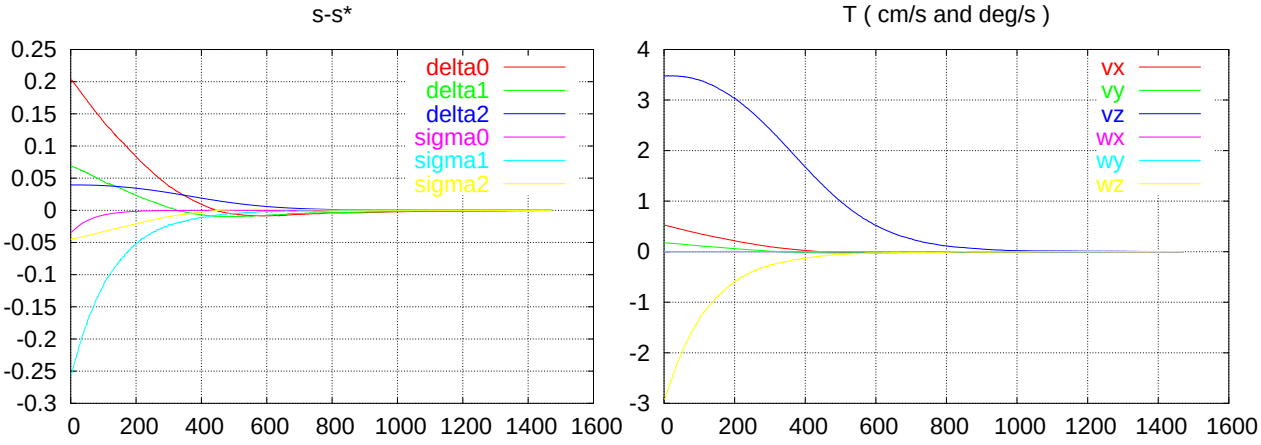
Utilisation de l'odométrie Pour connaître l'orientation de la caméra dans un repère fixe, on a dans un premier temps utilisé l'odométrie du robot uniquement. Pour cela, la mesure de ${}^f\boldsymbol{R}_{c(t_{final})}$ est faite dans une première phase « hors ligne », en amenant le robot à la position finale, et en relevant simplement l'orientation de la caméra dans le repère fixe, donnée par l'odométrie du robot. On peut alors initialiser $\boldsymbol{b}$ avec (66). Dans une deuxième phase, on place le robot à une position initiale inconnue, et à chaque itération, il suffit alors d'utiliser l'odométrie pour connaître le nouveau ${}^f\boldsymbol{R}_{c(t)}$ et en déduire $\boldsymbol{b}^*(t)$ avec (54).

On a testé cette méthode tout d'abord sans utiliser le simulateur de l'hélicoptère, en forçant les vitesses de roulis et de tangage à zéro. On voit sur la figure 21.b que l'on a bien atteint la position désirée dans l'image perspective (coordonnées des points image : $(-0.14, -0.1), (0.14, -0.1)(0.14, 0.1), (-0.14, 0.1)$). Les figures 22.a et 22.b montrent la décroissance exponentielle des erreurs $\boldsymbol{\delta}$ et $\boldsymbol{\sigma}$ à minimiser, et des vitesses commandées v_x, v_y, v_z et ω_z.



(a) (b)

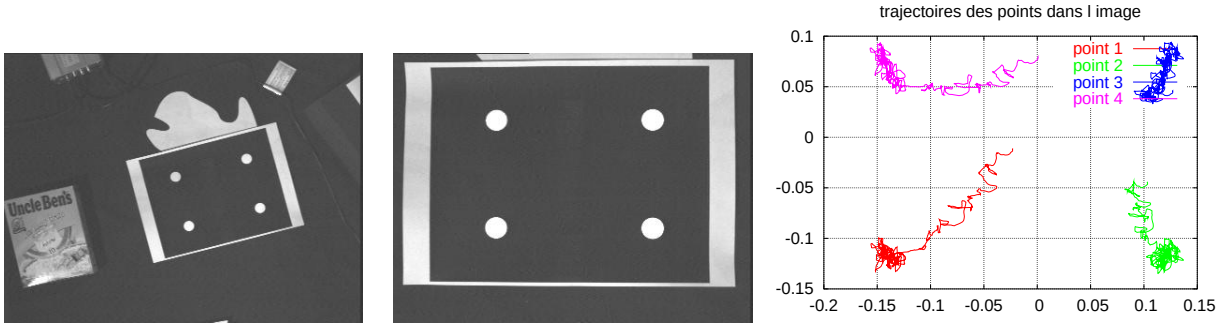
FIG. 21 – (a) image initiale, (b) trajectoire des points dans l'image perspective



(a) (b)

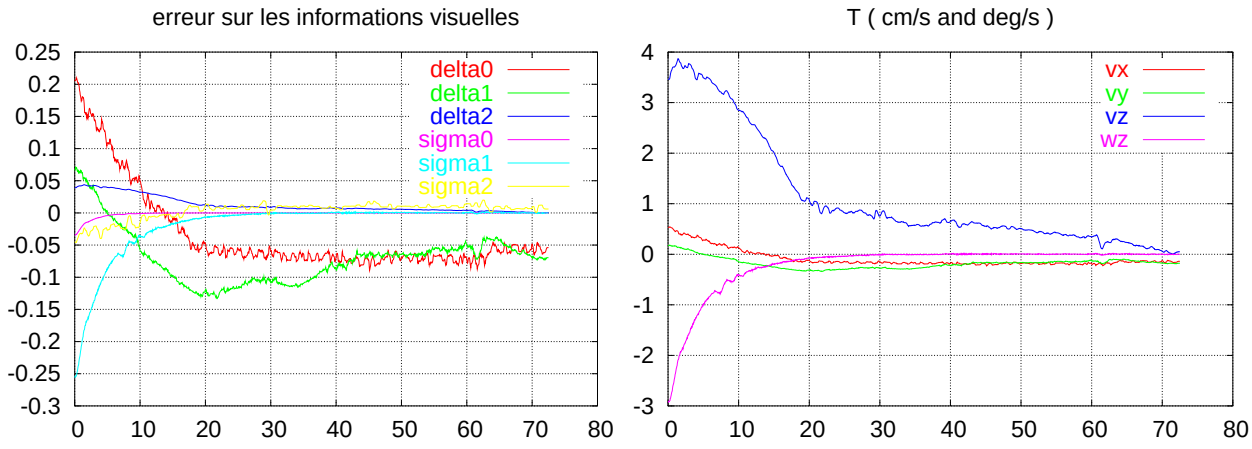
FIG. 22 – (a) composantes des erreurs $\boldsymbol{\delta}$ et $\boldsymbol{\sigma}$ à minimiser, (b) vitesses calculées par la loi de commande et envoyées au robot

Ayant de bons résultats avec un robot « parfait », on a alors testé cette méthode sur le robot contraint par la dynamique de l'hélicoptère. On observe une décroissance quasi exponentielle des vitesses (figure 24.b) et des erreurs à minimiser (figure 24.a). Cependant, on remarque un léger « biais » au niveau de l'erreur sur δ_0 et δ_1. On voit également ce phénomène sur la figure (figure 23.c) : les points n'ont pas convergé exactement à la position souhaitée. Cette erreur est due, comme dans les résultats obtenus avec les moments 2D (paragraphe 5.1.2) aux mouvements non contrôlés de roulis et tangage de l'hélicoptère (bien visibles sur la figure 25.b).



