Abstract Hovering flies are able to stay still in place when hovering above flowers and burst into movement towards a new object of interest (a target). This suggests that sensorimotor control loops implemented onboard could be usefully mimicked for controlling Unmanned Aerial Vehicles (UAVs). In this study, the fundamental head-body movements occurring in free-flying insects was simulated in a sighted twin-engine robot with a mechanical decoupling inserted between its eye (or gaze) and its body. The robot based on this gaze control system achieved robust and accurate hovering performances, without an accelerometer, over a ground target despite a narrow eye field of view (±5°). The gaze stabilization strategy validated under Processor-In-the-Loop (PIL) and inspired by three biological Oculomotor Reflexes (ORs) enables the aerial robot to lock its gaze onto a fixed target regardless of its roll angle. In addition, the gaze control mechanism allows the robot to perform short range target to target navigation by triggering an automatic fast “target jump” behaviour based on a saccadic eye movement.

Keywords Micro Aerial Vehicle, Automatic Navigation, Gaze Control, Visual Control, Hover Flight, Eye Movement, Oculomotor Control, Biorobotics

Acronyms

FOV: Field Of View.
rVOR: Rotational Vestibulo-Ocular Reflex.
tVOR: Translational Vestibulo-Ocular Reflex.
VFR: Visual Fixation Reflex.
ZSL: Zero-Setting System.
VFL: Visual Feedback Loop.
D-EYE: Decoupled eye robot.
F-EYE: Fixed eye robot.
PIL: Processor-In-the-Loop.
MLD: Maximum Lateral Disturbance.

1. Introduction

Making Unmanned Aerial Vehicles (UAVs) autonomous is likely to be a key research subject for the next 10 years. To achieve this autonomy, one of the most crucial steps is stabilizing the attitude of aerial vehicles. This objective has been classically achieved using an Inertial Measurement Unit (IMU) composed of rate gyros, accelerometers and magnetometers ([1–3]). But IMUs often have to be combined with other sensors to estimate the position of the UAV without drifts. A GPS has been used,
for example, to remove the bias from the pose estimates in outdoor applications ([4–6]). Many strategies involving vision and IMU combined have been adopted for indoor applications because they work efficiently in GPS-denied environments. For example, [7] used an external trajectory device yielding very fast dynamics, which was able to accurately measure both the position and the attitude of a robot equipped with markers. Likewise, [8] used a simple external CCD camera to measure the positions of two LED markers onboard a quadrotor. Although external visual sensing devices of this kind are quite accurate, their visual computational resources often have to be provided by a ground-based host computer. In other studies, the visual sensor was implemented onboard a robot, which was stabilized using two elliptic markers (one placed under the robot and the other in front of it) to determine its attitude and position ([9]). Five markers with a specific geometry were used by [10] to estimate the robot’s yaw and position, while an IMU gave the robot’s attitude and angular velocities. Another particularly cheap solution ([11]) comprised using a Wii remote sensor (Nintendo) to control a hovering vehicle, while the pitch and roll were measured by an IMU. Another solution ([12]) consisted of merging a five DOF estimation provided by a monocular camera with a gravity vector estimation provided by an IMU to achieve take-off, hovering and landing of a quadrotor. Successful control of a hovering quadrotor was achieved by cancelling the linear velocity based on optic flow measurements ([13], [14]). Optic flow was also used to estimate the altitude and the forward speed of an airborne vehicle ([15]). The robot’s attitude has usually been determined in these studies by means of an IMU combined with accelerometers compensating for the drift. However, the authors of several studies have used small or panoramic cameras to determine the robot’s attitude ([7]). Visual servoing was recently applied to stabilize a hovering quadrotor equipped with a fixed camera facing a ground-coloured target ([16]). When this approach was tested under natural outdoor conditions, its vision-based hovering accuracy was found to range between -2m and 2m. In [17], a monocular camera and computer vision with high delays was used to avoid drift obtained with a simple IMU pose estimation. A camera is also used in [18] as a dual-sensor to achieve a drift-free hover and perform obstacles avoidance on a quadrotor.

In our bio-inspired minimalistic framework, a new control strategy was developed, which consists of using only three sensors:

- an optical angular position sensing device with a small Field-Of-View (FOV) of only a few degrees, which is able to locate the angular position of a ground target with great accuracy ([19], [20])
- a low-cost rate gyro
- a simple proprioceptive sensor used to measure the eye orientation in the robot’s frame.

Using a decoupled eye onboard an UAV greatly reduces the computational complexity because a narrower FOV could be used, meaning fewer pixels were needed. In a previous robotic study ([21]), a decoupled eye is classically used as an observation device to track a target and compensate for the UAV displacements around this target.

