

Système d'évitement d'obstacles biomimétique basé sur le flux optique. Application à un drone à voilure fixe en environnement urbain simulé.

Laurent Muratet*, Stéphane Doncieux*, Jean-Arcady Meyer*, Patrick Pirim** et Thierry Druot***

*Université Pierre et Marie Curie, AnimatLab/LIP6.

**BEV, Paris.

***Université Paul Sabatier de Toulouse.

E-mail: muratet@poleia.lip6.fr

Résumé— À partir d'un modèle physique réaliste et d'un moteur graphique 3D, plusieurs stratégies d'évitement d'obstacles ont été testées sur un drone à voilure fixe évoluant en milieu urbain. Ces stratégies s'inspirent de la façon dont certains animaux utilisent le flux optique pour déterminer leur distance aux objets environnants. Il s'avère qu'une stratégie basée sur des déplacements en zigzag n'est pas assez réactive, compte tenu de la dynamique du drone utilisé. En revanche, une stratégie visant à éliminer la composante de rotation du flux optique fournit des résultats encourageants en simulation, validés par d'autres résultats obtenus en utilisant un flux optique parfait. L'implémentation d'une telle stratégie sur un drone réel sera cependant rendue plus difficile à cause des effets du vent et des vibrations liées au moteur.

I. Introduction

L'Approche Animat, née en 1990, vise à produire des "animats", c'est-à-dire des animaux artificiels, ou des robots réels, inspirés de la biologie et capables de "survivre" ou d'assurer leur mission dans des environnements changeants et potentiellement dangereux [10]. Comme décrit et théorisé par Ashby [1], un animat comprend des senseurs, des actionneurs et une architecture de contrôle chargée de maintenir à tout moment ses "variables essentielles" dans leur "zone de viabilité" (Figure 1).

L'objectif du travail décrit dans cet article est la réalisation d'un système d'évitement d'obstacles biomimétique basé sur la perception du flux optique et destiné au contrôle d'un drone à voilure fixe évoluant à basse altitude en milieu urbain. Le contrôle d'un tel robot volant est un défi ambitieux qui s'intègre parfaitement dans l'Approche Animat puisqu'une telle plate-forme est très instable, confrontée à un environnement imprévisible et perturbé qui menace continuellement sa survie ou son intégrité de fonctionnement.

A terme, ce système d'évitement d'obstacles est destiné à être implémenté sur un drone réel, le Pégase 50 produit par l'ENSICA de Toulouse.

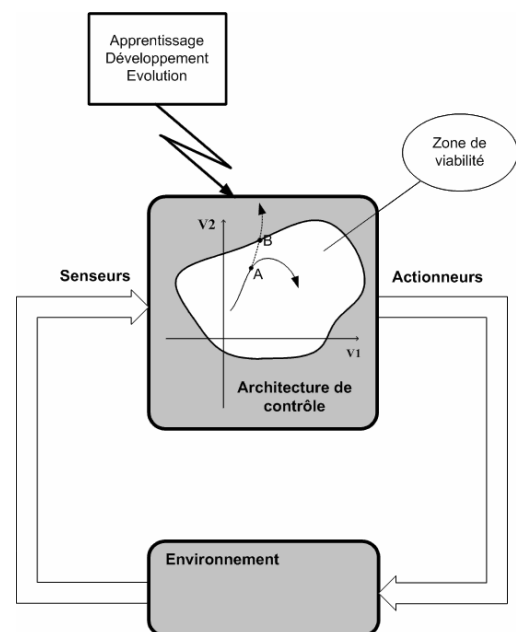


Figure 1. Organisation générale d'un animat. Un animat est "situé" dans un environnement réel ou simulé. Il a des senseurs qui le renseignent sur son état interne ou sur celui de l'environnement extérieur, des actionneurs qui lui permettent d'agir sur cet environnement, et une architecture de contrôle qui coordonne ses perceptions et ses actions. Le comportement d'un animat peut être qualifié d'adaptatif tant que son architecture de contrôle permet de maintenir ses variables essentielles (par exemple, $V1$ et $V2$) dans leur zone de viabilité. Ici, une action correctrice a été accomplie au point A, de façon à éviter de quitter la zone de viabilité au point B. L'architecture de contrôle d'un animat peut être modifiée par des processus adaptatifs d'apprentissage, développement et évolution.

Le travail présenté ici a été mené en simulation et a reposé sur la mise en œuvre d'un modèle physique réaliste du drone considéré, couplé à un moteur graphique 3D. Il a été ainsi possible de calculer quelle action exercer sur les actionneurs du drone en fonction des données visuelles qu'il acquiert à tout moment. Pour réaliser le moteur graphique, on a utilisé Crystal Space, une plate-forme de

développement 3D en libre accès servant à programmer des jeux vidéos. Quant à l'extraction des informations visuelles, elle a été réalisée grâce à une simulation de la puce GVPP [8] qui permet un traitement temps réel de données visuelles et qui sera utilisée sur le robot volant Pégase 50, comme elle l'a déjà été sur des robots terrestres [7].

Cet article décrit le modèle physique du drone utilisé, puis les principes de l'évitement d'obstacles par flux optique mis en œuvre dans ce travail. Il présente ensuite les résultats obtenus en implémentant 3 stratégies d'évitement différentes et évoque les développements futurs du système.

II. Modèle du drone

A. Modèle physique

Le robot simulé dans le cadre de ce travail est un drone à voilure fixe présentant six degrés de liberté: trois coordonnées (x, y, z) indiquent sa position et trois angles représentent son orientation dans l'espace: le cap (ψ) , l'assiette (θ) et l'inclinaison (ϕ) (Figure 2).