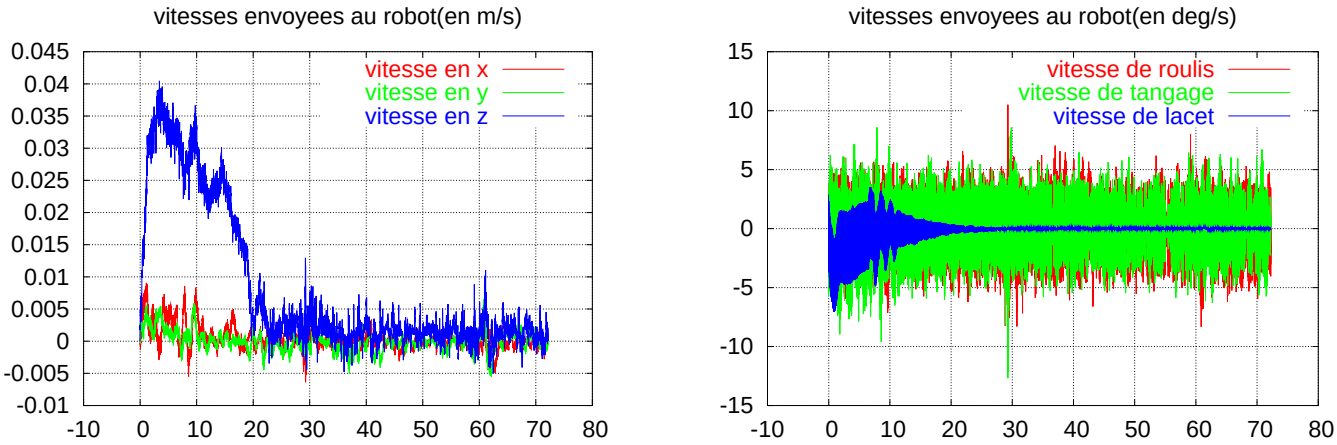
(a) (b) (c)

FIG. 23 – (a) image initiale, (b) image finale, (c) trajectoires des points dans l'image perspective



(a) (b)

FIG. 24 – (a) composantes des erreurs δ et σ à minimiser, (b) vitesses calculées par la loi de commande et envoyées au robot

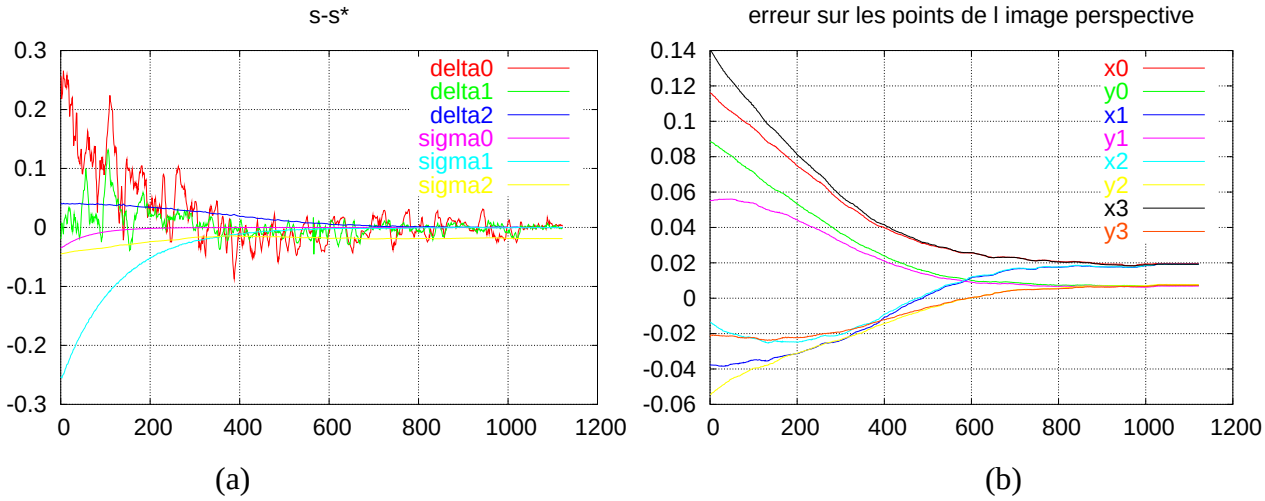


(a) (b)

FIG. 25 – Vitesses envoyées au robot par le simulateur : (a) vitesses de translation, (b) vitesses de rotation

Utilisation du calcul de pose Pour une utilisation réelle sur l'hélicoptère, aucune mesure d'odométrie n'est disponible. Cependant, la méthode développée dans la partie 4 permet d'estimer la position et l'orientation de l'hélicoptère, en utilisant uniquement les données de la vision.

On a alors testé l'asservissement visuel en mesurant l'orientation de la caméra par calcul de pose et filtrage de Kalman (voir partie 4), tout d'abord avec un robot « parfait » (sans la dynamique de l'hélicoptère) dont on commande quatre degrés de liberté, les vitesses de roulis et tangage étant fixées à zéro. Le repère fixe considéré pour le calcul de $\boldsymbol{b}$ et $\boldsymbol{b}^*(t)$ est le repère lié à l'objet. A chaque itération, la sortie du filtre de Kalman nous donne ${}^{c(t)}\boldsymbol{R}_f$ d'où on déduit ${}^{f}\boldsymbol{R}_{c(t)}$ par transposition. Pour l'initialisation, on considère l'orientation donnée par les spécifications : ${}^{f}\boldsymbol{R}_{c(t_final)} = \boldsymbol{I}_3$.



(a) (b)

FIG. 26 – (a) composantes des erreurs $\boldsymbol{\delta}$ et $\boldsymbol{\sigma}$ à minimiser, (b) erreur sur les coordonnées des points dans l'image perspective

On constate qu'en procédant de cette manière, les points ne convergent pas vers la position souhaitée : l'asservissement visuel converge (figure 26.a), mais l'écart entre les positions souhaitées et réelles dans l'image n'est pas nul (figure 26.b). En effet, le vecteur $\boldsymbol{b}$ est ici calculé en utilisant l'orientation finale « parfaite » que l'on souhaite obtenir (la matrice identité $\boldsymbol{I}_3$). Or, comme on ne commande pas les rotations de roulis et de tangage, si ces angles ne sont pas parfaitement nuls au départ, la véritable matrice de rotation obtenue à la convergence ne pourra jamais être l'identité. A la convergence, le vecteur $\boldsymbol{b}$ sera bien atteint, mais

il ne correspondra pas au vecteur $\boldsymbol{b}^*(t_{final})$ que l'on souhaitait atteindre.

Pour remédier à ce problème, il conviendrait de mesurer l'orientation réelle de la caméra par rapport à l'objet dans la position finale souhaitée (et atteignable), et d'utiliser cette mesure pour spécifier le vecteur $\boldsymbol{b}$ dans le repère fixe. On peut noter que seule la connaissance de l'orientation de l'axe e_3 de la caméra dans le repère objet est en fait nécessaire dans notre cas, puisque $\boldsymbol{b}^*(t_{final})$ n'a qu'une composante selon cet axe.

Cependant, si une telle mesure est réalisable avec le robot cartésien, elle ne pourra pas être réalisée avec l'hélicoptère. En effet, on ne peut pas connaître une valeur précise pour les angles de roulis et tangage de l'hélicoptère à l'état final, puisqu' ils varient sans cesse du fait de la dynamique de l'hélicoptère.

Le principal problème lié à la méthode proposée est l'estimation de l'orientaion de la caméra dans un repère fixe. Pour s'affranchir d'un tel problème, il faudrait utiliser des informations visuelles « purement 2D ».

5.2.5 Approximation de la passivité

La méthode précédente, bien qu'ayant l'avantage d'être dotée de la propriété de passivité, ne donne pas de bons résultats compte tenu du fait qu'elle utilise la mesure de l'orientation de la caméra dans sa loi de commande.

On a alors utilisé une méthode purement 2D, qui n'utilise donc que des informations de l'image, mais pour laquelle la propriété de passivité n'est qu'approchée.

Les mêmes informations visuelles que dans la méthode précédente (paragraphe 5.2.3) sont utilisées, mais on exprime cette fois l'information visuelle désirée dans le repère lié à la caméra. Ainsi, au lieu de calculer le nouveau $\boldsymbol{b}^*(t)$ à chaque itération (avec (54)), on conserve $\boldsymbol{b}^*(t) = \boldsymbol{b}^*(t_{final})$ tout au long de l'asservissement.

