Vector Field Aided Trajectory Tracking by a 10-gram Flapping-Wing Micro Aerial Vehicle
Abdoullah Ndoye, José de Jesus Castillo-Zamora, Sabrine Samorah Laki, Romain Miot, Edwin van Ruymbeke, Franck Ruffier

To cite this version:
Abdoullah Ndoye, José de Jesus Castillo-Zamora, Sabrine Samorah Laki, Romain Miot, Edwin van Ruymbeke, et al.. Vector Field Aided Trajectory Tracking by a 10-gram Flapping-Wing Micro Aerial Vehicle. IEEE International Conference on Robotics and Automation (ICRA 2023), IEEE, May 2023, Londres, United Kingdom. hal-03831115v2

HAL Id: hal-03831115
https://hal.science/hal-03831115v2
Submitted on 18 Mar 2023

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

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

Distributed under a Creative Commons Attribution 4.0 International License
Vector Field Aided Trajectory Tracking by a 10-gram Flapping-Wing Micro Aerial Vehicle

A. Ndoye$^1$,*, J. J. Castillo-Zamora$^1$, S. Samorah-Laki$^1$, R. Miot$^2$, E. Van Ruymbeke $^2$ and F. Ruffier$^1$

Abstract—Here we describe how a 10-gram Flapping-Wing Micro Aerial Vehicle (FWMAV) was able to perform an automatic trajectory tracking task based on a vector field method. In this study, the desired heading was provided by a vector field which was computed depending on the desired trajectory. The FWMAV’s heading was changed by a rear steering mechanism. This rear mechanism simultaneously (i) tenses one wing and relaxes the opposite wing, and (ii) moves the rudder in the same direction as the wing is relaxed. Due to the complex dynamics, system identification methods were used to identify simple linear models using a set of dedicated free flight tests. This yaw and roll simple models help to adjust the yaw controller and the inner loop roll controller. The experimental results obtained here show that a time-independent vector field-based strategy is robust to various initial position and/or speed conditions. The task of tracking circular and 8-shaped trajectories was accomplished successfully over tens of meters.

I. INTRODUCTION

During the last decade, Flapping-Wing Aerial Vehicles (FWA Vs), which have also been called ornithopters, have been attracting considerable attention. Recent studies have focused on improving the agility of these vehicles [1–4] and controlling their height of flight [5,6], thus enhancing their performances. Nowadays, FWA Vs are able to perform difficult tasks such as obstacle avoidance [7–9], fixed-object tracking [10] and trajectory tracking tasks [11–13]. Studies on micro FW AVs (called also Flapping-Wing Micro Aerial Vehicles FWMAVs) are quite rare [11].

In parallel, remote-controlled two-wing ornithopters weighing only 10 grams have been available on the market since 2010 [14,15] and sold as crash-resistant toys by the company XTIM-BionicBird. The successive versions of these RC (Radio-controlled) models based on the same propulsion and directional mechanisms have been named Avitron, BionicBird, and more recently, Metafly.

The Metafly (shown in Fig. 1) is endowed with 2 DC motors serving as its actuators: (i) the main DC motor is attached to a 2-stage planetary gearbox controlling the flapping frequency [14], and (ii) the second DC motor concomitantly tenses one wing and relaxes the opposite wing (in a differential manner) while adjusting the rudder to control the course [15]. The speed of the Metafly is poorly controllable, however, as its flapping rate affects both its altitude and its speed. As regards the agility of FWMAVs, it has been suggested that relying solely on a rudder to change the direction of FWMAVs may be insufficient as this does not generate enough force required to perform agile maneuvers, even using bang-bang control methods [16]. One commonly adopted solution here is the so-called banked turn in which roll motion is used to generate a change of direction [3,17] by means of dihedral wings [18,19], asymmetrical morphing of the wing surface [4], differential wing spoiler steering [20], or the generation of an asymmetric flapping amplitude [2,21], among other solutions.

Besides, the dynamics and kinematics of FWA Vs are so complex that a large number of dynamic models have been presented in the literature [6,22–25]. Due to this complexity, the authors of most previous studies consider fixed-wing models, without taking the intra-flapping processes [26] or the heading actuator dynamics [27] into account. In these cases, the results of grey-box [28,29] or black-box [12,30] system identification have been used to model FWAVs’ dynamics. Kim et al. [11,31] have succeeded in flying 4 ornithopters in formation along a circular trajectory. The cohesion of this formation was highly dependent, however, on specific initial conditions and on the fine tuning of these ornithopters, as the authors themselves mentioned [11]. As vector field-based trajectory tracking is time-independent [32–37], this strategy can be used to overcome the poor controllability of the Metafly’s flight speed when tracking various trajectories, as established in the present study. The present approach enables the vehicle (i) to track the trajectory robustly, regardless of the initial conditions, (ii) to follow complex trajectories, and (iii) to change its trajectory while in flight. The main contribution of this research can be summarized under the following three

---

$^1$ Aix Marseille Université, CNRS, ISM, Marseille, France
$^2$ XTIM - Bionic Bird, 13015 Marseille, France, www.bionicbird.com
* Corresponding author: abdoullah.ndoye@grenoble-inp.fr
The identification of simplified yaw and roll linear models for the Metafly based on enriched free flight data.