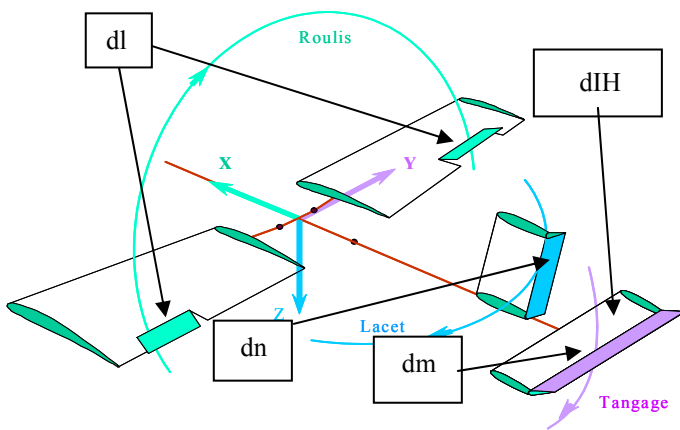


Figure 2: schéma de la plate-forme volante

Pour modéliser les lois physiques permettant le calcul de la trajectoire d'une telle plateforme volante soumise à des forces extérieures, nous avons considéré que l'expression de ces forces est linéaire, les coefficients correspondants étant déterminés d'après des essais en soufflerie ou par le calcul. Le modèle comporte trois repères orthonormés directs:

- Le repère sol sert de référence aux deux autres. Il permet de représenter la position du robot. Une convention en aérodynamique veut que l'axe des z soit orienté vers le bas.

- Le repère plate-forme a pour origine le centre de gravité du drone. L'axe des x correspond à l'axe du fuselage, il est orienté vers l'avant. L'axe des y est dans le plan défini par les ailes, il est orienté vers la droite de l'appareil lorsque celui-ci est vu de haut. Enfin, l'axe des z est orthogonal aux deux autres: le repère étant direct, il est orienté vers le dessous de l'avion. Ces trois axes sont représentés par leurs coordonnées exprimées dans le repère sol $(Xp_s, Yp_s$ et $Zp_s)$.

- Le repère aérodynamique a aussi pour origine le centre de gravité du robot. L'axe des x est le vecteur directeur de la trajectoire. L'axe des z est orthogonal à celui des x, orienté vers le bas, et il appartient au plan de symétrie de l'appareil. L'axe des y est orthogonal aux deux autres et son sens est défini de telle manière que le repère soit direct. Ces trois axes sont, eux aussi, représentés par leurs coordonnées exprimées dans le repère sol $(Xa_s, Ya_s$ et $Za_s)$. La projection de Xa_s dans le plan de symétrie de l'avion forme avec Xp_s un angle α appelé *incidence*. La projection de Xp_s dans le plan défini par Xa_s et Ya_s forme avec Xa_s un angle β appelé *dérapiage*. On ne définit pas ici de repère cinétique car, en l'absence de vent, ce dernier est confondu avec le repère aérodynamique. On peut passer d'un repère à l'autre en utilisant des matrices de changement de repère.

Dans ces conditions, le vecteur d'état du système s'écrit:

$$E = \begin{bmatrix} G \\ R \\ V \\ \Psi \\ C \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \\ \begin{bmatrix} X_{R1} \\ X_{R2} \\ X_{R3} \end{bmatrix} \\ \begin{bmatrix} Y_{R1} \\ Y_{R2} \\ Y_{R3} \end{bmatrix} \\ \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \\ \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \end{bmatrix} \\ \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \end{bmatrix}$$

Centre de gravité du drone
 Vecteur X de la base du drone
 Vecteur Y de la base du drone
 Vitesse du centre de gravité du drone
 Vecteur rotation du drone
 Vecteur de commande du drone

Le vecteur Z de la base du drone ne fait pas partie du vecteur d'état: une fois X et Y connus, il est déterminé par le fait que le repère (X, Y, Z) est orthonormé direct.

Un schéma d'intégration simple, de type Euler 1^{er} ordre, est utilisé pour actualiser le vecteur d'état du système:

$$E(t + dt) = E(t) + \left(\frac{dE}{dt} \right) dt$$

avec $dt = 40\text{ms}$.

Sa dérivée peut se calculer ainsi :

$$\frac{dE}{dt} = \left. \begin{array}{l} \left. \begin{array}{l} \left. \begin{array}{l} \left. \frac{dG}{dt} \Big|_{Sol} = V \\ \frac{dX}{dt} \Big|_{Sol} = \Psi \wedge X \\ \frac{dY}{dt} \Big|_{Sol} = \Psi \wedge Y \\ \frac{dV}{dt} \Big|_{Sol} = \frac{1}{m} \cdot \sum F_{ext} \end{array} \right\} a \\ \frac{d\Psi}{dt} \Big|_{Solide} = J_{[G]}^{-1} \cdot \left[\left(\sum M_{[G]}^{ext} \right) - \Psi \wedge \left(J_{[G]} \cdot \Psi \right) \right] \\ \frac{dC}{dt} = f(\text{commande}) \end{array} \right\} b \\ \end{array} \right\} c$$

a - Relations cinématiques.

b - Application du Principe Fondamental de la dynamique.

c - Loi de commande.

Pour connaître l'état du système à l'instant (t+dt), il suffit donc de calculer la dérivée de l'état à l'instant t en fonction des grandeurs du vecteur d'état et des sollicitations physiques appliquées au système.

Le programme de simulation complet intégrant le modèle physique, le moteur graphique et les différentes commandes comprend environ 15 000 lignes de code.

