

Drone quadrirotor suivant une ligne par vision

Guillaume Sanahuja

Heudiasyc UMR 7253, Université de Technologie de Compiègne, Centre de Recherches de
Royallieu, BP 20529,60205 COMPIEGNE cedex, FRANCE

guillaume.sanahuja@hds.utc.fr

Resumé

Cet article présente le démonstrateur drone issu de nos travaux de recherche ; il s'agit d'un quadrirotor, bien connu pour sa facilité de mise en œuvre. Ce démonstrateur (voir figure 1) fait partie de la flotte de drone que le laboratoire Heudiasyc va acquérir dans le cadre du projet Robotex, [9]. L'application proposée est un suivi de ligne de façon totalement autonome. Pour cela le drone est équipé notamment d'une caméra fixant le sol, et le DSP intégré au calculateur embarqué va traiter ses images et détecter la ligne par la transformée de Hough, ainsi que calculer le flux optique. La position de la ligne est utilisée pour stabiliser le drone latéralement et en angle de lacet, alors que le flux optique permet de réguler la vitesse d'avance du drone.

Mots-clés: drones, lois de commandes non linéaires, flux optique, transformée de Hough, systèmes embarqués.

1 Introduction

Un drone aérien ou *UAV* (*Unmanned Air Vehicule* en anglais) ou plus simplement drone, est un engin volant sans pilote. Les premiers essais d'avions sans pilote datent de 1897. Le français Octave Détable invente en effet un système de stabilité automatique basé sur une voilure à cônes divergents et l'utilise sur un planeur. Dans les années 1930, des avions télépilotés servent de cibles pour les entraînements de défense anti-aérienne. Plus tard, des bombes volantes ou missiles (type *V1* ou *V2*) sont conçus par l'armée allemande. C'est enfin pendant la guerre froide que les drones ont pris leur essor, les États-Unis développant alors des avions sans pilotes capables de faire de la reconnaissance sans engager de vies humaines sur le terrain ennemi.

De nos jours, les drones sont devenus beaucoup plus accessibles. En effet, la technologie ayant évolué, les systèmes se sont miniaturisés et les coûts ont baissé. De nombreuses équipes de recherche travaillent maintenant sur des drones miniatures, et la production scientifique sur le sujet est donc abondante. De nouvelles applications s'ouvrent alors au drone qui ne sont plus qu'à finalités militaires mais aussi civiles : surveillance des feux de forêt, surveillance du trafic autoroutier, inspection des ouvrages d'arts, inspection des lignes électrifiées, etc.

La problématique principale des drones est la stabilisation. En effet, l'engin étant autonome il doit être capable de mesurer son orientation, et éventuellement sa position, afin de pouvoir voler de manière stable pour réaliser des applications de surveillance.

Le premier problème est donc l'estimation de l'orientation et de la position. Il s'agit bien d'une estimation car aucun capteur ne délivre une telle information brute. Il faut donc avoir recours à différents capteurs afin de fusionner les données. Dans le cas de l'orientation,

ces capteurs sont généralement de types inertiels (accéléromètres et gyromètres), aidés de magnétomètres pour le cap. Pour la position, les systèmes les plus communs sont le *GPS*, les télémètres ou les caméras. C'est ainsi un système de vision que le démonstrateur présenté utilise afin de se localiser dans son environnement.

En effet, les systèmes de vision sont de plus en plus populaires en robotique, car les informations que peut délivrer une caméra sont beaucoup plus nombreuses et variées que tout autre type de capteur. Ainsi, le drone est équipé d'une caméra pointant vers le sol mesurant le flux optique, et effectuant une transformée de Hough pour détecter une ligne.

L'article se décompose par la suite de la façon suivante : la partie 2 présente le drone, la partie 3 décrit les algorithmes de vision et la partie 4 présente l'application de suivi de ligne.

2 Drone quadrirotor

2.1 Modèle dynamique

Ce démonstrateur est un drone de type *X4*, ou quadrirotor (voir figure 1). Ce drone à voilure tournante a l'avantage de décoller et atterrir verticalement, mais aussi de pouvoir effectuer du vol stationnaire. Le quadrirotor est sans doute l'une des configurations de drone la plus utilisée. En effet la forme de la structure lui permet de ne pas avoir d'autre actionneur que ses quatre moteurs, ce qui la rend mécaniquement très simple et facile à construire. Les moments de roulis et tangage sont obtenus en jouant sur les vitesses de rotation de deux moteurs opposés ; de plus, deux des quatre rotors ont une hélice à pas inverse ce qui permet d'annuler le couple de lacet.