A novel 3-state (full left, neutral, full right) directional control system whereby the directional control is carried out by tensing and relaxing the wings while positioning the rudder appropriately. The commands of tensioning/relaxing the wings and of actuating the rudder are mixed (Fig. 2). The amount of time $t_{\text{deflection}}$ the actuator remains full left or full right serves as the control input.

This is the first implementation of a vector field method on a FWMAV (as far as we know) in order to enable it to time-independently track a trajectory, while enlarging the range of possible initial conditions.

The modified Metafly FWMAV and the method used to control its heading are presented in detail in Section II. The system identification procedure, which was performed using real flight data and Matlab processing tools, is described, along with the simplified model, in Section III. The vector field strategy used to track various trajectories successfully is presented in Section IV. The results of various experimental tests validating the present vector field-based trajectory tracking approach are presented in Section V. Lastly, the authors’ concluding comments and suggestions for future studies are presented in Section VI.

II. THE METAFLY FWMAV AND ITS HEADING CONTROL

In the modified version of the Metafly FWMAV used here, a Deltang RX43d receiver controlling the two actuators was embedded, which adjusted the flapping frequency and actuated the directional mechanism, respectively. A 3.7V/58mAh Lipo battery was used as the power source. A list of the Metafly’s characteristics is given in Table I.

The heading control mechanism (described in Fig. 2) makes it possible to change the heading by initiating a banked turn (as depicted in Fig. 3). For example, in order to perform a left banked turn as in Fig. 3, the heading control mechanism tenses the right wing and relaxes the left wing while positioning the rudder to the left. This creates a negative rolling motion inducing a positive yaw moment. At the same time, there is also a considerable difference in the drag between the left and the right wings (because the right wing is tensed and the left wing relaxed), which also results in a positive yaw moment in the same direction as the turn we have referred to as the proverse yaw moment (Fig. 3). During the rolling motion, as the left wing rolls down (see insert in Fig. 3), an additional wind from below is created. This additional wind is added to the velocity-induced wind to give the total upward-oriented wind (grey arrows in Fig. 3). The resulting opposite deflection of the lift (backward and forward) creates a negative yaw moment called the adverse yaw (see red arrows in Fig. 3). The adverse yaw is transient as it due to the rolling motion. The

### Table I

<table>
<thead>
<tr>
<th>Parameter/Property</th>
<th>Nominal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>10 [g]</td>
</tr>
<tr>
<td>Length</td>
<td>190 [mm]</td>
</tr>
<tr>
<td>Wingspan</td>
<td>290 [mm]</td>
</tr>
<tr>
<td>Wing chord</td>
<td>85 [mm]</td>
</tr>
<tr>
<td>Wing amplitude</td>
<td>$\pm 55^\circ$</td>
</tr>
<tr>
<td>Flapping frequency</td>
<td>10 – 20 [Hz]</td>
</tr>
<tr>
<td>Flight endurance</td>
<td>8 [min]</td>
</tr>
</tbody>
</table>
rudder also creates a small amount of positive yaw moment in the same direction as the yaw induced by the rolling motion and the proverse yaw, which causes it to turn appropriately, making the effect of the adverse yaw negligible.

III. METAFLY MODELING FOR THE HEADING CONTROL

The identification of the Metafly dynamics between the directional command and the yaw (heading) was performed. As mentioned above, the Metafly must perform a roll motion in order to change its heading. If the roll angle $\Phi$ exceeds 35°, the turn is too tight and we observed that the Metafly will dive and crash. To overcome this problem, the roll motion is controlled and capped in an inner loop. Two transfer functions were identified: (i) $H_{\text{roll}/\text{deflection}}(s)$ which relates the roll $\Phi(s)$ to the control input $t_{\text{deflection}}(s)$, and (ii) $H_{\text{yaw}/\text{roll}}(s)$, which establishes the relation between the yaw $\Psi(s)$ and the roll $\Phi(s)$, so that:

$$H_{\text{roll}/\text{deflection}}(s) = \frac{\phi(s)}{t_{\text{deflection}}(s)} \quad \text{and} \quad H_{\text{yaw}/\text{roll}}(s) = \frac{\psi(s)}{\phi(s)} \quad (1)$$

A. Experimental Setup and Flight Tests

The experiments required for the identification took place in the flight arena. The effective volume in which the 10-gram Metafly can be tracked at an acquisition frequency of 500Hz by the Vicon motion capture system (17 cameras) was $5\times6\times3$ m$^3$ (lxLxH). For this purpose, a set of adhesive or spherical reflective markers was suitably placed over the Metafly. The Metafly were controlled by an external workstation (equipped with Matlab and ROS), which delivered the 2.4GHz DSMX signals to the RX43D receiver embedded on the Metafly (via an Ethernet-to-DSMX bridge, see Fig. 4).