On a donc maintenant, au lieu de (55)

$$\dot{\boldsymbol{b}}^* = 0 \tag{69}$$

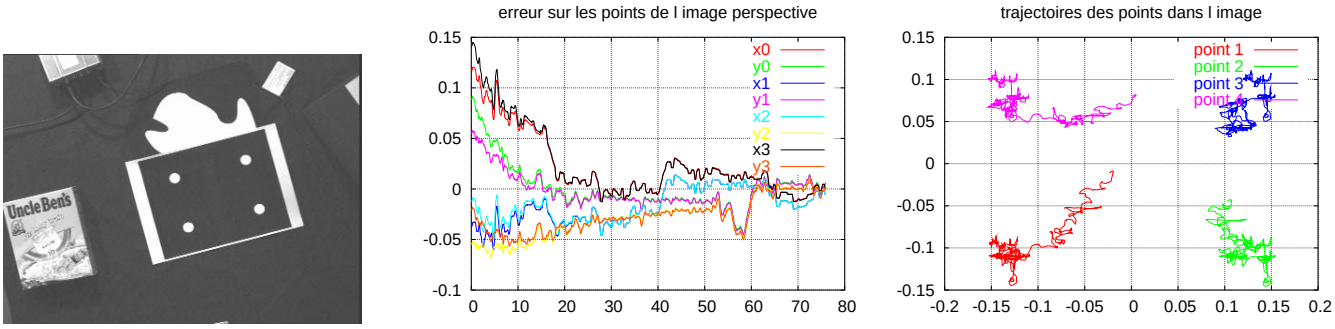
et la dynamique de l'erreur devient

$$\dot{\boldsymbol{\delta}} = -\mathrm{sk}(\boldsymbol{\omega})\boldsymbol{\delta} - \boldsymbol{Q}\boldsymbol{v} - \mathrm{sk}(\boldsymbol{\omega})\boldsymbol{b}^* \tag{70}$$

On remarque que le terme $\mathrm{sk}(\boldsymbol{\omega})\boldsymbol{b}^*$ vient s'ajouter au résultat que l'on avait obtenu en (57). Ce terme agit comme une perturbation et détruit la propriété de passivité des informations visuelles considérées.

Bien que le raisonnement portant sur la décroisssance de l'énergie de l'erreur ne puisse donc pas s'appliquer ici, on a négligé ce terme, et utilisé les même lois de commande que précédement : (59) pour contrôler les translations et (64) pour contrôler l'angle de lacet.

Les consignes spécifiées sont les mêmes que pour les expériences précédentes (paragraphe 5.2.4). On voit sur les figures 27.a, 27.b et 27.c que l'on atteint bien la position désirée dans l'image. Les écarts ne sont dus qu'aux mouvements de rotation de roulis et tangage de l'hélicoptère, mouvement permanent et non contrôlé. La figure 28.b montre une décroissance quasi exponentielle des vitesses calculées par la loi de commande. Ce profil de décroissance ne se retrouve cependant pas au niveau de l'erreur sur les informations visuelles (figure 28.a).

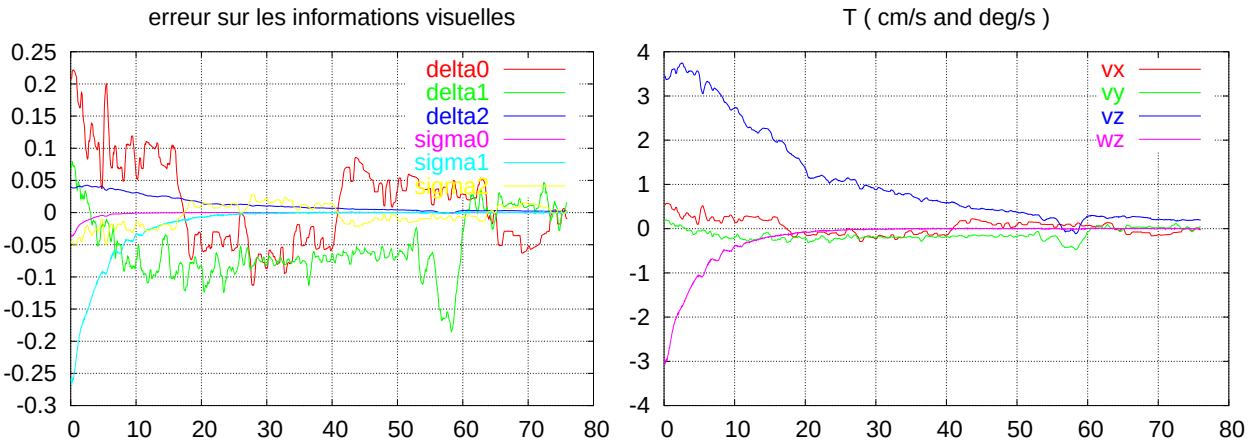


(a) (b) (c)

FIG. 27 – (a) image initiale, (b) erreur sur les coordonnées des points de l'image perspective, (c) trajectoires des points dans l'image perspective

5.3 Conclusion et perspectives

La première méthode expérimentée pour l'asservissement visuel de l'hélicoptère utilise les moments 2D de l'image et permet un découplage partiel des degrés de liberté. Elle nous a conduit à négliger les termes de roulis et de tangage dans la loi de commande, mais les résultats obtenus restent corrects : bonne décroissance des vitesses et de l'erreur $\boldsymbol{s} - \boldsymbol{s}^*$, convergence globale vers la position souhaitée. La seconde méthode envisagée se base sur des informations visuelles provenant de la projection de l'image perspective sur une surface sphérique. Cette approche permet de modéliser des informations visuelles passives, si on introduit une estimation de l'orientation de la caméra dans la loi de commande. Mais la nécessité



(a) (b)

FIG. 28 – (a) composantes des erreurs $\boldsymbol{\delta}$ et $\boldsymbol{\sigma}$ à minimiser, (b) vitesses calculées par la loi de commande et envoyées au robot

d'estimer cette matrice de rotation pose problème lorsqu'on se contraint à n'utiliser que les informations fournies par la vision, ce qui sera le cas avec l'hélicoptère. Nous avons donc finalement réalisé un asservissement visuel dérivé de la précédente méthode, dont la loi de commande n'utilise plus que des informations 2D, mais dont la propriété de passivité n'est plus qu'approchée.
Les méthodes utilisées ne semblent pas concluantes pour asservir visuellement l'hélicoptère : les rotations de roulis et de tangage, non contrôlées, perturbent l'évolution du système. Déterminer des informations visuelles découplées des angles de roulis et de tangage, ou ayant la propriété de passivité par rapport à ces degrés de liberté serait intéressant pour mieux commander l'hélicoptère.

Conclusion

Après avoir implanté le simulateur d'hélicoptère développé par le CEA/LIST sur le robot cartésien de l'IRISA, nous avons pu y valider une méthode qui pourra permettre à l'hélicoptère de se localiser par vision dans son environnement. La méthode proposée suppose que cet environnement est constitué de cibles connues, mais elle pourrait par la suite considérer des cibles inconnues, ou de mauvaises conditions de détection.

Pour l'asservissement visuel de l'hélicoptère, la stratégie la plus adaptée consisterait à déterminer des informations visuelles découplées des degrés de liberté non commandés de l'hélicoptère. Hélas, trouver de telles informations visuelles s'avère difficile. Une propriété approchant le découplage mais moins restrictive pourrait suffire : la passivité. Si de telles informations ont pu être déterminées, leur utilisation sur un système se basant uniquement sur la vision n'a pas donné de résultats très intéressants. Le principal problème de la méthode proposée réside dans le fait que sa loi de commande recquiert une estimation de l'orientation de la caméra. La modélisation d'informations visuelles purement 2D et ayant de bonnes propriétés vis à vis de la dynamique de l'hélicoptère permettra certainement par la suite de réaliser un meilleur asservissement visuel, bien adapté à la dynamique de l'hélicoptère.

A Présentation du laboratoire

A.1 L'IRISA

Depuis 1970, l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires), situé sur le verdoyant campus de Beaulieu à Rennes, n'a cessé de se développer. Il est aujourd'hui un laboratoire de recherche publique regroupant environ 500 personnes dont plus de 170 chercheurs ou enseignants chercheurs, 150 doctorants, 80 ingénieurs, techniciens, administratifs et de nombreux collaborateurs contractuels ou invités internationaux pour des séjours de plus courte durée. L'INRIA (Institut National de Recherche en Informatique et en Automatique), le CNRS (Centre National de la Recherche Scientifique), l'Université de Rennes 1 et l'INSA (Institut National des Sciences Appliquées) de Rennes sont les partenaires de cette unité mixte de recherche.

L'IRISA est organisé en une trentaine de projets, autour de grands thèmes scientifiques : les réseaux et systèmes — le logiciel — les interactions homme-machine et le traitement de données et des connaissances — la bio-informatique — le traitement d'images et la réalité virtuelle — la modélisation, la simulation et l'optimisation de systèmes complexes. Ces thèmes génériques se déclinent dans de nombreux domaines d'application, tels que les télécommunications, le multimédia et les technologies avancées pour la santé et l'environnement.