FIGURE 1 – Photo du démonstrateur.

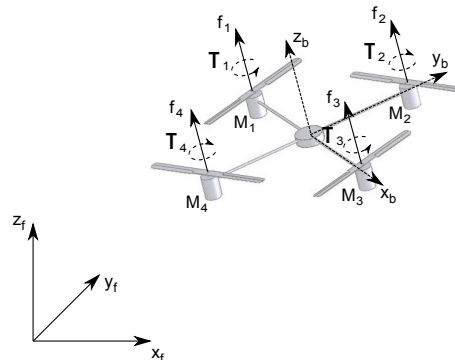


FIGURE 2 – Schéma du quadrirotor.

Les différents couples et forces dus aux moteurs pris en compte dans cette étude sont représentés sur la figure 2. Ainsi, chacun des quatre moteurs M_i produit une force f_i et un couple τ_i sur l'axe z_b . La poussée totale est donc $u = f_1 + f_2 + f_3 + f_4$. Le couple τ_{x_b} autour de l'axe x_b est obtenu par la différence de forces $f_2 - f_4$ et le couple τ_{y_b} autour de l'axe y_b par la différence de forces $f_1 - f_3$. Enfin le couple τ_{z_b} autour de l'axe z_b est obtenu par la somme des couples produits par les moteurs $\tau_1 + \tau_3 - \tau_2 - \tau_4$; M_1 et M_3 tournant en effet dans le sens positif alors que M_2 et M_4 tournent en sens inverse.

Le modèle dynamique du quadrirotor est obtenu en supposant que les pales des rotors sont indéformables et que la force de traînée ainsi que l'effet de sol sont négligeables. Malgré cela, le modèle obtenu (voir [1]) est non linéaire et contient de nombreux termes de couplage entre les différents états. Cependant, le quadrirotor est souvent étudié dans le

cas du vol stationnaire ou quasi stationnaire (déplacements latéraux très lents par faibles changements d'orientations). Dans ce cas, le modèle peut être linéarisé autour de la position d'équilibre et revient à :

$$m\ddot{x} = u(\theta + \psi\phi) \quad (1a)$$

$$m\ddot{y} = u(\theta\psi - \phi) \quad (1b)$$

$$m\ddot{z} = u - mg \quad (1c)$$

$$I_{b_{xx}}\ddot{\phi} = \tau_{x_b} \quad (1d)$$

$$I_{b_{yy}}\ddot{\theta} = \tau_{y_b} \quad (1e)$$

$$I_{b_{zz}}\ddot{\psi} = \tau_{z_b} \quad (1f)$$

où $\xi = [x, y, z]^T$ représente la position du centre de gravité du quadrirotor par rapport au repère fixe, $\eta = [\phi, \theta, \psi]^T$ sont ses angles d'Euler, m est sa masse et $I_{b_{xx}}$, $I_{b_{yy}}$, $I_{b_{zz}}$ les différents moments d'inertie.

2.2 Lois de commande

La loi de commande retenue est basée sur des fonctions de saturations, dans laquelle chaque état est borné séparément. La stabilité d'une telle loi a été prouvée dans les travaux de [8]. Appliqué au quadrirotor, nous obtenons :

$$u_z = \frac{-\sigma_{pz}(k_{pz}(z - z_d)) - \sigma_{dz}(k_{dz}\dot{z}) + mg}{\cos\theta \cos\phi} \quad (2a)$$

$$\tau_{x_b} = \frac{I_{b_{xx}}}{g} \left[\sigma_{py}(k_{py}(y - y_d)) + \sigma_{dy}(k_{dy}\dot{y}) - \sigma_{p\phi}(k_{p\phi}\phi) - \sigma_{d\phi}(k_{d\phi}\dot{\phi}) \right] \quad (2b)$$

$$\tau_{y_b} = -\frac{I_{b_{yy}}}{g} \left[\sigma_{px}(k_{px}(x - x_d)) + \sigma_{dx}(k_{dx}\dot{x}) + \sigma_{p\theta}(k_{p\theta}\theta) + \sigma_{d\theta}(k_{d\theta}\dot{\theta}) \right] \quad (2c)$$

$$\tau_{z_b} = -I_{b_{zz}} \left(\sigma_{p\psi}(k_{p\psi}\psi) + \sigma_{d\psi}(k_{d\psi}\dot{\psi}) \right) \quad (2d)$$