B. Senseurs et actionneurs

Les différentes stratégies d'évitement d'obstacles décrites dans cet article utilisent au plus deux capteurs : une caméra et une centrale inertielle.

La caméra est simulée grâce à un environnement 3D réaliste généré par Crystal Space. A partir de cet environnement virtuel, on produit 25 images par seconde, simulant ainsi un signal vidéo qui est lui-même transmis au simulateur de GVPP. Ce dernier extrait à partir de ce signal le flux optique nécessaire à l'évitement d'obstacles.

La centrale inertielle est simulée en utilisant les angles d'attitude (ψ , θ , φ) et la vitesse du drone fournis par le modèle physique. Ces valeurs sont ensuite bruitées pour simuler une erreur de mesure : on remplace la valeur de chacun des angles d'attitude V_{angle} par une valeur tirée d'une loi normale, de moyenne V_{angle} et d'écart-type un pourcentage de V_{angle} (par exemple 1%). Les résultats expérimentaux obtenus montrent que les stratégies utilisant la centrale inertielle sont assez robustes à ces erreurs de mesure lorsque celles-ci n'excèdent pas quelques pourcents - ce qui est une hypothèse réaliste.

Le drone est piloté au moyen de quatre ou cinq commandes (Figure 2):

- Trois gouvernes appelées dl, dm et dn ayant respectivement une action directe sur le roulis, le tangage et le lacet.

- La poussée moteur dx exprimée en pourcentage de la poussée maximale.

- Eventuellement diH, qui est le contrôle de l'angle du plan horizontal arrière par rapport au plan des ailes. Son influence sur le tangage est supérieure à dm car la surface de la gouverne de profondeur est nécessairement plus faible que le plan horizontal arrière qui la supporte.

Aucun degré de liberté du drone n'est figé. Cependant, le pilotage à l'aide du flux optique que nous avons considéré dans ce travail ne concerne que l'évitement d'obstacles dans le plan horizontal. Or, la survie du drone dépend non seulement de sa capacité à éviter les obstacles, mais aussi de sa capacité à ne pas s'écraser au sol. C'est pourquoi, cette consigne n'étant pas contrôlée par le flux optique, il faut s'assurer que le drone évolue à altitude constante et ajouter un automatisme adéquat aux deux autres automatismes servant à le contrôler. Les trois automatismes suivants sont donc mis en œuvre :

- Le rayon de virage est inversement proportionnel à la tangente de l'inclinaison φ de l'appareil. On peut donc éviter les obstacles en pilotant l'inclinaison grâce à dl. On

fixe une valeur de consigne φ_C et le système se charge de l'atteindre et de s'y maintenir. La formule s'écrit:

$$dl = \frac{1}{2} (\varphi - \varphi_C)$$

A la vue des résultats obtenus, il n'est pas apparu nécessaire d'ajouter un terme d'amortissement de cette consigne.

- La variation d'altitude dépend principalement de la poussée moteur et des commandes de profondeur (dm et diH), mais une inclinaison trop importante peut aussi entraîner une perte d'altitude. Il se trouve qu'on peut assurer une altitude presque constante en jouant simultanément sur dx et diH. Les commandes s'écrivent:

$$dx = \text{coeff}_1 + \text{coeff}_2 \times \frac{|\varphi_C|}{\frac{\pi}{6}}$$

$$diH = \text{coeff}_3 + \text{coeff}_4 \times \frac{|\varphi_C|}{\frac{\pi}{6}}$$

Les coefficients ont été obtenus par interpolation linéaire à partir de valeurs de dx, diH et φ de façon à ce que la variation d'altitude soit négligeable. Cette méthode très simple donne de bons résultats: la variation moyenne d'altitude est inférieure à 10 cm/s.

- Enfin, on se sert de dm pour stabiliser le *mode phugoïde* qui se traduit par une oscillation de θ autour de 0. Cet effet est dû à des variations opposées de la pente et de la vitesse; il provoque un échange entre énergie cinétique et énergie potentielle. La valeur de compensation utilisée dans ce travail est:

$$dm = \text{coeff}_5 \times \frac{V_z}{V}$$

III. Utilisation du flux optique

Le flux optique est la vitesse de défilement des objets sur le plan-image d'une caméra embarquée. Engendré par les mouvements de translation, il peut servir à évaluer à quelle distance un objet se trouve ou dans combien de temps un choc frontal est susceptible de se produire. Ce point a été découvert et exploité par la nature et sert de base à l'application biomimétique décrite ici.

A. Inspiration biologique

Dès 1865, Von Helmholtz avait expliqué selon quel principe certains animaux peuvent se baser sur leurs propres déplacements et sur la "parallaxe de mouvement" qui en résulte pour apprécier la distance qui les sépare d'objets environnants. Depuis, de nombreux travaux ont précisé les mécanismes correspondants. En particulier, ceux de Franceschini [4], ont permis de comprendre comment l'organisation de l'œil composé de la mouche et les traitements neuronaux des informations visuelles acquises en vol permettent à cet insecte d'estimer la distance des obstacles perçus latéralement et de les éviter. Ces connaissances se sont avérées suffisamment précises pour donner lieu à la réalisation d'un système opto-électronique inspiré du système visuel de la mouche et permettant à divers robots roulants ou volants d'éviter les obstacles [4], [5] et [6]. Les contrôles correspondants ne portent toutefois que sur un seul degré de liberté, de sorte que les animats volants ainsi utilisés ne sont pas entièrement libres de leurs mouvements. De plus, le problème de l'extraction du flux optique en temps réel est simplifié par l'usage d'environnements particuliers, composés de bandes blanches et noires pour exagérer les contrastes.