The latter was successfully tracked by an autonomous helicopter, the position and attitude of which were determined using a combination of GPS and IMU. In the present study, it was proposed to use the robot’s own gaze (i.e., the orientation of the eye in the inertial frame) as a reference value in order to determine its rotations and positions.

An eye with a restricted FOV of only a few degrees (like a kind of fovea1) was adopted here for the following main reasons:

- a small FOV means that very few pixels have to be processed because a small FOV can be obtained by selecting a small Region Of Interest (ROI)
- a small ROI means a high frame rate
- a fovea makes it possible to greatly reduce the computational complexity of the visual processing algorithms, as described in [22]. Therefore that tiny, low-cost, low consumption microcontroller can be implemented onboard the robot.

The robot’s eye was taken to be a visual Position-Sensing Device (PSD), which is able to locate an edge (or a bar) accurately thanks to its small FOV (here, FOV = ±5°, in opposition to a classical FOV of ±25°). This visual sensor was endowed with hyperacuity ([23]), i.e., the ability to locate a contrasting target with a much better resolution than that dictated by the pixel pitch. The implementation of this kind of hyperacute sensor visual sensor was described by previous authors ([19, 20, 24–26]).

The simulated robot and its decoupled eye are described in section 2 along with the nonlinear dynamic model. The Processor-In-the-Loop (PIL) system used to perform the simulations was also described in this section. The original nonlinear observer presented in section 3 estimates the linear speed, the position, the attitude (on the roll axis only) in this paper), and the rate gyro’s bias. Then the gaze controller as well as the whole robot controller based on these estimations is described in detail. In section 4, a new navigation strategy for aerial robots in terms of automatic target to target navigation based on a saccadic flight is outlined. Lastly, the advantages of using a decoupled eye with a gaze control strategy are discussed before the conclusion.

2. System overview

In this part, we will describe first the PIL system and the custom toolchain we developed to perform the robot’s simulation. Then we will present how the biology has inspired our approach and finally we will present the twin-engine robot’s model.

2.1 Processor-In-the-Loop (PIL) simulation

All the simulations presented in this study were performed using a Processor-In-the-Loop (PIL) system. Figure 1 shows the environment and the links existing between the host computer and the embedded processor board. The host computer simulates the robot’s dynamics in real time and monitors the computations performed by the embedded autopilot board which controls the robot.

---

1 The fovea is a part of the eye which is responsible for sharp central vision, useful where visual detail is of primary importance.
As shown in figure 2, the robot consists of a twin-engine aerial robot with two degrees of freedom. This robot was considered to fly at a constant altitude $H$. It is therefore able to rotate freely $\theta_r$ around its roll axis, translate along the horizontal axis $X$ and a third degree of freedom enables the eye to rotate with respect to the robot’s body ($\theta_e$) thanks to a mechanical decoupling between the eye and the body. The eye’s orientation is assumed to be finely controlled by a very fast, accurate servomotor.

As discussed in [27] and [28], the design of our hovering robot and its sensing abilities results from our bio-inspired approach where biology is a source of new ideas for implementing new mechanisms and control strategies. In this sense, our sighted robot with its drifting rate gyro, its proprioception and its visual sensor with a small FOV accounts for key mechanisms observed in flying insects in terms of sensing abilities (see [29]), gaze stabilization (see [30–32]), flight behaviour ([33–36]) and target localization ([37]). The following main similarities between the robot and the fly shown in figure 2 are listed below:

- **A rate gyro**: the fly's eyes are able to locate a contrasting target placed in a small frontal part of its visual field ([33] and [41], [42]). The robot's eye, which is endowed with hyperacuity ([19] and [20]), can locate a contrasting target within its small FOV.
- **An optical position sensing device**: the fly's eyes are able to locate a contrasting target placed in a small frontal part of its visual field ([33] and [41], [42]). The robot's eye, which is endowed with hyperacuity ([19] and [20]), can locate a contrasting target within its small FOV.
- **A neck**: neck movements are essential to the fly, which has no less than 23 pairs of muscles controlling the orientation of its head ([43]). The robot is assumed to be provided with a simpler but nevertheless highly efficient degree of freedom via a fast ($60^\circ$ within 70ms),

```plaintext
2.1.1. The autopilot board : Gumstix Verdex Pro XL6P

In its future version, our robot, a quadrotor, will be equipped with a Gumstix, which is a Linux-based Computer-On-Module (COM). This processor board is equipped with an XScale-PXA270 processor cadenced at 600MHz with 128MB of DRAM and 32MB of Flash memory. This powerful micro computer was chosen to make our robot fully autonomous in terms of computational resources. In addition, the Gumstix can be easily managed via a wifi connection and a console application (such as PuTTY) from the host computer.

2.1.2. Rapid prototyping tool: from Simulink to Gumstix

The novel fast prototyping tool we have developed can be used to automatically and easily download a Simulink model (containing the controllers and the observers in our case) onto the autopilot board. This development toolchain uses a user friendly interface, such as Matlab Simulink, to execute in real-time digital controllers designed in the Simulink environment. This novel automatic code generation tool makes the implementation of a digital control system considerably easier and faster because:
- it generates and compiles a bug-free C code
- it gives block-level access to on-chip peripherals (wifi, serial connections, etc.)
- it can be used to directly design systems using floating-point system design, simulation and scaling procedures thanks to the 32-bit FPU of the Gumstix (it also handles all the fixed-point data).

2.2 Biology as a source for new robot’s abilities

As shown in figure 2, the robot consists of a twin-engine aerial robot with two degrees of freedom. This robot was designed to emulate the future connection between the high-level control board and the low-level control board of the robot.

![Figure 1. Description of the PIL structure. The robot and the ground station are connected by two links. A wireless link (wifi) monitors the control board computations (the Gumstix VERDEX Pro XL6P), emits the Linux commands and the high-level set-points. A serial link to make our robot fully autonomous in terms of computational resources.](image)

**A Novel Strategy for Implementing Saccadic Flight-based Navigation in GPS-denied Environments**

accurate (0.1°) servomotor controlling the orientation of its gaze.

- A proprioceptive sensor located in the neck: a pair of prosternal organs consisting of a pair of mechanosensitive hair fields provide the fly with accurate information about its head orientation ([44]). Along similar lines, the robot is able to measure the orientation of the eye relative to its body by means of a contactless position sensor.

In the first part of this study, the robot was assumed to be hovering above a single stationary ground target. As shown in figure 3 and 4, the robot’s roll (θr) is controlled by applying a differential rotational speed to the propellers. [15], [11] and [8] have established that the altitude control can be completely decoupled from the roll and lateral control in systems belonging to the class of planar Vertical Taking Off and Landing (VTOL) aircraft. It is therefore assumed here that the hovering altitude of the robot is constant and equal to H. To control the position of this underactuated robot, it is necessary to adjust the attitude around the roll axis. The robot’s equilibrium therefore corresponds to a null roll angle, θr = 0°.

Figure 3 gives an overview of the robot’s control architecture, which is based on an observer combined with a state-feedback controller controlling the robot’s attitude and position. Figure 3 presents the mechanical decoupling between the eye and the robot by showing that the robot and its eye receive separate reference input signals (T*, δ* and θr*).

The following notations and angles will be used in what follows:
- θg: angular position of the target relative to the robot’s position in the inertial reference frame.
- θr: the robot’s roll angle.
- θer: eye-in-robot angle, given the gaze direction in the robot’s frame. This angle is mechanically constrained to a maximum value: |θer| < θer MAX.

2.3 Non-linear model of the hovering robot

The nonlinear dynamic model of the robot was developed classically using rigid body equations, and simplified to account for the simple case where the robot can perform only roll rotations and horizontal translations:
The eye’s dynamics are modelled as a first order transfer function cascaded with a rate limiter corresponding to the maximum rotational speed. The mechanical constraints (the angular course) are modelled in terms of the saturation (see the block named ‘Eye’ in figure 4).

For greater clarity, the stabilization of the robot with its decoupled eye was decomposed into two independent problems. Stabilization of the robot’s attitude and position, and the rate gyro’s bias, are now determined on the sole basis of the robot’s measured angular speed $\Omega_r$, the angle of the robot’s gaze $\bar{\theta}_r$, and the retinal error $\varepsilon_r$. Thanks to the control strategy we have implemented based on oculomotor reflexes (see subsection 3.3), the robot hovers above a target by minimizing the retinal error signal.

In this part, we propose explaining how the eye and the hovering controller were designed, and how they interact with each other.

3.1 The nonlinear observer

The nonlinear observer is the cornerstone of the present hovering strategy, as shown in figure 5. By combining the robot’s eye orientation ($\bar{\theta}_r$) with the retinal error ($\varepsilon_r$) measurements, the robot’s position relative to the target ($\bar{\theta}_r$) can be calculated as follows in the robot’s frame:

$$\bar{\theta}_r = \theta_r \mp \varepsilon_r$$ (3)

As the robot’s rotational speed measurement ($\dot{\Omega}_r$) is known, a nonlinear observer can be implemented. This observer is able to deliver at its output an estimation of the linear speed $\dot{V}_x$, the position $\hat{X}$, the roll angle $\hat{\theta}_r$, and the rate gyro’s bias $\Delta \Omega_r$:

$$\dot{\hat{X}} = V_x$$

$$\dot{\hat{\theta}_r} = \Omega_r$$

$$\dot{\hat{V}_x} = -\frac{(F_1 + F_2) \sin(\bar{\theta}_r)}{m} - K_{xy} V_x$$

The nonlinear observer is the cornerstone of the present hovering strategy, as shown in figure 5. By combining the robot’s eye orientation ($\bar{\theta}_r$) with the retinal error ($\varepsilon_r$) measurements, the robot’s position relative to the target ($\bar{\theta}_r$) can be calculated as follows in the robot’s frame:

$$\bar{\theta}_r = \theta_r \mp \varepsilon_r$$ (3)

As the robot’s rotational speed measurement ($\dot{\Omega}_r$) is known, a nonlinear observer can be implemented. This observer is able to deliver at its output an estimation of the linear speed $\dot{V}_x$, the position $\hat{X}$, the roll angle $\hat{\theta}_r$, and the rate gyro’s bias $\Delta \Omega_r$:

$$\dot{\hat{X}} = V_x$$

$$\dot{\hat{\theta}_r} = \Omega_r$$

$$\dot{\hat{V}_x} = -\frac{(F_1 + F_2) \sin(\bar{\theta}_r)}{m} - K_{xy} V_x$$
\[
\begin{pmatrix}
\dot{V}_r \\
\dot{X} \\
\dot{\theta}_r \\
\dot{\Delta}_g
\end{pmatrix} =
\begin{pmatrix}
-\frac{T}{m} \sin(\hat{\theta}_r) - K_{Vx} \dot{V}_x + L_1 \dot{\theta}_r \\
V_x + L_2 \dot{\theta}_r \\
\Omega_r - \Delta_\theta + L_3 \dot{\theta}_r \\
L_4 \dot{\theta}_r
\end{pmatrix} + \begin{pmatrix}
\theta^*_{\text{r,x}} \\
\mathrm{tVOR} \\
\mathrm{VFR}
\end{pmatrix} \theta^*_{\text{r,x}}
\]

(4)

Remark 2: The value of \( \hat{\theta}_r \) in equation (4) is obtained from equation (3). \( \Omega_r - \Delta_\theta \) is therefore simply the modelling term accounting for the rate gyro’s bias and \( L_3 \dot{\theta}_r \) is the innovative term. As mentioned above, the rate gyro’s bias is determined on the sole basis of the target angle error \( \dot{\Delta}_g = L_4 \dot{\theta}_r \).

3.3 Eye controller

The eye’s controller keeps the gaze locked onto a target placed on the ground. Since the field of view is very small, the closed loop gaze system has to act very fast and efficiently. The requisite efficiency is achieved using a bio-inspired approach, involving a combination of three complementary oculomotor reflexes:

- A rotational vestibulo-ocular reflex called the rVOR, which has to counteract the effects of all the rotations performed by the robot. This reflex yields the signal \( \theta^*_{\text{r,x}, \text{r}} \), which is simply the opposite of the roll angle \( \hat{\theta}_r \).
- A translational vestibulo-ocular reflex called the tVOR compensates for the effects of any robot’s translations in the retina. The output signal \( \theta^*_{\text{r,x}, \text{t}} \) is based on the robot’s position \( X \) and its altitude \( H \), contributing to keeping the eye locked onto the target.
- A visual fixation reflex called the VFR, consists of a visual feedback loop with which any retinal signal errors \( \epsilon_r \) are cancelled by controlling the eye’s orientation \( \theta_{\text{r,x}} \) via the control input signals \( \theta^*_{\text{r,x}, \text{VFR}} \) (see figure 5).

The VFR reflex plays a key role here. It is the main reflex serving to track the target under all conditions. Thanks to this reflex, the robot is able to:

- hover “robustly” above a target,
- reject any lateral disturbances resulting from gusts of wind by quickly correcting the eye’s orientation, which is not possible if the robot has no decoupling, because of its relatively large inertia.