L'IRISA s'implique également fortement dans l'organisation de manifestations (congrès, séminaires, journées de veille technologique, ...), et mène de nombreuses actions industrielles et internationales.

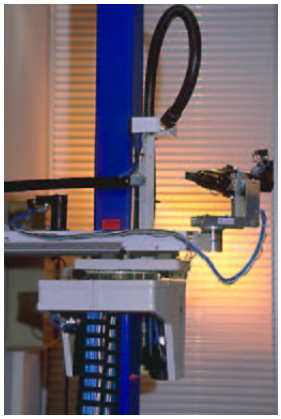
A.2 L'équipe Lagadic

L'équipe Lagadic (« lagadic » signifie « petit œil » en breton) qui m'a accueillie pendant mon stage a vu le jour début 2004, après de nombreuses années passées au sein du projet ViSTA (Vision Spatio-Temporelle Active). Les axes de recherche de l'équipe Lagadic concernent la vision robotique et l'asservissement visuel. Son objectif est plus précisément d'élaborer des stratégies de perception et d'action à partir d'images pour des applications en robotique, vision par ordinateur, réalité virtuelle et réalité augmentée. Sous la responsablité de François Chaumette, de nombreuses collaborations sont en cours avec des industriels, des universités étrangères, dans le cadre des projets RobEA (Robotique et Entités Artificielles) ou d'autres projets nationaux. Actuellement l'équipe compte onze personnes, dont deux chercheurs, un ingénieur, quatre doctorants, un post-doctorant,

deux ingénieurs contractuels et une assistante de projet.

Des plates-formes de test permettent à l'équipe de tester ses développements. Un robot cartésien à six axes équipé d'une caméra (figure 29.a), un robot cylindrique équipé d'une caméra pan-tilt (figure 29.b), un Cycab (petit véhicule électrique, figure 29.c) équipé lui aussi d'une caméra sont utilisés régulièrement depuis de nombreuses années. Des logiciels ont été développés pour chaque plate-forme et permettent de les utiliser simplement, en n'agissant que haut niveau. Un robot médical anthropomorphe (figure 29.d) est en cours de mise au point et devrait permettre à terme la reconstruction d'images 3D à partir d'un dispositif échographique 2D fixé sur le bras du robot.

(a)



(b)

(c)

(d)

FIG. 29 – Les plates-formes : (a) le robot cartésien, (b) le robot cylindrique, (c) le Cycab, (d) le robot anthropomorphe

B Matériel et logiciel utilisés

Le robot cartésien de l'IRISA m'a permis de tester les algorithmes développés de manière plus réaliste qu'en simulation, même si des tests réels seront réalisés par la suite sur le X4-flyer. J'ai également utilisé la micro-caméra HF dont sera équipé l'hélicoptère.

B.1 Le robot cartésien

Ce robot (figure 29.a), fabriqué par la société Afma Robots basée à Tours a été installé à l'IRISA en 1990. Il s'agit d'un robot cartésien à six degrés de liberté équipé d'une caméra CCD Sony monochrome (focale de 12 mm) embarquée sur son effecteur. Il est piloté par un PC Linux équipé d'une carte de numérisation ICcomp de la société Imaging Technology ; le PC communique à l'aide d'un coupleur de bus avec la commande numérique (AICO/ITMI fonctionnant sur le bus VME).

Une interface (figure 30) permet de commander le robot en spécifiant la position désirée dans les repères articulaire, fixe ou le repère lié à la caméra.

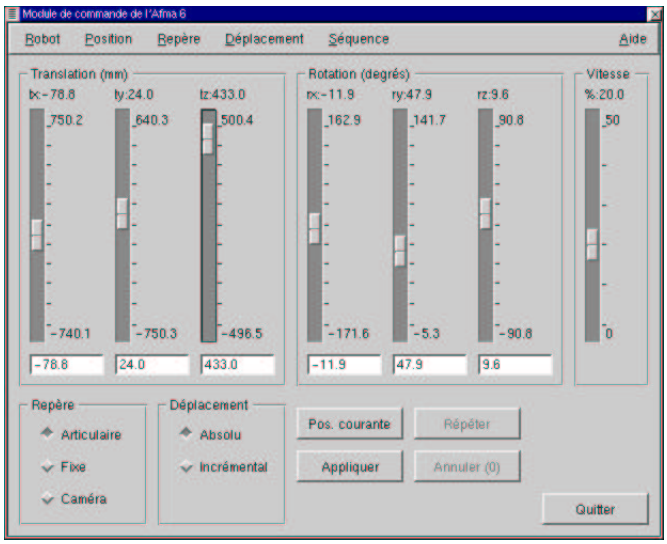


FIG. 30 – Interface pour la commande du robot cartésien

B.2 Le logiciel ViSP (Visual Servoing Platform)

ViSP est une plate-forme logicielle codée en C++ et dédiée à l'asservissement visuel ; elle contient la plupart des fonctionnalités de base nécessaires. Par exemple, citons les lois de commandes 2D, 2D 1/2 et 3D, une bibliothèque de traitement d'images qui permet le suivi de primitives visuelles (point, segment, ellipse, spline, etc.) à la cadence vidéo, une bibliothèque de calcul de pose, des fonctions permettant de manipuler les matrices homogènes, de communiquer avec les robots, d'acquérir des images, etc.

ViSP est utilisé très régulièrement par les membres de l'équipe Lagadic, et continue à évoluer.

B.3 La micro caméra

L'hélicoptère ne pouvant pas effectuer le traitement d'image lui-même (ce qui nécessiterait d'embarquer des sytèmes de calcul lourds et encombrants), les images acquises par sa caméra embarquée seront transmises par liaison HF (haute fréquence) à un poste de pilotage qui se chargera de traiter les images et de calculer les commandes (elles-mêmes retransmises à l'hélicoptère).

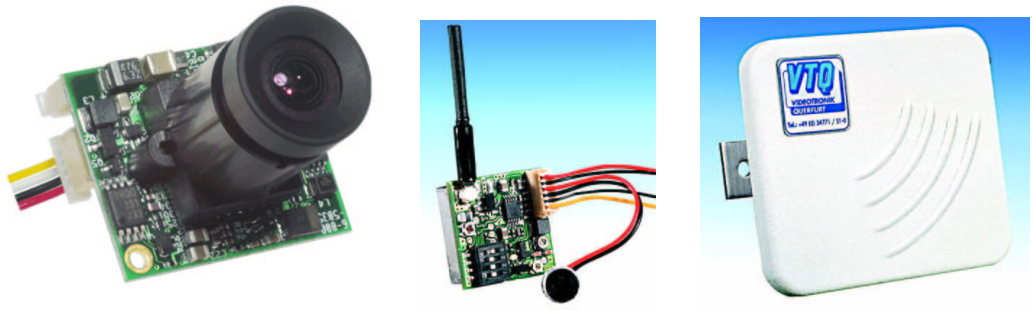
Ainsi, les contraintes de taille et de poids ont conduit les personnes en charge du choix de la caméra à choisir le modèle VTQ-54 (figure 31.a), commercialisé par la société Opto Vision de Toulouse. Cette caméra couleur pèse environ 30 grammes, et mesure 26 x 22 mm. Il s'agit d'une caméra CCD entrelacée, 542 x 586 pixels. Elle s'alimente en 12 Volts continu, et est équipée d'une focale de 3.7 mm. Pour pouvoir mieux s'adapter aux scènes que l'on souhaite visualiser, un objectif de 8 mm a également été acquis. Par ailleurs, un micro émetteur HF audio-vidéo (18 mm x 18 mm, 8 grammes, figure 31.b) permettant l'émission du signal vidéo sera également installé sur l'hélicoptère.

L'équipe Lagadic s'est dotée de ce matériel, ainsi que d'une antenne (figure 31.c) et d'un récepteur HF, afin de pouvoir réaliser des tests dans les conditions les plus approchantes de la réalité, bien que ne disposant pas de l'hélicoptère lui-même.

Pour ma part, j'ai pu tester en partie les algorithmes de localisation sur les images transmises par cette caméra. En effet, du fait de sa livraison tardive, la caméra n'a pas encore pu être installée sur le robot cartésien, et je me suis donc contentée d'acquérir des séquences en bougeant « à la main » la caméra.

J'ai pu observer les effets de distorsion lorsqu'on s'approche trop de la cible (bien visible sur la partie gauche de la figure 32.a) ainsi que les effets parasites

liés à la transmission HF du signal vidéo (figure 32.b). Ces effets parasites n'apparaissent cependant que rarement et n'ont pas perturbé l'algorithme de localisation.