On a, de même, compris relativement récemment comment d'autres animaux utilisent le flux optique frontal pour estimer le *temps-de-contact*, c'est-à-dire le temps qui les sépare d'un choc frontal avec un objet. On suppose, par exemple, que c'est cette information qui est utilisée par le Fou de Bassan, lorsqu'il plonge à grande vitesse pour attraper un poisson, afin de décider du moment où replier ses ailes avant de pénétrer dans l'eau [9].

B. Principes de l'évitement

Le flux optique exploitable pour l'évitement d'obstacles est généré par une translation pure. Il apparaît en effet que, dans ce cas particulier, la vitesse angulaire de défilement d'un objet est inversement proportionnelle à la distance qui sépare cet objet de l'observateur. D'après la Figure 3, au moyen de quelques relations trigonométriques simples, on peut établir l'expression analytique de la vitesse horizontale angulaire $\dot{\beta}$:

$$\dot{\beta} = \frac{V_0 \times \sin(\beta)}{d} \quad (1)$$

Cette expression formalise la relation entre la vitesse angulaire de défilement d'un objet $\dot{\beta}$ et sa distance d à l'observateur.

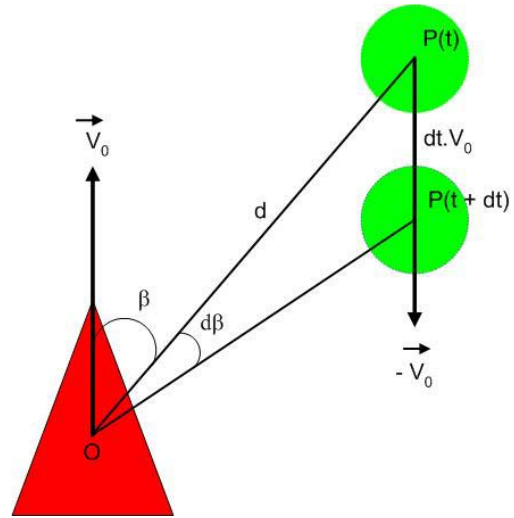


Figure 3: le drone perçoit l'objet vert suivant un angle β par rapport à sa direction. $\dot{\beta}$ est la vitesse de variation de cet angle. d est la distance séparant l'objet du drone. V_0 est la vitesse du drone.

Les stratégies d'évitement d'obstacles que nous avons mises en oeuvre sont inspirées des travaux de Duchon [3] et visent à maintenir le drone à égale distance des obstacles perçus sur sa droite et sur sa gauche. Dans la mesure où la puce GVPP fournit les vecteurs vitesses des pixels de l'image, et

non pas $\dot{\beta}$ directement, on préfère utiliser la projection de ces vecteurs sur l'axe horizontal, notée V_x – ce qui d'après (3) est équivalent. Dans ces conditions, la consigne d'inclinaison permettant l'évitement d'obstacle s'écrit:

$$\varphi_C = k \times (V_{x_{gauche}}^{moyenne} - V_{x_{droite}}^{moyenne}) \quad (2)$$

$V_{x_{gauche}}^{moyenne}$ et $V_{x_{droite}}^{moyenne}$ étant les moyennes des déplacements horizontaux des points de l'image vers la gauche et la droite, k étant un facteur qui pondère la sensibilité du drone à la différence des deux valeurs précédentes.

Il est clair toutefois que, si le drone se dirige perpendiculairement vers un obstacle, les flux optiques latéraux seront parfaitement égaux et que la stratégie d'équilibre qui vient d'être évoquée le conduira droit dans l'obstacle. Il faut donc ajouter à cette stratégie un réflexe qui tienne compte du temps-de-contact τ , lequel se déduit des vitesses observées proches de la direction de déplacement de l'agent (Figure 4). En effet, la valeur τ s'écrit:

$$\tau = \frac{1}{n} \sum \left(\frac{|\dot{\beta}|}{\beta} \right)$$

$\dot{\beta}$ est la vitesse angulaire d'un point observé. β est l'angle formé par le sens de déplacement du drone et la direction du point par rapport au drone. n est le nombre de points concernés par cette moyenne.

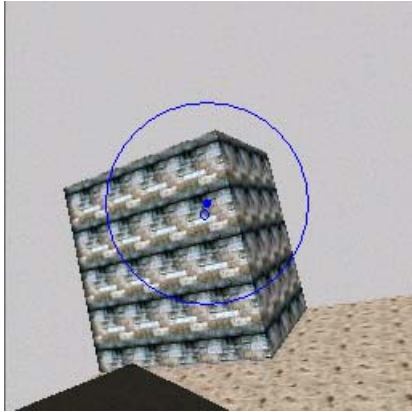


Figure 4: les vitesses des points intérieurs au cercle permettent de calculer τ . O est le point vers lequel le drone se dirige.

Si τ est inférieur à une valeur seuil, l'impact est imminent. Il faut alors interrompre la stratégie d'équilibre et générer une commande capable d'éviter la collision. Dans le cas d'un robot roulant, il est possible d'effectuer un demi-tour sur place. Une telle manœuvre ne pouvant être directement transposée à un drone à voilure fixe qui n'a pas de capacité de vol stationnaire, on se contente d'entreprendre un virage le plus serré possible du côté où le flux optique est le plus faible, c'est-à-dire du côté où les obstacles sont supposés être les plus éloignés.