The VFR controller consists of a simple proportional integral controller, which keeps the retinal error close to 0°. This controller yields a reference angle \( \theta^*_{\text{r,x}, \text{VFR}} \) in order to contribute to the eye’s orientation, see equation (7). In short, this reflex is responsible for target tracking in every situation, which is the first step towards hovering robustly and accurately determining the robot’s attitude and position.

The VOR is composed of a combination of two reflexes keeping the target in the FOV even if the robot is moving. VORs, which were directly inspired by the visual processes occurring in insects, compensate for two kinds of robot movements:

- roll: the rVOR uses the roll angle estimation \( \hat{\theta}_r \) provided by the nonlinear observer to compensate for any body rotations \( \theta^*_{\text{r,x}, \text{r}} = -\hat{\theta}_r \).
- lateral translation: the tVOR, which is based on the estimation of the robot’s linear position, minimizes the effects of any lateral displacements on the retinal error \( \epsilon_r \), \( \theta^*_{\text{r,x}, \text{t}} = -\arctan \frac{X}{H} \).

\[
\theta^*_{\text{r,x}} = \theta^*_{\text{r,x}, \text{r}} + \theta^*_{\text{r,x}, \text{t}} + \theta^*_{\text{r,x}, \text{VFR}}
\]

(7)

The reference angle \( \theta^*_{\text{r,x}} \) (see figure 5), which therefore results from the joint contribution of three reflexes (rVOR, tVOR and VFR), can be expressed as follows:
The VORs were used as feedforward control signals cancelling the movements perceived by the robot (thanks to the gyro and the eye-in-robot angle). Additionally, the VFR is used here as a feedback control signal cancelling any undetected movements and the modelling errors present in the feedforward terms. Simulation results and discussions about the efficiency of these reflexes could be found in our previous works ([52] and [53]).

3.4 Rotational speed controller of the propellers

The rotational speed of the robot is used by the attitude and position controller as an input control signal to stabilize the robot and achieve the desired position (figure 4). In this way, the rotational speed inner loop makes the robot faithfully adopt the speed rotation set-points ($\Omega^\star_r$) imposed by the attitude-position controller (see subsection 3.5). It was assumed here that the propellers are driven directly by the thrust, which can be deduced from the propeller’s rotational speed in line with equation (8). The rotational speed is therefore controlled by a simple differential thrust between the two propellers.

$$F_i = c_T \omega_i^2$$  

(8)

Where $F_i$ is the thrust generated by the propeller $i$, $\omega_i$ is the rotational speed of the propeller $i$, and $c_T$ is the thrust coefficient, which can be identified using static thrust tests.

In order to account more satisfactorily for the rate gyro’s dynamics, a classical LQG controller-observer structure was chosen instead of a simple PID. The state space representation for the rotational speed loop, including the gyro dynamics, can therefore be written:

$$
\begin{pmatrix}
\dot{\Omega}_r \\
\dot{\Omega}_e
\end{pmatrix} = 
\begin{pmatrix}
\frac{-2L}{\tau_{	ext{rot}} \tau_{	ext{gyr}}} & 0 \\
\frac{1}{\tau_{	ext{rot}}} & \frac{-1}{\tau_{	ext{gyr}}}
\end{pmatrix}
\begin{pmatrix}
\Omega_r \\
\Omega_e
\end{pmatrix}
+ 
\begin{pmatrix}
\frac{2L}{\tau_{	ext{rot}} \tau_{	ext{gyr}}} \\
0
\end{pmatrix}
\delta^\star
$$  

(9)

Where $\Omega_r$ is the actual roll rotational speed and $\Omega_e$ is the rotational speed measured by the rate gyro.

The closed loop time response obtained for $\Omega_r$ was less than $20 ms$, and the differential control noise was less than $2\%$.

3.5 Position and attitude controller

The position and attitude controller were implemented using LQR methods and the estimated states ($\dot{V}_x$, $\dot{X}$ and $\dot{\theta}_y$) given by the nonlinear observer. When designing this controller, it was assumed that the rotational speed feedback loop described above (see subsection 3.4) is much faster than the attitude-position loop.

The LQR state feedback was designed for the system (1) linearized around the origin with the equilibrium control input $\Omega_{req} = 0$. To cancel the steady state error, an integral term was added to the position error.

4. New outlooks provided by the hovering by gazing control strategy

In previous studies carried out at our laboratory ([52] and [53]), we demonstrated that a robot equipped with a decoupled eye (denoted D-EYE) was able to reject lateral and rotational disturbances better than the same robot with a fixed eye (denoted F-EYE) even if the field of view of the decoupled eye robot is smaller.

The implementation of a decoupled eye onboard an aerial vehicle also opens up new perspectives:

- it enables a robot to finely control its horizontal position with respect to the target without being given any explicit information about its position by an external camera, a motion capture system, or a GPS as occurs in more classical systems,