(a) (b) (c)

FIG. 31 – Matériel pour l'hélicoptère : (a) la caméra, (b) l'émetteur, (c) l'antenne de réception

(a) (b)

FIG. 32 – Image transmise par la caméra : (a) distorsion, (b) bruit lié à la transmission

B.4 Programmation multi-threads

Afin de pouvoir faire fonctionner simultanément le simulateur (cadencé à 10 ms) et le traitement d'image (à 40 ms), nous avons choisi de mettre en place une stratégie de programmation multi-threads (processus légers). Par rapport à l'utilisation de plusieurs process, le multi-thread présente essentiellement l'avantage de

fournir des outils simples d'utilisation permettant aux différents threads de communiquer entre eux. Dans notre application, nous nous sommes donc contentés d'utiliser des mutex, permettant de gérer l'exclusion mutuelle des threads lors de l'accès aux zones de mémoires partagées (figure 33). Ces zones contiennent les informations devant transiter du traitement d'image au simulateur : mesure de la pose pour la localisation, vitesses de l'hélicoptère pour l'asservissement visuel.

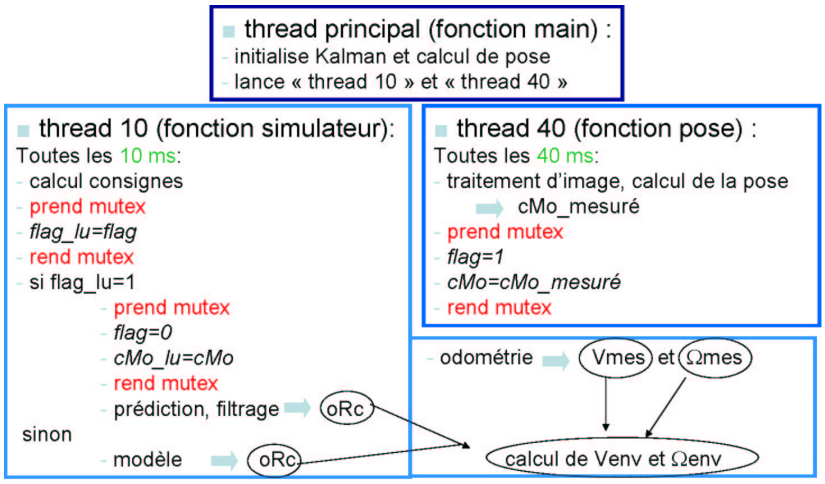


FIG. 33 – Schéma de fonctionnement du programme multi-thread pour la localisation

On peut noter que les échéances de 40 ms et de 10 ms ne sont pas toujours forcément respectées (figure 33) . On s'est en effet contenté d'utiliser un système Linux « classique », non temps réel. Cependant, les légers dépassements n'ont pas posé problème dans notre application.

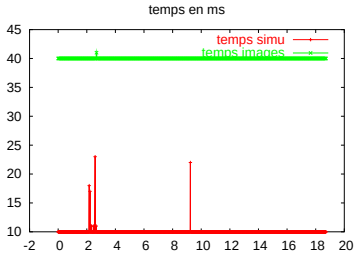


FIG. 34 – Cadence des boucles de simulation et de traitement d'image

C Equation d'état pour la partie rotation

On cherche, connaissant l'orientation $\boldsymbol{\theta}(t)$ et la vitesse de rotation $\boldsymbol{\omega}(t)$, à connaître la nouvelle orientation $\boldsymbol{\theta}(t+dt)$ de la caméra, dont le mouvement est à vitesse de rotation constante. On note $\mathcal{R}_{\mathcal{O}}$ le repère lié à l'objet, et $\mathcal{R}_{\mathcal{C}(t)}$ le repère lié à la caméra à l'instant t. ${}^{\mathcal{C}(t)}\mathbf{R}_{\mathcal{O}}$ est la matrice de rotation du repère caméra vers le repère objet à l'instant t.

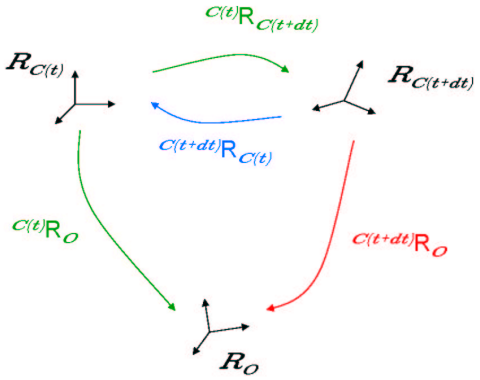


FIG. 35 – Repère caméra en mouvement, repère objet fixe, et matrices de rotations pour passer de l'un à l'autre des repères

On a :

$$\boldsymbol{R}(\boldsymbol{\theta}(t)) = {}^{\mathcal{C}(t)}\mathbf{R}_{\mathcal{O}} \tag{71}$$

où $\boldsymbol{R}(\boldsymbol{\theta}(t))$ est la formule de Rodrigues appliquée au vecteur de rotation $\boldsymbol{\theta}(t)$. On aura alors à l'instant $t+dt$:

$$\boldsymbol{R}(\boldsymbol{\theta}(t+dt)) = {}^{\mathcal{C}(t+dt)}\mathbf{R}_{\mathcal{O}} \tag{72}$$

Or,

$$\boldsymbol{R}(\boldsymbol{\omega}(t)dt) = {}^{\mathcal{C}(t)}\mathbf{R}_{\mathcal{C}(t+dt)} \tag{73}$$

D'où (figure 35) :

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{\theta}(t+dt)) &= {}^{\mathcal{C}(t+dt)}\mathbf{R}_{\mathcal{C}(t)}\,{}^{\mathcal{C}(t)}\mathbf{R}_{\mathcal{O}} \\ &= \boldsymbol{R}^T(\boldsymbol{\omega}(t)dt)\boldsymbol{R}(\boldsymbol{\theta}(t)) \end{aligned} \tag{74}$$

Il suffit alors de passer de la forme matricielle à la forme vecteur de rotation pour obtenir $\boldsymbol{\theta}(t+dt)$.
A partir d'une matrice de rotation $\mathbf{R}$ telle que

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{75}$$

on obtient le vecteur de rotation $\boldsymbol{\theta}$ par ([14, 2]) :

$$\boldsymbol{\theta} = \frac{1}{2\mathrm{sinc}\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} \qquad \text{si} \qquad \begin{cases} 0 \leq \theta < \pi \\ \pi < \theta < 2\pi \end{cases} \tag{76}$$

où $\theta = \arccos((r_{11} + r_{22} + r_{33} - 1)/2)$ et où la fonction $\mathrm{sinc}\theta$ (sinus cardinal), définie par $\sin\theta = \theta\mathrm{sinc}\theta$, est C^∞ et s'annule pour $\theta = (2n+1)\pi$ avec $n \in \mathbb{Z}$. Notons Φ la procédure permettant de passer de la matrice de rotation au vecteur de rotation, et rappelons que $\boldsymbol{R}$ (définie par l'équation (4)) permet la transformation inverse.
On a finalement, à partir de l'équation (74) :

$$\boldsymbol{\theta}(t+dt) = \Phi(\boldsymbol{R}^T(\boldsymbol{\omega}(t)dt)\boldsymbol{R}(\boldsymbol{\theta}(t))) \tag{77}$$

D Modélisation du bruit d'état

D.1 Cas d'un modèle à vitesse constante

Soit $y(t)$ la position d'un objet se déplaçant à vitesse constante, et $\dot{y}(t+dt)$ sa vitesse. Les équations d'état du système sont alors :

$$\left\{ \begin{array}{lcl} y(t+dt) & = & y(t) + \dot{y}(t)dt \\ \dot{y}(t+dt) & = & \dot{y}(t) \end{array} \right. \tag{78}$$

Supposons à présent que l'objet en question ne se déplace pas exactement à vitesse constante. On peut représenter les erreurs sur la position et sur la vitesse par les bruits w_1 et w_2.