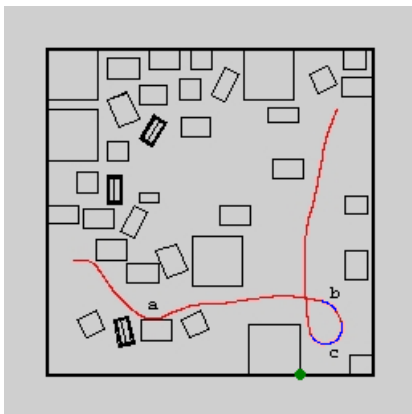


Figure 5: Trajectoire d'un drone vue de haut. Les obstacles sont représentés en noir.
a : l'obstacle n'est évité que tardivement car il est d'une hauteur faible, il ne déclenche pas le réflexe de virage.
b : le réflexe de virage est déclenché.
c : le réflexe est à nouveau déclenché ; la persistance dans le choix du sens du virage permet d'éviter l'angle en vert.

Il faut cependant rajouter de la persistance dans le choix du côté où l'on tourne afin de ne pas se diriger vers un angle formé par des bâtiments (Figure 5c). Ainsi, si τ est inférieur à une valeur-seuil, on choisit de tourner du côté où le flux optique est le plus faible, cette manœuvre étant poursuivie tant que τ est inférieur à ce seuil. Dans ces conditions, l'architecture de contrôle du drone est une *architecture de subsumption* [2]: le réflexe de virage d'urgence est prioritaire par rapport à la stratégie d'équilibre.

IV. Résultats expérimentaux

Comme évoqué précédemment, le flux optique nécessaire à l'évitement d'obstacles est généré par translation car il comporte une information sur la distance des objets perçus. En revanche, le flux optique généré par rotation ne dépendant pas de la distance des obstacles, il est inutilisable et parasite la mesure effectuée. Or, la translation et la rotation sont des déplacements qui peuvent être représentés par des matrices 3x4 homogènes, réalisant une simple somme:

$$F_{total} = F_{translation} + F_{rotation}$$

Partant de cette formule, trois possibilités existent:

- calculer analytiquement $F_{translation}$ pour se soustraire aux erreurs de mesure et se concentrer sur la réalisation d'une commande efficace.

- maintenir $F_{rotation}$ en dessous d'un seuil pour la considérer comme négligeable.

- calculer analytiquement $F_{rotation}$ pour la retrancher de la valeur observée.

Nous avons testé chacune de ces possibilités et présentons les résultats correspondants ci-dessous.

A. Flux optique théorique

Dans un premier temps, on cherche à supprimer les problèmes liés à l'extraction du flux optique pour se concentrer sur la détermination d'une commande efficace appliquée au drone à tout instant. Pour ce faire, on simule le

flux optique de translation $F_{translation}$ en tout point de l'image. Puisque cette dernière est calculée par le logiciel utilisé, il est possible de connaître, pour chaque pixel, la distance du point perçu (Figure 6). On peut donc calculer la vitesse horizontale angulaire grâce à la relation (1).

On cherche maintenant la correspondance entre cette vitesse angulaire $\dot{\beta}$ et la vitesse observée sur le plan-image V_x (Figure 7) pour simuler exactement la sortie de la puce GVPP. On ne s'intéresse ici qu'à la direction horizontale.

Puisque $Vx = Xp(t+dt) - Xp(t)$,

il vient: $Vx = f \times (\tan(\beta + \varepsilon) - \tan(\beta))$.

Par conséquent:

$$Vx = f \times \left(\tan\left(\beta + \dot{\beta} \times dt\right) - \tan(\beta) \right) \quad (3)$$

β dépend de la position horizontale du point par rapport au centre de l'écran:

$\beta = \arctan\left(\frac{Xp(t)}{f}\right)$, en supposant que le centre de l'écran est l'origine du repère horizontal.

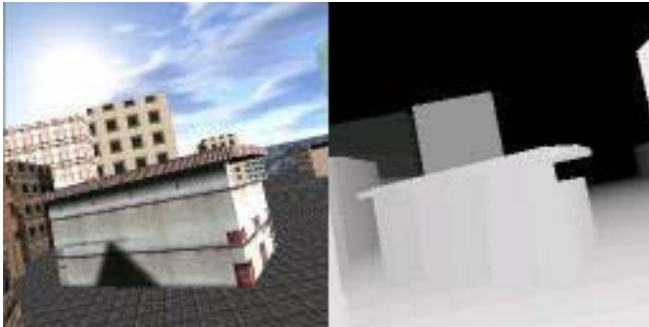


Figure 6: une image et la carte de profondeur associée en niveaux de gris. Plus un pixel est clair, plus il est proche.

Une fois ce déplacement horizontal théorique calculé, on peut générer la consigne traduite par la relation (2).

Cette méthode donne de bons résultats. Le drone a toutefois des difficultés à enchaîner des virages opposés serrés. Cette limite est due à l'inertie de l'appareil et au décalage dans le temps entre la commande et son effet.

On souhaite donc améliorer cette stratégie pour que le drone puisse anticiper ses décisions. Pour ce faire, on rajoute un terme dépendant de la dérivée de la valeur de décision. La nouvelle consigne s'écrit:

$$\varphi_C = k \times (Vx_{gauche}^{moyenne} - Vx_{droite}^{moyenne}) + w \times \frac{d(Vx_{gauche}^{moyenne} - Vx_{droite}^{moyenne})}{dt}$$

Dans cette équation, w est un coefficient déterminé de manière empirique. Ce terme d'amortissement de la consigne améliore considérablement les trajectoires du drone qui peut ainsi enchaîner des virages plus serrés.

Ces premiers résultats paraissent valider la stratégie d'équilibre basée sur le flux optique considérée ici, puisque le drone se déplace dans un environnement urbain réaliste sans heurter les bâtiments. Ils nous ont également permis de régler finement les paramètres k et w en édulcorant les problèmes inhérents à la mesure du flux optique.