- it enables a robot to perform shifted hovering with respect to the target. This point has interesting consequences regarding the robot’s disturbance rejection ability. More generally, the robot is therefore not constrained to stay in a small restricted area around the target (see subsection 4.1),

Figure 5. Block-diagram of the controller. The nonlinear observer determines the attitude of the robot on the sole basis of the angular speed and eye orientation measurements. The eye controller is based on three Oculomotor Reflexes (ORs), which keep the eye locked on the target regardless of the translational or rotational disturbances that may occur. Measured signals are presented in red and control input signals are presented in green. Note that the set-point $\mathbf{X}^\star$, which is explained in subsection 4.2, can be used to make a change of target of interest.
FOV = ± 25°

\[ \text{MLD} = H \tan \left( \frac{\text{FOV}}{2} \right) \]

4.1 Shift hovering

The following discussion demonstrates that the decoupling allows for hovering not only above a target but also near a target. Let us start by defining the Maximum Lateral Disturbance (MLD):

**Definition 1:** The maximum lateral disturbance (denoted MLD) corresponds to the maximum range of horizontal displacement which can occur without losing sight of the target (i.e., without the target ever leaving the FOV). In other words, the MLD corresponds to the robot’s maximum lateral displacement which can arise between two consecutive updates of the visual control loop (i.e., during the sample time of the VFR). If a disturbance is greater than the MLD, the visual feedback loop’s sampling rate will be too slow to detect this movement and the robot will not be able to reject this disturbance. The MLD is expressed in metres, and MLD<sub>R</sub> and MLD<sub>L</sub> correspond to the right and left MLD, respectively.

Figure 6 shows the ability of the F-EYE robot with an FOV of ±25° and that of the D-EYE robot with a 5-fold smaller FOV (±5°) to reject lateral disturbances. From this figure, it is possible to determine the MLD corresponding to each robot (F-EYE robot and D-EYE robot):

**F-EYE system:** the robot’s possible positions are restricted to \( |X| \leq H \tan \left( \frac{\text{FOV}}{2} \right) \):

\[
\begin{align*}
\text{MLD}_R(X) &= H \tan \left( \frac{\text{FOV}}{2} \right) - X \\
\text{MLD}_L(X) &= H \tan \left( \frac{\text{FOV}}{2} \right) + X
\end{align*}
\]

**D-EYE system:** theoretically, the robot can adopt any position on the X axis.

\[
\begin{align*}
\text{MLD}_R(X) &= H \tan \left( \frac{\text{FOV}}{2} + \arctan \left( \frac{X}{H} \right) \right) - X \\
\text{MLD}_L(X) &= X - H \tan \left( \frac{\text{FOV}}{2} + \arctan \left( \frac{X}{H} \right) \right)
\end{align*}
\]

When hovering perpendicularly above the target, the F-EYE robot benefits from its large FOV and can reject a 93-centimetre MLD, whereas the F-EYE robot can cope with an MLD of only 17.5 centimetres due to its smaller FOV. However, by shifting the robot’s horizontal position away from the target, it is possible to greatly increase the D-EYE robot’s MLD rejection capacity, which becomes larger than that of the F-EYE robot after simply making a 2-m lateral shift. This simple strategy therefore makes it possible to overcome the problem focusing on the trade-off between the width of the FOV and the MLD. Even with its 5-fold smaller FOV, the D-EYE robot can reject much larger MLDs than the F-EYE robot. In addition, unlike the F-EYE, the decoupled eye enables the D-EYE robot to reject disturbances of practically the same amplitude on both sides (right and left).

One of the main advantages of the novel strategy consisting of shifting the robot laterally with respect to the target focuses on the Maximum Lateral Disturbances (MLD) that can be rejected. From figure 6, it can be seen that the F-EYE robot with a ±25° FOV cannot even achieve a 2-m shift without losing sight of the target. Unlike the F-EYE robot, the D-EYE robot with a ±5° FOV decoupled eye hovering at 2m and offset by 2m from the target can reject both right and left lateral disturbances of up to 38cm while keeping a positioning accuracy of more than 1cm.

4.2 Saccadic flight behaviour

A last useful advantage of the decoupled eye is its ability to make a change of target. For example, when the robot hovers above a first target, a very quick eye rotation (similar to a catch-up saccade) is generated to make the robot’s eye “jump” to a second target, the position of which is known. A catch-up saccade is one of the four types of eye movement (saccadic, smooth pursuit, vestibulo-ocular reflex and vergence) used to track a trajectory. Catch-up saccades occur in humans when the smooth pursuit system is not sufficiently fast to track a moving target [54].
In this study, this mechanism was used first to make the robot shift from one target to another. Like the steering by gazing strategy ([25]), the aerial robot will move toward the target at which its gaze is looking.

4.2.1. Notations and implications:

Here we define some notations used to distinguish between the variables used in the various reference frames: \( \frac{R}{I} \) describes the variable \( U \) from \( A \) (the object in terms of which the vector is expressed) in the frame of reference \( B \) (in which the measurements are made.)

For example, \( \frac{R}{I}X \) gives the actual position of the robot \( R \) in the inertial frame \( I \) and \( \frac{R}{I}X^* \) denotes the reference position of the robot \( R \) relative to the target frame \( T \).

Three frames of reference were therefore used in this study:
- the Robot’s frame of reference, denoted \( R \),
- the Inertial frame, denoted \( I \),
- the Target’s frame, denoted \( T \).

Remark 1: Note that in the previous part, the target frame \( T \) was the same as the inertial frame \( I \) because the two origins were the same, i.e.,: \( \frac{T}{I}X = X_T \) and \( \frac{T}{I}X = 0 \).

The relation between the estimated robot’s position in the inertial frame, the estimated robot’s position in the target’s frame and the target’s position in the inertial frame can be written:

\[
\frac{R}{I}X = \frac{R}{T}X + \frac{T}{I}X
\]

(12)

Where \( \frac{T}{I}X \) is the position of the current target in the inertial frame. The observer described by equation (4) therefore becomes:

\[
\begin{pmatrix}
\frac{R}{I}X \\
\frac{R}{I}X^* \\
\frac{R}{I}A^*
\end{pmatrix} =
\begin{pmatrix}
\frac{\Delta}{I} \sin(\frac{\Delta}{T} \theta) - K_v \frac{\Delta}{I} X + L_v \frac{\Delta}{I} \theta \\
\frac{\Delta}{I} \Omega - \frac{\Delta}{A} X + L_{\Omega} \frac{\Delta}{I} \theta \\
\frac{\Delta}{I} \theta
\end{pmatrix}
\]

(13)

\[
\frac{\Delta}{I} \theta = \frac{\Delta}{T} \theta - \frac{\Delta}{I} \theta
\]

4.2.2. “Target Jump” procedure:

To make the robot shift from one target to another, a simple change of variable suffices. The previous value of the target’s position is then replaced by the new one. The coordinates of the new target just need to be updated as follows:

\[
\frac{T}{I}X(t) = \frac{T}{I}X_k \quad \text{before jump.} \quad (14)
\]

\[
\frac{T}{I}X(t + \Delta t) = \frac{T}{I}X_{k+1} \quad \text{after jump.} \quad (15)
\]

Where \( \Delta t \) corresponds to the nonlinear observer’s sampling time and \( \frac{T}{I}X_k \) with the subscript \( k \) or \( k + 1 \) corresponds to the target’s identifier. \( \frac{T}{I}X_k \) and \( \frac{T}{I}X_{k+1} \) are therefore the position of the target \( k \) before the “target jump” and the position of target \( k + 1 \) after the “target jump”, respectively.

The eye’s “jump” (or saccade) is then expressed by changing the value of the feedforward term \( \theta_{er}^* \), which is written with the new notation: \( \theta_{er}^* = -\arctan \left( \frac{\frac{T}{I}X}{\frac{T}{I}X^*} \right) \).

When the operator wants to make the robot change target, he/she just needs to send the robot the new target coordinates, expressed in the inertial frame (\( \frac{T}{I}X_k \)).

Once the target’s new coordinates have been specified, the robot immediately changes its eye orientation and re-positions itself appropriately with respect to the reference \( \frac{T}{I}X^* \). A simulation of the robot’s target-jumping behaviour is illustrated in the figures 7, 8 and 9.

Remark 2: These simulations were performed with the Processor-In-the-Loop (PIL) environment described in the section 2.1, with the set of parameters given in table 1. The whole control strategy presented in section 3 was developed under Matlab/Simulink and the computations were performed at a frequency of 400Hz (except for the VFR which was computed at only 40 Hz by considering a realistic refresh rate of the visual sensor). This computations corresponded to less than 20% of the whole CPU load. All the calculations and control inputs were monitored at the same frequency in real time via the wifi connection before being plotted and compared with the “real” values simulated on the host computer.

The following scenario was simulated in figures 7 and 9:
- step 1: the robot made a 1-m lateral shift with respect to the target
- step 2: after 5s, the robot jumped to target 2 and the previous lateral shift with respect to the target was maintained (the figure 8 shows only the eye jump occurring at \( t = 5.0s \))
- step 3: after 5s, the robot jumped to target 3

We therefore have: \( \frac{T}{I}X_1 = 0m, \frac{T}{I}X_2 = -1m \) and \( \frac{T}{I}X_3 = -2m \).

In conclusion, thanks to the fast eye dynamics, the “eye jump” is performed very quickly because the new target is locked by the robot’s eye in less than 0.13s, as shown in figure 8. Once this new target is locked, the robot’s body automatically reaches the desired relative to target position \( \frac{T}{I}X^* \) in less than 1.5s, as shown in figures 7 and 9.

5. Advantages of the decoupled eye

Here we summarize the advantages of having a decoupled eye with a small FOV:

- **Less distortion:** having a large FOV necessarily implies to use a fish-eye lens and more pixels to keep a good angular resolution. The fish-eye lens induces a lot of distortions in the peripheral region of the image which need to be compensated. Finally, the use of a large FOV implies taking into account more pixels and cancelling the lens distortions, which leads to more computations. However, a decoupled eye with a small FOV allows for