Several identification flight tests were performed in order to identify the models. In these experiments, the PID controller:

$$C_z(s) = \frac{u_{\text{flapping}}}{e_z} = k_d s + k_p + \frac{k_i}{s} = 1000 s + 200 + \frac{0.1}{s} \quad (2)$$

was previously implemented to make the Metafly hold its altitude constant with respect to the ground by adapting its flapping frequency. The gains were manually tuned and the command $u_{\text{flapping}}$ was sent directly to the onboard RX43D receiver that will actuate the wing-flapping mechanism. The latter version led us to take the flapping frequency as the only control input used to stabilize the altitude of the Metafly and to use the heading control mechanism (Fig. 2) as the only actuator of the rotational motion. Nonetheless, as it was not possible to adjust the angle $\delta R$ throughout the predefined range, we adopted the 3-state turning command strategy presented in Fig 2, thus taking the time $t_{\text{deflection}}$ as the actual rotational control input. Note that the directional control (full left, neutral, or full right) was defined by the sign of $t_{\text{deflection}}$, so that a negative value allowed to turn right, and a positive value allowed to turn left. As shown in Fig. 5, the $t_{\text{deflection}}$, roll and yaw data were recorded and then used in the Matlab System Identification Toolbox to retrieve the models.

B. Identification of the model

Based on the data recorded during 4 different free flight tests, the functions $H_{\text{roll}/\text{deflection}}(s)$ and $H_{\text{yaw}/\text{roll}}(s)$ were identified, using the Gauss-Newton search method, as follows:

$$H_{\text{roll}/\text{deflection}}(s) = -\frac{0.4536 s + 0.09858}{s^2 + 0.7815 s + 0.1387} \quad (3)$$
$$H_{\text{yaw}/\text{roll}}(s) = -\frac{7.455 s + 8.179}{s^2 + 1.493 s + 0.01036} \quad (4)$$

The transfer functions were identified with a given fit percentage of 78% and 86% for $H_{\text{roll}/\text{deflection}}(s)$ and $H_{\text{yaw}/\text{roll}}(s)$,
were used in the simulations performed for controller tuning purposes, which are presented in the following section.

IV. VECTOR FIELD-BASED TRAJECTORY TRACKING

A. The Vector Field theory

A vector field assigned a vector to each 2D coordinate in the \((x, y)\) plane with a direction enabling it to converge with a desired path [34]. Let us consider the \((x, y)\) position of the robot, \(\xi = [x \ y]^T \in \mathbb{R}^2\) and recall the orientation of the robot \(\Psi_{\text{meas}} \in \mathbb{R}\) [34], so that its velocity can be written as follows:

\[
\dot{\xi} = v_{\text{flight}} \begin{bmatrix} \cos(\Psi_{\text{meas}}) \\ \sin(\Psi_{\text{meas}}) \end{bmatrix} = v_{\text{flight}} \chi_u \in \mathbb{R}^2
\]  

(5)

where \(v_{\text{flight}} \in \mathbb{R}\) is the flight speed (taken for instance to be constant, since the flapping frequency was fixed by the altitude controller) and \(\chi_u \in \mathbb{R}^2\) is the directional unit vector. In addition, one can define a curvilinear desired path by the implicit equation:

\[
\mathcal{P} = \{ \xi_d : \phi(\xi_d) = 0 \} \subset \mathbb{R}^2
\]  

(6)

so that the function \(\phi: \mathbb{R}^2 \to \mathbb{R}\) is \(C^2\)-smooth. The geometric curve \(\mathcal{P}\) can be written in several implicit forms. The main restriction imposed on the curvilinear path is regularity, i.e. in the vicinity of \(\mathcal{P}\), one has:

\[
\| \nabla \phi(\xi_d) \| = \left\| \begin{bmatrix} \frac{\partial \phi(\xi_d)}{\partial x_d} \\ \frac{\partial \phi(\xi_d)}{\partial y_d} \end{bmatrix} \right\| \neq 0
\]  

(7)

where \(\| \cdot \|\) stands for the Euclidean norm.