Les deux autres séries d'expériences décrites maintenant reposent sur la gestion d'un flux optique qu'il n'est plus possible d'obtenir de manière exacte. Elles capitalisent sur

l'utilisation de GVPP pour obtenir un flux optique bruité, tel qu'il sera disponible sur le Pégase 50.

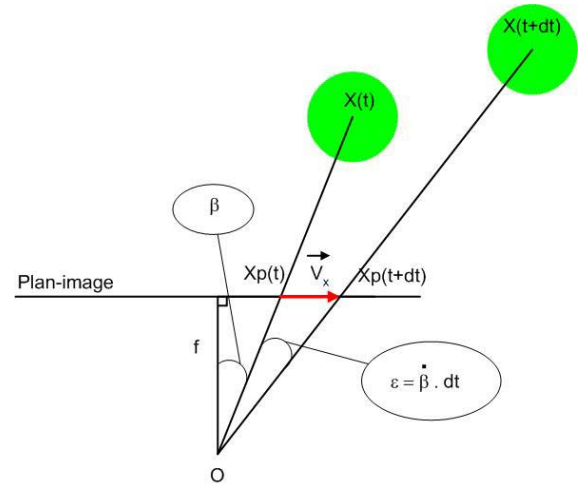


Figure 7: Correspondance entre vitesse angulaire et déplacement sur le plan-image. O est le centre optique de la caméra, f est la distance focale entre le centre optique et l'écran. Au temps t , l'agent perçoit l'objet X en vert selon une direction de β par rapport à son orientation. Au temps $t+dt$, l'objet X s'est déplacé par rapport à l'agent: il est perçu suivant un angle de $\beta+\varepsilon$. Xp est la projection de X sur le plan-image.

B. Stratégie de zigzag de la mouche

Dans la mesure où le flux optique exploitable pour l'évitement d'obstacles est généré par une translation pure, ce qui exclut toute composante de rotation, il apparaît que la mouche navigue de manière à réduire la composante de rotation pour prendre une décision correcte. A cet effet, elle adopte un déplacement en zigzag composé de lignes droites qui permettent de prendre des décisions, suivies de virages serrés durant lesquels le flux optique n'est pas exploité [4]. Cette stratégie ayant fait ses preuves, nous avons tenté de la transposer à notre plate-forme volante.

Ainsi, après avoir effectué une translation pure durant laquelle on calcule la consigne exprimée par la relation (2), on applique la commande correspondante tant que l'inclinaison φ n'a pas atteint sa valeur de consigne. Une fois que cette valeur est atteinte, la valeur de consigne est ramenée à 0. Toute décision à l'aide du flux optique est inhibée pendant cette phase de virage car le mouvement est entaché de plusieurs composantes de rotation. Ce cycle est répété pendant tout le vol.

Pour déterminer la durée d'une phase de translation pendant laquelle on peut considérer que la mesure du flux optique n'est pas parasitée par une composante de rotation, on utilise la centrale inertielle pour contrôler les variations de cap ψ , celles-ci ne devant pas excéder en valeur absolue une valeur-seuil ε . Pour déterminer cette valeur, en deçà de laquelle une rotation ne doit pas pouvoir être détectée par l'algorithme d'extraction du flux optique, on a tenu compte du fait que GVPP ne peut détecter des mouvements

inférieurs à $\frac{1}{3}$ de pixel. On a ensuite utilisé la correspondance entre une variation angulaire et un déplacement observé sur le plan-image dans la partie IV.A (Figure 7). On cherchait donc ε tel que $V_x \leq \frac{1}{3}$ pour tout point de l'écran. Or les points au bord de l'écran ont les vitesses les plus importantes: ils correspondent à une valeur de β de la moitié du champ de vision que l'on note FOV (*Field Of View*). D'où l'équation:

$$\varepsilon = \arctan\left(\frac{1}{3 \times f} + \tan\left(\frac{FOV}{2}\right)\right) + \frac{FOV}{2}$$

Les résultats obtenus avec cette stratégie de zigzag sont médiocres. En effet, l'inertie du drone affecte longuement la stabilisation du cap après chaque virage. Les décisions sont en conséquence très éloignées dans le temps les unes des autres. Si on relâche la contrainte ε , les résultats sont tout aussi décevants car la décision tombe vite sous l'influence de la rotation et tous les virages ont tendance à être effectués dans le même sens. Dans ce cas, en effet, le drone prend une décision alors qu'il est encore en train de tourner: le flux optique résultant est altéré dans le sens opposé au virage et la nouvelle décision est alors identique à l'ancienne.

Ainsi, il apparaît que la stratégie de zigzag est très peu réactive et donc inadéquate aux environnements encombrés. Un argument supplémentaire pousse à l'abandonner: la stabilisation du cap est très lente malgré l'absence complète de vent. Cette hypothèse est très réductrice et compromet la possibilité d'implémentation sur un drone à voilure fixe incapable d'exécuter un vol stationnaire. Toutefois, dans des environnements peu encombrés et en l'absence de vent, il se pourrait que cette stratégie produise des comportements acceptables.

C. Stratégie de compensation des rotations

La stratégie précédente a été écartée car aucun drone ne peut encore rivaliser avec la mouche en termes de voltige, qu'il s'agisse de rayon de virage ou de stabilisation du cap. Il apparaît cependant que cet animal, de moins de 2cm et d'à peine 1g, lorsqu'il est soumis à des perturbations extérieures, ne peut pas stabiliser son cap parfaitement et qu'il utilise l'information fournie par ses gyroscopes post-alaires pour compenser la composante de rotation intervenant dans sa détection du flux optique [4]. C'est cette idée que l'on a appliquée à notre drone dans cette dernière série d'expériences.