Les équations d'état s'écrivent alors :

$$\left\{ \begin{array}{lcl} y(t+dt) & = & y(t) + \dot{y}(t)dt + w_1 \\ \dot{y}(t+dt) & = & \dot{y}(t) + w_2 \end{array} \right. \tag{79}$$

Par ailleurs, les développements au second ordre de $y(t+dt)$ et de $\dot{y}(t+dt)$ donnent :

$$\left\{ \begin{array}{lcl} y(t+dt) & = & y(t) + \dot{y}(t)dt + \ddot{y}(t)\frac{dt^2}{2} \\ \dot{y}(t+dt) & = & \dot{y}(t) + \ddot{y}(t)dt \end{array} \right. \tag{80}$$

Par identification entre les équations (79) et (80), on en déduit l'expression des bruits w_1 et w_2 :

$$\left\{ \begin{array}{lcl} w_1 & = & \ddot{y}(t)\frac{dt^2}{2} \\ w_2 & = & \ddot{y}(t)dt \end{array} \right. \tag{81}$$

Et on obtient pour les coefficients de corrélation entre w_1 et w_2 :

$$\left\{ \begin{array}{lcccl} E[w_1^2] & = & E[\ddot{y}(t)^2]\frac{dt^4}{4} & = & \sigma_r^2\frac{dt^4}{4} \\ E[w_2^2] & = & E[\ddot{y}(t)^2]dt^2 & = & \sigma_r^2 dt^2 \\ E[w_1 w_2] & = & E[\ddot{y}(t)^2]\frac{dt^3}{2} & = & \sigma_r^2\frac{dt^3}{2} \end{array} \right. \tag{82}$$

D.2 Cas d'un modèle à accélération constante

La démarche est la même que pour le cas à vitesse constante, en allant à l'ordre trois dans les développements.

Dans le cas général d'un modèle à accélération constante, on a :

$$\left\{\begin{array}{lcl} y(t+dt) & = & y(t)+\dot{y}(t)dt+\ddot{y}(t)\frac{dt^2}{2}+w_1 \\ \dot{y}(t+dt) & = & \dot{y}(t)+\ddot{y}(t)dt+w_2 \\ \ddot{y}(t+dt) & = & \ddot{y}(t)+w_3 \end{array}\right. \tag{83}$$

D'autre part, un développement au troisième ordre donne :

$$\left\{\begin{array}{lcl} y(t+dt) & = & y(t)+\dot{y}(t)dt+\ddot{y}(t)\frac{dt^2}{2}+\dddot{y}(t)\frac{dt^3}{6} \\ \dot{y}(t+dt) & = & \dot{y}(t)+\ddot{y}(t)dt+\dddot{y}(t)\frac{dt^2}{2} \\ \ddot{y}(t+dt) & = & \ddot{y}(t)+\dddot{y}(t)dt \end{array}\right. \tag{84}$$

Par identification entre (83) et (84), on en déduit l'expressions des bruits w_1, w_2 et w_3 :

$$\left\{\begin{array}{lcl} w_1 & = & \dddot{y}(t)\frac{dt^3}{6} \\ w_2 & = & \dddot{y}(t)\frac{dt^2}{2} \\ w_3 & = & \dddot{y}(t)dt \end{array}\right. \tag{85}$$

On peut alors calculer les coefficients de corrélation entre les bruits w_1, w_2, et w_3 :

$$\left\{\begin{array}{lcll} E[w_1^2] & = & E[\dddot{y}(t)^3]\frac{dt^6}{36} & = \sigma_t^2\frac{dt^6}{36} \\ E[w_2^2] & = & E[\dddot{y}(t)^3]\frac{dt^4}{4} & = \sigma_t^2\frac{dt^4}{4} \\ E[w_1^2] & = & E[\dddot{y}(t)^3]dt^2 & = \sigma_t^2 dt^2 \\ E[w_1 w_2] & = & E[\dddot{y}(t)^3]\frac{dt^5}{12} & = \sigma_t^2\frac{dt^5}{12} \\ E[w_1 w_3] & = & E[\dddot{y}(t)^3]\frac{dt^4}{6} & = \sigma_t^2\frac{dt^4}{6} \\ E[w_2 w_3] & = & E[\dddot{y}(t)^3]\frac{dt^3}{2} & = \sigma_t^2\frac{dt^3}{2} \end{array}\right. \tag{86}$$

E Calcul des termes de A

On a

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R}(-\boldsymbol{\omega}(t)dt) & \mathbf{0}_3 & -\boldsymbol{S}(-\boldsymbol{\omega}(t)dt)dt & \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\omega}} & -\boldsymbol{T}(-\boldsymbol{\omega}(t)dt)\frac{dt^2}{2} \\ \mathbf{0}_3 & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\theta}} & \mathbf{0}_3 & \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\omega}} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 dt \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 \end{pmatrix} \quad (87)$$

E.1 Calcul exact

$\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\omega}}$ correspond à la dérivation d'un vecteur ($\boldsymbol{f}_1 = \boldsymbol{p} = (p_x, p_y, p_z)$) par rapport à un vecteur ($\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$). On peut donc l'écrire sous la forme d'une matrice 3X3 :

$$\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\omega}} = \begin{pmatrix} \frac{\partial p_x}{\partial \omega_x} & \frac{\partial p_x}{\partial \omega_y} & \frac{\partial p_x}{\partial \omega_z} \\ \frac{\partial p_y}{\partial \omega_x} & \frac{\partial p_y}{\partial \omega_y} & \frac{\partial p_y}{\partial \omega_z} \\ \frac{\partial p_z}{\partial \omega_x} & \frac{\partial p_z}{\partial \omega_y} & \frac{\partial p_z}{\partial \omega_z} \end{pmatrix} \quad (88)$$

A partir du résultat donné par Maple, nous avons alors identifié les termes, et nous avons écrit le résultat sous la forme matricielle :

$$\begin{array}{rcccccc} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\omega}} = K_2 \mathrm{sk}(\boldsymbol{p}) & + & \big(a\mathrm{sk}(\boldsymbol{\omega}dt) + b\mathrm{sk}^2(\boldsymbol{\omega}dt)\big)\boldsymbol{p}\boldsymbol{\omega}^T & + & K_1\big(\boldsymbol{\omega}\boldsymbol{p}^T - 2\boldsymbol{p}\boldsymbol{\omega}^T + (\boldsymbol{\omega}^T\boldsymbol{p})\boldsymbol{I}_3\big) \\ -dt\big[K_4 \mathrm{sk}(\boldsymbol{v}) & + & \big(c\mathrm{sk}(\boldsymbol{\omega}dt) + d\mathrm{sk}^2(\boldsymbol{\omega}dt)\big)\boldsymbol{v}\boldsymbol{\omega}^T & + & K_3\big(\boldsymbol{\omega}\boldsymbol{v}^T - 2\boldsymbol{v}\boldsymbol{\omega}^T + (\boldsymbol{\omega}^T\boldsymbol{v})\boldsymbol{I}_3\big)\big] \\ -\frac{dt^2}{2}\big[K_6 \mathrm{sk}(\boldsymbol{a}) & + & \big(e\mathrm{sk}(\boldsymbol{\omega}dt) + f\mathrm{sk}^2(\boldsymbol{\omega}dt)\big)\boldsymbol{a}\boldsymbol{\omega}^T & + & K_5(\boldsymbol{\omega}\boldsymbol{a}^T - 2\boldsymbol{a}\boldsymbol{\omega}^T + (\boldsymbol{\omega}^T\boldsymbol{a})\boldsymbol{I}_3)\big] \end{array} \quad (89)$$

avec

$$\begin{aligned} a &= \frac{-\cos\|\boldsymbol{\omega}dt\|dt^2}{\|\boldsymbol{\omega}dt\|^2} + \frac{\sin\|\boldsymbol{\omega}dt\|dt^2}{\|\boldsymbol{\omega}dt\|^3} \\ b &= \frac{\sin\|\boldsymbol{\omega}dt\|dt^2}{\|\boldsymbol{\omega}dt\|^3} - \frac{2(1-\cos\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^4} \\ c &= -b \\ d &= \frac{(1-\cos\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^4} - \frac{3(\|\boldsymbol{\omega}dt\|-\sin\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^5} \\ e &= \frac{2(\cos\|\boldsymbol{\omega}dt\|-1)dt^2}{\|\boldsymbol{\omega}dt\|^4} + \frac{6(\|\boldsymbol{\omega}dt\|-\sin\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^5} \\ f &= \frac{-4(\|\boldsymbol{\omega}dt\|^2-2+2\cos\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^6} + \frac{2dt^2}{\|\boldsymbol{\omega}dt\|^4} - \frac{2\sin\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^5} \end{aligned}$$

$$\begin{aligned} K_1 &= \frac{(1-\cos\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^2} & , K_2 &= \frac{\sin\|\boldsymbol{\omega}dt\|dt}{\|\boldsymbol{\omega}dt\|} \\ K_3 &= \frac{(\|\boldsymbol{\omega}dt\|-\sin\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^3} & , K_4 &= \frac{(1-\cos\|\boldsymbol{\omega}dt\|)dt}{\|\boldsymbol{\omega}dt\|^2} \\ K_5 &= \frac{(\|\boldsymbol{\omega}dt\|^2-2+2\cos\|\boldsymbol{\omega}dt\|)dt^2}{\|\boldsymbol{\omega}dt\|^4} & , K_6 &= \frac{2(\|\boldsymbol{\omega}dt\|-\sin\|\boldsymbol{\omega}dt\|)dt}{\|\boldsymbol{\omega}dt\|^3} \end{aligned}$$

On rappelle que $\mathrm{sk}(\boldsymbol{\theta})$ est la matrice antisymétrique associée à $\boldsymbol{\theta}$ et $\boldsymbol{I}_3$ la matrice identité de rang 3.