où k_{pz} , k_{dz} , k_{py} , k_{dy} , $k_{p\phi}$, $k_{d\phi}$, k_{px} , k_{dx} , $k_{p\theta}$, $k_{d\theta}$, $k_{p\psi}$ et $k_{d\psi}$ sont des constantes positives. Les variables x_d , y_d et z_d représentent la position désirée. Les fonctions σ_{pz} , σ_{dz} , σ_{py} , σ_{dy} , $\sigma_{p\phi}$, $\sigma_{d\phi}$, σ_{px} , σ_{dx} , $\sigma_{p\theta}$, $\sigma_{d\theta}$, $\sigma_{p\psi}$ et $\sigma_{d\psi}$ sont des fonctions de saturations du type :

$$\sigma_{b_i}(s) = b_i, \text{ si } :s > b_i \quad (3a)$$

$$\sigma_{b_i}(s) = s, \text{ si } : -b_i \leq s \leq b_i \quad (3b)$$

$$\sigma_{b_i}(s) = -b_i, \text{ si } :s < -b_i \quad (3c)$$

2.3 Le démonstrateur

Le drone est basé sur une structure de *Mikrokopter* et possède des moteurs *brushless* contrôlés par des drivers *i2c BICtrlv2*. La masse totale du drone est de 1200g. Il possède une batterie *LiPO 11.1V 6000mAh* lui permettant de voler environ 20 minutes. L'électronique de commande est basée sur un module *IGEPv2* possédant un *System On Chip (SOC) DM3730* de *Texas Instruments*. Ce *SOC* à l'avantage de posséder un cœur *ARM CortexA8* à 1GHz et un cœur *DSP C64x+* à 800MHz. La partie *ARM* permet de faire fonctionner

Linux ainsi que son extension temps réel *Xenomai*. La loi de commande fonctionne ainsi en temps réel à 100Hz. Le *DSP* de son coté est particulièrement adapté pour effectuer le traitement d'images, voir la section 3.

Le drone est par ailleurs équipé de plusieurs capteurs : une centrale inertielle *3DMGX3-25* délivrant les informations d'attitude à 100Hz, un télémètre à ultra sons *SRF08* donnant une mesure de l'altitude à 50Hz (de 0 à 2m) et une caméra *PS3Eye* pouvant fournir des images en 320x240 pixels jusqu'à 120 images par secondes.

Enfin, le drone est relié à sa station sol par une liaison *Wifi*. Depuis cette station, il est possible de voir en temps réel les graphes des mesures des différents capteurs ainsi que les résultats des lois de commande. Il est aussi possible de changer tous les paramètres des lois de commandes et filtres depuis la station sol et à tout moment. Pour les phases de vol manuelles, une manette de *Playstation 3* (reliée en *Bluetooth* à la station sol) est utilisée. Celle-ci permet alors d'envoyer des ordres de haut niveau au drone.

3 Vision

3.1 Flux optique

Le flux optique est une estimation du champ dense de déplacement visuel des points entre deux images. Ce dernier est en fait la projection en 2D sur l'image du déplacement des points 3D d'une scène. C'est donc un concept géométrique, qu'il n'est pas toujours possible de mesurer. En effet, le flux optique est en général calculé par une méthode basée sur le gradient supposant que l'intensité d'un point d'une scène est constante. Le calcul du flux optique ne peut donc pas fonctionner sur des objets uniformes, n'ayant pas de gradient d'intensité. De plus, le problème d'ouverture ne permet d'estimer le flux optique que dans la direction orthogonale à l'orientation locale de l'objet (voir [2] ou [3]).

Afin de décrire plus simplement le champ de déplacement, considérons un capteur visuel sphérique de rayon unitaire. L'image est alors formée par projection sphérique. Cela permet que tous les points de l'image soient géométriquement équivalents. La figure 3 représente ce modèle. Les points sur la sphère sont repérés par leur azimut ψ et leur élévation θ . Un objet est alors caractérisé par sa distance $D(\psi, \theta)$ suivant la direction $d(\psi, \theta)$, ce dernier vecteur étant unitaire. Le champ de déplacement est noté par un vecteur $p(\psi, \theta)$, tangent à la sphère. Les vecteurs T et R représentent respectivement la partie translationnelle et la partie rotationnelle du déplacement de la caméra dans son environnement.