Grâce aux informations inertielles simulées - c'est-à-dire grâce aux variations des angles d'attitude (ψ , θ , ϕ) - on peut, en effet, déterminer $F_{rotation}$ pour le soustraire du flux observé F_{total} et obtenir ainsi l'information $F_{translation}$ nécessaire à la décision à tout instant.

Pour ce faire, on représente la composante de rotation par une matrice R 3×3 . On appelle R_ϕ , R_θ et R_ψ les matrices de

rotation autour des axes respectifs X, Y et Z. Ces matrices s'écrivent:

$$R_\phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\Delta\phi) & -\sin(\Delta\phi) \\ 0 & \sin(\Delta\phi) & \cos(\Delta\phi) \end{pmatrix}$$

$$R_\theta = \begin{pmatrix} \cos(\Delta\theta) & 0 & -\sin(\Delta\theta) \\ 0 & 1 & 0 \\ \sin(\Delta\theta) & 0 & \cos(\Delta\theta) \end{pmatrix}$$

$$R_\psi = \begin{pmatrix} \cos(\Delta\psi) & -\sin(\Delta\psi) & 0 \\ \sin(\Delta\psi) & \cos(\Delta\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

On compose ces matrices pour obtenir la matrice R . L'ordre de composition est très important car il dépend de la modélisation physique choisie. Ici, on détermine d'abord le cap, puis l'assiette et enfin l'inclinaison. On compose par la gauche les rotations dans cet ordre. On a:

$$R = R_\psi \times R_\theta \times R_\phi$$

Soient O le centre optique, f la distance focale et $X_p(t)$ le point (i, j) de l'écran (Figure 8). Les coordonnées de X_p dans le repère caméra sont (f, i, j) .

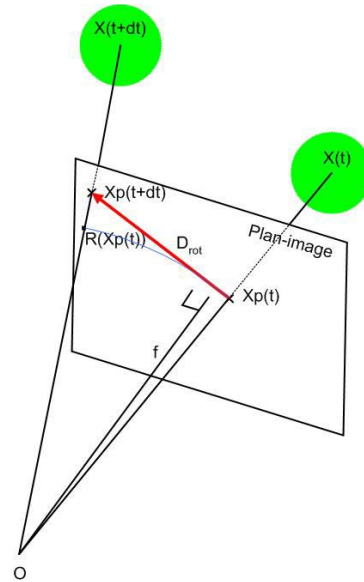


Figure 8: calcul de la composante de rotation

Au temps t , l'objet en vert est situé au point $X(t)$, le drone perçoit sa projection sur l'écran notée $X_p(t)$. Au temps $t+dt$, le drone a effectué une composition de rotations caractérisée par R . L'objet s'est donc déplacé par rapport à la caméra: il est au point $X(t+dt)$, sa projection sur le plan-image est $X_p(t+dt)$. Le déplacement dû à la rotation est noté D_{rot} . L'image de $X(t)$ par R , notée $R(X(t))$, n'appartient pas nécessairement à l'écran. $R(X(t))$ donne la direction du point de l'écran $X_p(t+dt)$. On projette $R(X(t))$ sur l'écran comme suit:

$$Xp(t+dt) = \frac{f}{X_{R(Xp(t))}} \times R(Xp(t))$$

Pour chaque pixel dont le mouvement est identifié par GVPP, on peut alors corriger le déplacement observé $D_{total}(t)$ et le déplacement engendré par la translation s'écrit:

$$D_{trans}(t) = D_{total}(t) - (Xp(t+dt) - Xp(t))$$

On ne tient pas compte de la coordonnée x de profondeur puisque l'on se situe dans le plan-image.

On exprime $D_{trans}(t)$ en notation polaire (ρ, θ) . Le critère de décision s'obtient en projetant ce vecteur sur la direction horizontale de la manière suivante:

$$V_{horizon} = \rho \times \cos(\theta + \varphi)$$

La règle de décision devient alors:

$$\varphi_c = k \times V_{horizon}^{moyenne} + w \times \frac{d(V_{horizon}^{moyenne})}{dt}$$

Les résultats obtenus avec cette règle sont mauvais parce que générés par des décisions contradictoires : le drone tourne successivement à droite, puis à gauche, et ainsi de suite. La compensation théorique est trop forte par rapport à la mesure. Le problème vient de l'imprécision avec laquelle GVPP calcule le flux optique (Figure 9). C'est pourquoi, dans le but d'éliminer les points pour lesquels GVPP indique un mouvement aberrant, nous n'avons utilisé pour le calcul du flux optique que les points qui, sur leurs 8 voisins possibles, en ont au moins n qui indiquent la même direction de déplacement. Pour estimer n, nous avons utilisé une caméra fixe que l'on a fait tourner suivants des angles donnés en différents endroits de l'environnement (Figure 9). On a ainsi déterminé qu'en fixant n à 3 on obtenait un bon compromis entre une faible erreur et une bonne représentativité des points.

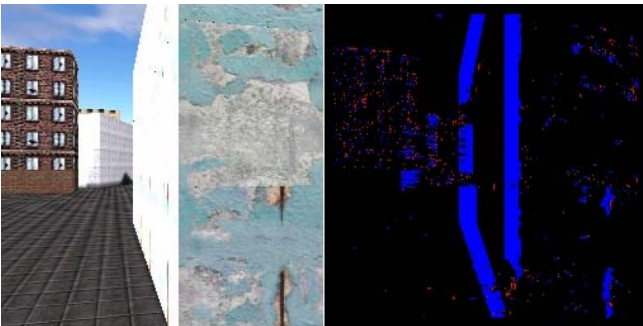


Figure 9: Suite à une rotation de la caméra vers la droite, GVPP singularise les points bleus comme se déplaçant vers la gauche et les points rouges comme se déplaçant vers la droite. Les points rouges sont donc aberrants.