Nous avons également utilisé les relations suivantes, montrées dans [14] :

$$\begin{aligned} \mathrm{sk}(\boldsymbol{u})\mathrm{sk}(\boldsymbol{v}) &= \boldsymbol{v}\boldsymbol{u}^T - (\boldsymbol{u}^T\boldsymbol{v})\boldsymbol{I}_3 \\ \mathrm{sk}(\boldsymbol{u})\mathrm{sk}(\boldsymbol{v}) - \mathrm{sk}(\boldsymbol{v})\mathrm{sk}(\boldsymbol{u}) &= \boldsymbol{v}\boldsymbol{u}^T - \boldsymbol{u}\boldsymbol{v}^T = \mathrm{sk}\big(\mathrm{sk}(\boldsymbol{u})\boldsymbol{v}\big) \end{aligned}$$

Restent à calculer $\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\theta}}$ et $\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\omega}}$. Les expressions obtenues avec Maple n'ayant rien de « sympathique », il s'est avéré difficile de les factoriser par des coefficients et des matrices simples. Nous avons alors généré le code correspondant à partir de Maple, et nous nous sommes contentés d'insérer ce code dans notre programme.

E.2 Calcul approché

Un calcul approché semble intéressant pour les termes $\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\theta}}$, $\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\omega}}$ et $\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\omega}}$, puisque leur calcul exact donne des formules très lourdes. Il s'agit de raisonner en vitesses au lieu de raisonner sur les positions, ce qui revient à approcher les dérivées partielles au premier ordre.

L'équation d'état concernant $\boldsymbol{p}$ est issue de l'intégration de $\dot{\boldsymbol{p}}$. Or,

$$\dot{\boldsymbol{p}}(t) = -\boldsymbol{v} - \boldsymbol{\omega} \times \boldsymbol{p} - \boldsymbol{a}t \tag{90}$$

ce qui s'écrit également

$$\begin{aligned} \dot{\boldsymbol{p}}(t) &= -\boldsymbol{v} - \mathrm{sk}(\boldsymbol{\omega})\boldsymbol{p} - \boldsymbol{a}t \\ &= -\boldsymbol{v} + \mathrm{sk}(\boldsymbol{p})\boldsymbol{\omega} - \boldsymbol{a}t \end{aligned} \tag{91}$$

D'où, en approximant au premier ordre,

$$\frac{\boldsymbol{p}(t+dt) - \boldsymbol{p}(t)}{dt} \simeq -\boldsymbol{v}(t) + \mathrm{sk}\big(\boldsymbol{p}(t)\big)\boldsymbol{\omega}(t) \tag{92}$$

et donc

$$\boldsymbol{p}(t+dt) \simeq \boldsymbol{p}(t) - \boldsymbol{v}(t)dt + \mathrm{sk}\big(\boldsymbol{p}(t)\big)\boldsymbol{\omega}(t)dt \tag{93}$$

On en déduit :

$$\frac{\partial \boldsymbol{p}(t+dt)}{\partial \boldsymbol{\omega}} \simeq \mathrm{sk}\big(\boldsymbol{p}(t)\big)dt \tag{94}$$

Par le calcul exact, on obtenait pour $\frac{\partial \boldsymbol{p}(t+dt)}{\partial \boldsymbol{\omega}}$ le résultat donné en (89).
Or,

$$\lim_{\boldsymbol{\omega}\to\mathbf{0}_3} \big(a\mathrm{sk}(\boldsymbol{\omega}dt) + b\mathrm{sk}^2(\boldsymbol{\omega}dt)\big)\boldsymbol{p}\boldsymbol{\omega}^T + K_1\big(\boldsymbol{\omega}\boldsymbol{p}^T - 2\boldsymbol{p}\boldsymbol{\omega}^T + (\boldsymbol{\omega}^T\boldsymbol{p})\boldsymbol{I}_3\big) = \mathbf{0}_3 \quad (95)$$

et les deux autres termes similaires en $\boldsymbol{v}$ et $\boldsymbol{a}$ tendent aussi vers zéro.

De plus,

$$\lim_{\boldsymbol{\omega}\to\mathbf{0}_3} K_2 = dt \quad ; \lim_{\boldsymbol{\omega}\to\mathbf{0}_3} K_4 = \frac{1}{2}dt \quad ; \lim_{\boldsymbol{\omega}\to\mathbf{0}_3} K_6 = \frac{1}{3}dt \quad (96)$$

d'où

$$\lim_{\boldsymbol{\omega}\to\mathbf{0}_3} \frac{\partial \boldsymbol{p}(t+dt)}{\partial \boldsymbol{\omega}} = \mathrm{sk}\big(\boldsymbol{p}(t)\big)dt - \frac{1}{2}dt^2\mathrm{sk}\big(\boldsymbol{v}(t)\big) - \frac{1}{6}dt^3\mathrm{sk}\big(\boldsymbol{a}(t)\big) \quad (97)$$

c'est-à-dire, en approximant au premier ordre,

$$\lim_{\boldsymbol{\omega}\to\mathbf{0}_3} \frac{\partial \boldsymbol{p}(t+dt)}{\partial \boldsymbol{\omega}} = \mathrm{sk}\big(\boldsymbol{p}(t)\big)dt \quad (98)$$

ce qui correspond bien à la valeur trouvée en (94).

Calculons à présent $\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\theta}}$ et $\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{\omega}}$. Dérivons (74) par rapport au temps. On obtient

$$\dot{\boldsymbol{R}}\big(\boldsymbol{\theta}(t+dt)\big) = \dot{\boldsymbol{R}}^T\big(\boldsymbol{\omega}(t)dt\big)\boldsymbol{R}\big(\boldsymbol{\theta}(t)\big) + \boldsymbol{R}^T\big(\boldsymbol{\omega}(t)dt\big)\dot{\boldsymbol{R}}\big(\boldsymbol{\theta}(t)\big) \quad (99)$$

Par ailleurs, (74) donne par transposition

$$\boldsymbol{R}^T(\boldsymbol{\theta}(t+dt)) = \boldsymbol{R}^T\big(\boldsymbol{\theta}(t)\big)\boldsymbol{R}\big(\boldsymbol{\omega}(t)dt\big) \quad (100)$$

En multipliant à droite (99) par (100), on obtient alors

$$\begin{aligned}&\dot{\boldsymbol{R}}\big(\boldsymbol{\theta}(t+dt)\big)\boldsymbol{R}^T(\boldsymbol{\theta}(t+dt)) = \\ &\dot{\boldsymbol{R}}^T\big(\boldsymbol{\omega}(t)dt\big)\boldsymbol{R}\big(\boldsymbol{\omega}(t)dt\big) + \boldsymbol{R}^T\big(\boldsymbol{\omega}(t)dt\big)\dot{\boldsymbol{R}}\big(\boldsymbol{\theta}(t)\big)\boldsymbol{R}^T\big(\boldsymbol{\theta}(t)\big)\boldsymbol{R}\big(\boldsymbol{\omega}(t)dt\big)\end{aligned} \quad (101)$$

Or, une propriété des matrices de rotation est

$${}^0\dot{\boldsymbol{R}}_1\,{}^0\boldsymbol{R}_1^T = \mathrm{sk}({}^0\boldsymbol{\omega}_1) \quad (102)$$

où ${}^0\boldsymbol{\omega}_1$ est le vecteur vitesse de rotation du repère 0 par rapport au repère 1.