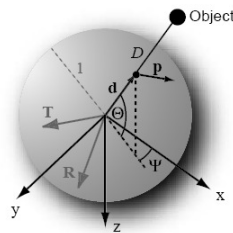


FIGURE 3 – Modèle de la caméra sphérique. Source : [4].

Ainsi, le champ de déplacement est donné par [5] :

$$p(\psi, \theta) = -\frac{T - (T^T d(\psi, \theta))d(\psi, \theta)}{D(\psi, \theta)} - [R]_{\times} d(\psi, \theta) \quad (4)$$

La méthode de Lucas et Kanade (voir [6] et [7]) est une des méthodes les plus populaire pour déterminer le flux optique. Elle suppose que le flux (u, v) est constant dans une petite fenêtre de dimension $m \times m$ (avec $m > 1$) centrée sur le pixel (x, y) . De plus, afin de traiter des champs de déplacement élevés une technique dite pyramidale peut être appliquée. Il s'agit d'effectuer plusieurs itérations, en commençant avec une échelle grossière et en affinant petit à petit. Le résultat d'une étape est alors utilisé à l'étape suivante afin de translater l'image de la valeur trouvée.

Cependant, le calcul du flux optique est assez lourd et est rarement embarqué dans un drone de petite taille tel que notre quadrirotor. L'algorithme de Lucas et Kanade a donc été spécialement optimisé pour le *DSP* embarqué, permettant de calculer le flux optique de 64 points d'une image de 320x240 pixels, à 100 images par secondes ; en utilisant deux niveaux de pyramides.

3.2 Transformée de Hough

La transformée de Hough est une technique de traitement d'image permettant de détecter des lignes ([10], [11]), qui a ensuite été généralisée pour tout type de formes ([12]). Pour notre application, nous utilisons la transformée de Hough classique, permettant de détecter des lignes droites.

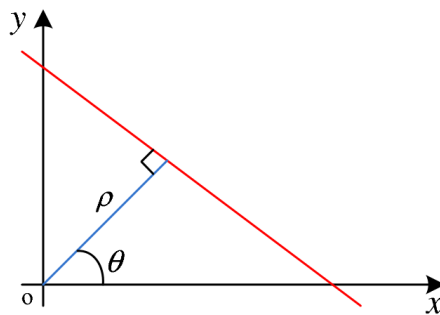


FIGURE 4 – Paramétrisation des droites

Son principe est de représenter toute droite par le couple (ρ, θ) , voir figure 4. Cette représentation est en effet meilleure qu'en utilisant une équation du type $y = ax + b$ qui ne permet pas de représenter une droite verticale. Pour chaque point (x_0, y_0) de l'image, l'ensemble des lignes passant par ce point peut ainsi se représenter de la manière suivante :

$$\rho(\theta) = x_0 \cos(\theta) + y_0 \sin(\theta) \quad (5)$$

La technique consiste alors à discrétiser les paramètres ρ et θ afin d'obtenir un nombre fini de droites. Ces deux paramètres forment un tableau nommé accumulateur et représentant l'ensemble des droites possibles de l'image. Pour chaque point de l'image pouvant appartenir à une droite, l'accumulateur est incrémenté d'un à chaque emplacement correspondant à une droite passant par ce point. L'accumulateur est ensuite classé par ordre décroissant, ce qui permet d'obtenir les droites rassemblant le plus grand nombre de points.

Cependant, avant d'appliquer la transformée de Hough, les images subissent d'autres traitements d'images :

- conversion en niveaux de gris,
- détecteur de contours par filtre de Sobel,
- seuillage pour éliminer le bruit et obtenir une image binaire.

Le filtre de Sobel permet de ne garder que les contours des objets, et donc de la ligne recherchée. Ce filtre va donc diminuer le nombre de points à traiter par la transformée de Hough. De même le seuillage élimine le bruit et donc le nombre de points à traiter par la suite. L'image binaire est alors utilisée pour remplir l'accumulateur.

Le calcul de la transformée a été optimisé pour utiliser le *DSP* embarqué. Notamment l'accumulateur doit pouvoir tenir dans la mémoire cache (64Ko) afin d'accélérer les accès (ceux ci étant aléatoires et non séquentiels, l'intégralité de l'accumulateur doit être en cache). De même, pour accélérer le traitement, une *lookup table* est créé pour précalculer les sinus et cosinus. Ces optimisations permettent de faire fonctionner l'ensemble des traitements (conversion en niveaux de gris, flux optique, détection de contours, seuillage et transformée de Hough) à 90 images par secondes.