Les résultats obtenus dans ces conditions sont nettement meilleurs. Si les décisions correspondantes ne sont pas indéfiniment correctes, le drone parvient néanmoins souvent

à enchaîner des virages de manière opportune. Dès lors la question des améliorations possibles de ces résultats se pose et est envisagée dans la discussion qui suit.

V. Discussion

La validité de la stratégie d'équilibre a été démontrée dans le cas où le flux optique était parfaitement exact. Lorsque ce n'est pas le cas, une erreur de mesure peut entraîner une mauvaise décision. La puce GVPP est plus ou moins sensible aux différentes textures utilisées en simulation et fournit un flux optique bruité. On peut penser qu'une prochaine version de la puce, encore en développement et susceptible de détecter des mouvements dans 360 directions, fournira un flux optique considérablement amélioré.

On peut également envisager d'implémenter dans la puce un algorithme intégrant les informations fournies par une centrale inertielle réelle en vue d'orienter la recherche du mouvement. En effet, dans le cas présent, on extrait F_{total} à partir de la séquence d'images, puis on calcule $F_{rotation}$ pour en déduire $F_{translation}$. L'idée serait d'utiliser les informations de la centrale inertielle directement au niveau de l'extraction pour orienter la recherche. On pourrait ainsi anticiper les mouvements de chaque pixel et gérer les 360 directions sans faire exploser la complexité. De plus, la puce fournirait directement $F_{translation}$, ce qui éviterait un calcul supplémentaire en aval de l'extraction.

Il est probable, enfin, que la contrainte de se limiter au plan horizontal pour éviter les obstacles restreint considérablement le domaine de viabilité du drone considéré. Dans les prolongements futurs de cette recherche, il conviendrait d'évaluer les conséquences sur la fiabilité du flux optique extrait, comme sur l'efficacité des stratégies d'évitement résultantes, de la possibilité de prendre de l'altitude pour passer au dessus d'un obstacle donné.

D'autres recherches doivent, à l'évidence, être consacrées à la mise en œuvre sur le drone Pégase 50 des stratégies d'évitement mises au point en simulation. Une attention particulière devra être portée aux effets du vent en milieu urbain - effets qu'on ne sait pas vraiment modéliser - ainsi qu'à ceux des vibrations dues au moteur. On peut craindre que ces effets soient difficiles à compenser lors de la décision car ils peuvent se produire sur un mode très rapide. De plus, ils pourraient dégrader fortement la qualité de l'image et jouer ainsi sur la pertinence du flux optique extrait.

VI. Conclusion

Les résultats décrits ici suggèrent qu'il est possible d'envisager d'utiliser la puce GVPP pour implémenter sur le Pégase 50 une stratégie d'évitement d'obstacles basée sur le flux optique. Plutôt que d'utiliser une stratégie basée sur des déplacements en zigzag, laquelle s'est avérée trop peu réactive, il semble qu'une stratégie de compensation des

rotations, basée sur un système inertielle, puisse s'avérer efficace, surtout si une version plus élaborée de GVPP est disponible à cet effet. Il reste que les conséquences du vent et des vibrations dues aux moteurs, conséquences non considérées dans ce travail, augmenteront certainement la difficulté de cette tâche.

[10] Meyer, J.A. and Guillot, A. (1991). A. Simulation of adaptive behavior in animats: review and prospect. *From animals to animats. Proceedings of the First International Conference on Simulation of Adaptive Behavior*. J.A. Meyer and S. Wilson (Eds.). Cambridge: The MIT Press. 1991.

BIBLIOGRAPHIE

[1] Ashby, W. R. (1960). *Design for a brain. The Origin of Adaptive Behavior*. Chapman & Hall, London.

[2] Brooks, R. A., (1991b). New approaches to robotics. *Science*. 253, 1227-1232.

[3] Duchon, A. P. (1996). Maze navigation using optical flow. *From animals to animats 4: Proceedings of the fourth international conference on simulation of adaptive behavior*, P. Maes, M. J. Mataric, J. A. Meyer, J. Pollack, S. W. Wilson (Eds). Cambridge: The MIT Press.

[4] Franceschini, N., Pichon, J. M., and Blanes, C. (1992). From insect vision to robot vision. *Philosophical Transactions of the Royal Society of London. Series B*, 283-294.

[5] Franceschini, N. and Mura, F. (1994). Visual control of altitude and speed in a flying agent. *From animals to animats 3..* D. Cliff, P. Husbands, J. A. Meyer, S. W. Wilson (Eds). Cambridge: The MIT Press.

[6] Franceschini, N. and Netter, T. (1999). Neuromorphic optical flow sensing for nap-of-the-earth flight. *Mobile robots XIV*, SPIE Vol. 3838. Bellingham.

[7] Gourichon, S., Meyer, J.A., Pirim, P. (2002). Using colored snapshots for short-range guidance in mobile robots. *International Journal of Robotics and Automation : Special Issue on Biologically Inspired Robotics*. 17, 4, 154-162.

[8] <http://gvpp.org/Technology.htm>

[9] Lee, D.N. (1980). The optic flow field: the foundation of vision. *Philosophical Transactions of The Royal Society of London*, Series B. 290, 169-179.