The updating of the target position described in equation (15) involves a jump in the feedforward term \( \theta^e_{I+1} \) (red curve), which triggers the change of target. Retinal error during the target jumping phase. It takes only 0.13s for the eye to lock the new target in its narrow field of view. Therefore, the new target enters into the FOV 0.13s after the target change was asked. Figure 8 shows more precisely the retinal error signals during a target jumping.

\[
\begin{align*}
\theta^e & = \text{Retinal error during target jumping} \\
\theta^e_{I+1} & = \text{Jump in FOV responsible of FOV limit}
\end{align*}
\]

Figure 7. a) Estimation of the position and the corresponding tracking expressed in the different frames of reference. b) Evolution of the eye angle. The updating of the target position described in equation (15) involves a jump in the feedforward term \( \theta^e_{I+1} \) (red curve), which triggers the change of target. Retinal error during the target jumping phase. It takes only 0.13s for the eye to lock the new target in its narrow field of view. Therefore, the new target enters into the FOV 0.13s after the target change was asked. Figure 8 shows more precisely the retinal error signals during a target jumping.

Figure 8. Evolution of retinal error during a target jumping. The blue line describes the actual retinal error, and the red one corresponds to the measured retinal error which includes the sensor quantification, discretization, dynamics, internal delay and the eye’s limited FOV (see table 2). Here, the new target is locked in less than 0.13s and is kept locked into the FOV. In addition, the retinal error is kept under 1.4° in less than 0.15s (i.e., \( |\epsilon| < 1.4° \)) and under 0.2° in less than 0.7s. avoiding the lens distortions and reducing the number of pixels by maintaining the target at the centre of the FOV where the acuity is the best.

- **Reduced computational complexity**: as shown in section 4.2.2 of [53], the D-EYE robot rejects disturbances more efficiently with its two pixels than does a F-EYE robot featuring a large FOV having 1000 pixels (a classical camera with 1000 pixels has the same 0.1°-angular resolution as the D-EYE sensor endowed with hyperacuity). The small number of pixels of a decoupled eye reduces drastically the computational resources required to extract the angular position of the target.

- **Faster disturbance rejection**: with the F-EYE robot (equipped with a small FOV), a lateral disturbance resulting from a wind gust cannot be quickly counteracted. To compensate sharply for a lateral disturbance, the robot has to take a stiff roll angle, which results in the loss of the target; whereas it is possible with the D-EYE system to adopt a stiff attitude without losing sight of the target (see section IV of [52]).

- **Robustness with respect to lateral wind**: in the same way, the decoupled eye enables the robot to hover not only above a target but next to a target. With this simple method the ground area covered by the FOV is extended, which increases the Maximum Lateral Disturbance (MLD) which can be rejected. In addition, the robot is able to hover relatively far from the target without losing sight of this target (which is not possible with a robot having a fixed eye, as shown figure 6).

- **More than hovering**: Lastly, the decoupled eye enables the robot to change target and move above the next target, as explained in subsection 4.2. This makes it possible to impose a specific path of travel to the robot.

**GPS and gaze strategy**: if a GPS signal is available, it would be possible to communicate the absolute position of a moving target. Our strategy could be used efficiently to track a moving target (these results are not presented in the paper but the D-EYE robot might be able to track a 1.5-\( m/s \) moving target.) It is then possible to communicate the absolute position of the target as the sum of the robot’s GPS position (low-pass filtered) and the relative to target robot’s position estimated thanks to the gaze strategy:

\[
T_X = \frac{R}{I} X_{GPS} + \frac{R}{I} X_{gaze} = \frac{R}{I} X_{GPS} - \frac{R}{I} X_{gaze} \quad (16)
\]

Where \( \frac{R}{I} X_{GPS} \) is the filtered GPS position and \( \frac{R}{I} X_{gaze} \) is the estimated position of the robot with respect to the target.
controlling the 3-D position of a robot by anchoring its gaze onto particular objects of interest. In a future study, the method presented here will be implemented onboard a small fully autonomous quadrotor with six degrees of freedom.

7. Acknowledgements

The authors would like Jessica Blanc for correcting the English manuscript. This work was supported by CNRS, Aix-Marseille University and the French National Research Agency (ANR) with the EVA, IRIS and Equipex/Robotex projects (EVA project and IRIS project under ANR grant numbers ANR608-CORD-007-04 and ANR-12-INSE-0009, respectively).

8. References