4 Suivi de ligne

Les traitements d'images présentés précédemment sont ensuite utilisés pour déplacer le drone, grâce aux lois de commandes (voir section 2.2). De plus, deux cas sont distingués suivant que la ligne soit ou non dans le champ de vision de la caméra. Cela permet alors dans ce dernier cas au drone de revenir vers la ligne en intégrant le flux optique pour obtenir une estimation de la position. L'utilisation des différentes variables peut se résumer de la façon suivante :

- la ligne est détectée :
 - z et \dot{z} proviennent du capteur à ultra sons
 - y et ψ proviennent de la transformée de Hough
 - \dot{x} et \dot{y} proviennent du flux optique
 - θ , ϕ , $\dot{\theta}$, $\dot{\phi}$ et $\dot{\psi}$ proviennent de la centrale inertielle
- la ligne n'est pas détectée :
 - z et \dot{z} proviennent du capteur à ultra sons
 - \dot{x} , y et \dot{y} proviennent du flux optique
 - θ , ϕ , ψ , $\dot{\theta}$, $\dot{\phi}$ et $\dot{\psi}$ proviennent de la centrale inertielle

La preuve détaillée de la stabilité du système entre les deux cas peut se trouver dans nos travaux précédents [13].

5 Conclusion

Cet article a présenté notre démonstrateur drone issu de la flotte de drone Robotex. Celui-ci a été intégralement construit dans notre laboratoire et permet entre autres de valider expérimentalement les lois de commandes proposées par notre équipe. Un des points forts de ce démonstrateur est que le traitement d'images s'effectue en embarqué, ce qui lui confère un grand degré d'autonomie.

Ainsi, nous proposons une démonstration où le drone effectue un vol autonome en suivant la ligne tracée au sol. Le système est robuste aux perturbations et sait traiter le cas où la ligne sort du champ de vision.

Nos travaux futurs seront de tester nos algorithmes à plus grande échelle, en tentant de suivre par exemple une route. Dans ce cas nous envisageons d'utiliser un avion plutôt qu'un quadricoptère ; l'avion étant une plateforme sur laquelle travaille aussi notre laboratoire.

6 Remerciement

G. Sanahuja est cofinancé par le Labex MS2T. Il a été soutenu par le gouvernement français, à travers les programmes *Investissements d'avenir* gérés par l'Agence Nationale de la Recherche (Référence ANR-11-IDEX-0004-02).

G. Sanahuja est cofinancé par l'Union européenne. L'Europe s'engage en Picardie avec le Fonds Européen de Développement Régional.

Références

- [1] P. Castillo, R. Lozano and A. Dzul, *Modelling and Control of Mini-Flying Machines*, Springer-Verlag in Advances in Industrial Control, juillet 2005.
- [2] D. Marr, *Vision : A Computational Investigation into the Human Representation and Processing of Visual Information*, W. H. Freeman, 1982.
- [3] H.A. Mallot, *Computational Vision : Information Processing in Perception and Visual Behavior*, The MIT Press, 2000.
- [4] J.C. Zufferey, *Bio-inspired vision-based flying robots*, Thèse de l'École Polytechnique Fédérale de Lausanne, 2005.
- [5] J.J. Koenderink and A.J. Van Doorn, *Facts on optic flow*, Biological Cybernetics, 56(4) : p247-254, juin 1987.
- [6] B.D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision*, Imaging understanding workshop, 1981.
- [7] B.D. Lucas, *Generalized Image Matching by the Method of Differences*, Carnegie Mellon University, 1984.
- [8] G. Sanahuja, P. Castillo and A. Sanchez, *Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system*, International Journal of Robust and Nonlinear Control, 2009, DOI : 10.1002/rnc.1494.
- [9] Site internet du projet Robotex : <http://equipex-robotex.fr>
- [10] Hough, *Method and means for recognizing complex patterns*. U. S. Patent 3, 069 654 , December 18 , 1962.
- [11] R.O. Duda and P.E. Hart, *Use of the Hough Transformation to Detect Lines and Curves in Pictures*, Comm. ACM, Vol. 15, pp. 11 ?15 (January, 1972)
- [12] D.H. Ballard, *Generalizing the Houghtransform to detectarbitraryshapes*, Pattern Recognition, Elsevier, Volume 13, Issue 2, pp. 111 ?122 (1981)
- [13] L.R. Garcia Carrillo, G. Flores, G. Sanahuja and R. Lozano, *Quad-Rotor Switching Control : An Application for the Task of Path Following*, American Control Conference, juin 2012.