En appliquant cette relation à (101), on obtient

$$\mathrm{sk}(\boldsymbol{\omega_{\theta 2}}) = -\mathrm{sk}(\boldsymbol{\omega}(t)) + \boldsymbol{R}^T\big(\boldsymbol{\omega}(t)dt\big)\mathrm{sk}(\boldsymbol{\omega_{\theta 1}})\boldsymbol{R}\big(\boldsymbol{\omega}(t)dt\big) \quad (103)$$

en notant

– $\boldsymbol{\omega_{\theta 2}}$ la vitesse de rotation de $\boldsymbol{R}\big(\boldsymbol{\theta}(t+dt)\big)$,

– $\boldsymbol{\omega_{\theta 1}}$ la vitesse de rotation de $\boldsymbol{R}\big(\boldsymbol{\theta}(t)\big)$, et

– $\boldsymbol{\omega}$ la vitesse de rotation de $\boldsymbol{R}\big(\boldsymbol{\omega}(t)dt\big)$.
En négligeant le terme $\boldsymbol{R}^T\big(\boldsymbol{\omega}(t)dt\big)\mathrm{sk}(\boldsymbol{\omega_{\theta 1}})\boldsymbol{R}\big(\boldsymbol{\omega}(t)dt\big)$, on obtient

$$\mathrm{sk}(\boldsymbol{\omega_{\theta 2}}) \simeq -\mathrm{sk}(\boldsymbol{\omega}(t)) \tag{104}$$

et donc

$$\boldsymbol{\omega_{\theta 2}} \simeq -\boldsymbol{\omega}(t) \tag{105}$$

Par ailleurs, d'après [2, 14], la matrice d'interaction associée au vecteur de rotation $\theta\boldsymbol{u}$ (où $\boldsymbol{u}$ est unitaire et où θ est la norme de l'angle) est donnée par

$$\boldsymbol{L_\omega} = \boldsymbol{I}_3 - \frac{\theta}{2}\mathrm{sk}(\boldsymbol{u}) + \Big(1 - \frac{\mathrm{sinc}\theta}{\mathrm{sinc}^2\frac{\theta}{2}}\Big)\mathrm{sk}^2(\boldsymbol{u}) \tag{106}$$

On a donc, avec (105)

$$\begin{aligned} \dot{\boldsymbol{\theta}}(t+dt) &= \boldsymbol{L_{\omega_{\theta 2}}}\boldsymbol{\omega_{\theta 2}} && (107)\\ &\simeq -\boldsymbol{L_{\omega_{\theta 2}}}\boldsymbol{\omega} && (108) \end{aligned}$$

En procédant de la même manière que pour le cas de $\boldsymbol{p}$ (en (92)), on a alors

$$\frac{\boldsymbol{\theta}(t+dt) - \boldsymbol{\theta}(t)}{dt} \simeq -\boldsymbol{L_{\omega_{\theta 2}}}\boldsymbol{\omega} \tag{109}$$

d'où

$$\boldsymbol{\theta}(t+dt) \simeq \boldsymbol{\theta}(t) - \boldsymbol{L_{\omega_{\theta 2}}}\boldsymbol{\omega}dt \tag{110}$$

On en déduit

$$\begin{aligned} \frac{\partial\boldsymbol{\theta}(t+dt)}{\partial\boldsymbol{\theta}} &\simeq \boldsymbol{I}_3 && (111)\\ \frac{\partial\boldsymbol{\theta}(t+dt)}{\partial\boldsymbol{\omega}} &\simeq -\boldsymbol{L_{\omega_{\theta 2}}}dt && (112) \end{aligned}$$

où $\boldsymbol{L_{\omega_{\theta 2}}}$ est donné par (106), avec $\boldsymbol{\theta}(t+dt)$ pour vecteur de rotation.

Références

[1] Y. Bar-Shalom et X.-R. Li
Estimation and tracking : principles, techniques, and software
Artech House, Boston, London
1993.

[2] F. Chaumette
Dans : *La commande des robots manipulateurs*, Asservissement visuel
W. Khalil (ed.), Chapitre 3, P. 105-150,
Traité IC2, Hermès, 2002.

[3] F. Chaumette
A first step toward visual servoing using image moments
IROS'02, Volume 1, P. 378-383,
Lausanne, Suisse, Octobre 2002.

[4] A. Comport, E. Marchand, F. Chaumette
A real-time tracker for markerless augmented reality
ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03, P. 36-45,
Tokyo, Japon, Octobre 2003.

[5] D. Dementhon, L. Davis
Model-based object pose in 25 lines of code
International Journal of Computer Vision, Volume 15, P. 123-141,
Juin 1995.

[6] C. Fermuller, Y. Aloimonos
Observability of 3D motion
International Journal of Computer Vision, 37(1) : 43-64,
Juin 2000.

[7] T. Drummond, R. Cipolla
Real-Time Visual Tracking of Complex Structures
IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 24, No.7, P. 932-946
Juillet 2002.

[8] T. Hamel, R. Mahony
Visual servoing of an under-actuated rigid body system : An image based approach
IEEE-Transactions on Robotics and Automation, Volume 18, No.2, P. 187-198
Avril 2002.

[9] K. Hashimoto ed.
Visual Servoing : Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback
World Scientific Series in Robotics and Automated Systems, Volume 7, World Scientific,
Singapore, 1993.

[10] S. Hutchinson, G. Hager, P.I. Corke
A tutorial on visual servo control
IEEE-Transactions on Robotics and Automation, Volume 12, No. 5, P. 651-670
1996.

[11] D. Koller, K. Daniilidis, et H.-H. Nagel
Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes
International Journal of Computer Vision, Volume 10, No.3, P. 257-281,
Juin 1993.

[12] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, M. Tuceryan
Real-time Vision-Based Camera Tracking for Augmented Reality Applications
ACM Symposium on Virtual Reality Software and Technology, P. 87-94,
Lausanne, Suisse, Septembre 1997.

[13] R. Mahony, T. Hamel, F. Chaumette
A Decoupled Image Space Approach to Visual Servo Control of a Robotic Manipulator
IEEE Int. Conf. on Robotics and Automation, ICRA'02, P. 3781-3786,
Washington, Mai 2002.

[14] E. Malis
Contributions à la modélisation et à la commande en asservissement visuel
Thèse de l'Université de Rennes 1, Télécommunications et Traitement du Signal,
Novembre 1998.

[15] E. Marchand, F. Chaumette
Virtual Visual Servoing : a framework for real-time augmented reality
Eurographics'2002, Volume 21(3), P. 289-298,
Saarebrück, Allemagne, Septembre 2002.

[16] P. Pounds, R. Mahony, P. Hynes, J. Roberts
Design of a Four-Rotor Aerial Robot

Proc. 2002 Australiasian Conference on Robotics and Automation, P.145-150, Auckland, Novembre 2002.

[17] M. Pressigout, E. Marchand
Model-free augmented reality by virtual visual servoing
ICPR Int. Conf. on Pattern Recognition, ICPR'04,
Cambridge, Royaume-Uni, Août 2004.

[18] P. Rives, J.R. Azinheira
Linear Structures Following by an Airship using Vanishing Point and Horizon Line in Visual Servoing Scheme
IEEE Int. Conf. on Robotics and Automation, ICRA'04, P. 255-260,
New Orleans, Avril 2004.

[19] S. Saripalli, J.F. Montgomery, G. Sukhatme
Vision-based Autonomous Landing of an Unmanned Aerial Vehicle
IEEE Int. Conf. on Robotics and Automation, ICRA'02, P. 2799-2804,
Washington, Mai 2002.

[20] O. Tahri, F. Chaumette
Application of moment invariants to visual servoing
IEEE Int. Conf. on Robotics and Automation, ICRA'03, P. 4276-4281,
Taipeh, Taiwan, Mai 2003.

[21] O. Tahri
Application des moments à l'asservissement visuel et au calcul de pose
Thèse de l'Université de Rennes 1, mention informatique,
Rennes, Mars 2004.

[22] G. Welch, G. Bishop
An introduction to the Kalman filter
Technical Report TR 95-041, 2001,
University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175.

[23] Z. Zhang
Analyse du mouvement à partir d'une séquence de scènes stéréoscopiques et applications à la robotique mobile
Thèse de l'Université de Paris-Sud, Centre d'Orsay,
Octobre 1990.