The tracking error \(e_{\text{track}}\) is defined by a strictly increasing \(C^1\)-function \(\varepsilon(\cdot)\) so that:

\[
e_{\text{track}}(\xi) = \varepsilon[\phi(\xi)] \in \mathbb{R}
\]  

(8)

The choice of \(\varepsilon(\cdot)\) is a free parameter. By definition, \(e_{\text{track}} = 0\) if and only if \(\xi \in \mathcal{P}\). Thus, according to [38], the vector field \(\chi\) can be simulated by the expression:

\[
\chi(\xi_d) = (I - k_n e_{\text{track}}(\xi_d) I_2) \nabla \phi(\xi_d)
\]  

(9)

with \(k_n = \text{const} > 0\),

\[
\Gamma = \begin{bmatrix} 0 & \gamma \\ -\gamma & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2} \quad \text{and} \quad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 2}
\]  

(10)

Note that \(\gamma \in \{-1, 1\}\) determines the direction in which \(\mathcal{P}\) will be tracked. Eq. (9) was therefore used to find the desired directional unit vector field \(\chi_{\text{desired}}\), and consequently, \(\Psi_{\text{desired}}\) as follows:

\[
\chi_{\text{desired}} = \frac{\chi(\xi_d)}{\|\chi(\xi_d)\|} = \begin{bmatrix} \chi_{x_d} \\ \chi_{y_d} \end{bmatrix}
\]  

(11)

\[
\Psi_{\text{desired}} = \tan(\chi_{x_d}, \chi_{y_d})
\]  

(12)

In addition, the derivative of \(\chi_{\text{desired}}\) along its trajectories, \(\chi_{\text{desired}}\), is given by:

\[
\frac{d}{dt} \chi(\xi_d) = -\omega_d \frac{d}{dt} \chi_{\text{desired}} = \begin{bmatrix} \chi_{x_d} \\ \chi_{y_d} \end{bmatrix}
\]  

(13)

Assuming that \(\|\chi_{\text{desired}}\| = 1\) and \(\chi_{\text{desired}} \chi_{\text{desired}} = 0\), \(\omega_d\) is a scalar function that can be obtained explicitly from the set of equations:

\[
\begin{cases}
\dot{\chi}(\xi_d) = (I - k_n e_{\text{track}}(\xi_d) I_2) H(\xi_d) \dot{\xi}_d - k_n e_{\text{track}}(\xi_d) \nabla \phi(\xi_d) \\
e_{\text{track}}(\xi_d) = M(\phi(\xi_d)) \|\nabla \phi(\xi_d)\| \dot{\xi}_d \\
\chi_{\text{desired}} = \frac{d}{dt} \left( \frac{\chi(\xi_d)}{\|\chi(\xi_d)\|} \right) = -\frac{1}{\|\chi(\xi_d)\|} \Gamma \chi_{\text{desired}}^T \Gamma \chi(\xi_d)
\end{cases}
\]  

(14)
where \( H(\xi_d) \) stands for the Hessian:

\[
H(\xi_d) = \begin{bmatrix}
\frac{\partial^2}{\partial x^2} \varphi(\xi_d) & \frac{\partial^2}{\partial y \partial x} \varphi(\xi_d) \\
\frac{\partial^2}{\partial y \partial x} \varphi(\xi_d) & \frac{\partial^2}{\partial y^2} \varphi(\xi_d)
\end{bmatrix}
\] (15)

For instance, let us recall the implicit function that describes a circle with the radius \( r > 0 \) centered on \( c = [c_x, c_y]^T \in \mathbb{R}^2 \), thus defining \( \varphi(\xi_d) \) as:

\[
\varphi(\xi_d) = (x_d - c_x)^2 + (y_d - c_y)^2 - r^2
\] (16)

Taking \( e_{\text{track}}(\xi) = \varphi(\xi), \gamma = -1 \) and \( k_a = 1 \), Eq. (9) applied over Eq. (16), we obtain the vector field:

\[
\chi(\xi_d) = 2 \begin{bmatrix}
-\varphi(\xi_d) & -1 & 1 & -\varphi(\xi_d) \\
1 & -\varphi(\xi_d) & -\varphi(\xi_d) & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\] (17)

from which the directional unit vector field and its derivative, \( \chi_d \) and \( \dot{\chi}_d \), can be computed. Fig. 7 shows the directional unit vector field of a circle with \( r = 1 \) and \( c = [0, 0]^T \), which was used to perform the experimental tests, the vector field of which is described by the expression:

\[
\chi_0(\xi_d) = \begin{bmatrix}
-x^3 - xy^2 + x - y \\
-y^3 - x^2y + x + y
\end{bmatrix}
\] (18)

\section*{B. Roll and Yaw controllers}

Once the desired attitude has been set, based on the vector field (Eq. 17), the heading error and its derivative can be written as follows:

\[
e_{\psi} = \Psi_{\text{desired}} - \Psi_{\text{meas}} = \arctan(2(\chi_{xy} - \chi_d) - \Psi_{\text{meas}})
\] (19)

\[
\dot{e}_{\psi} = \chi_{xy} - \chi_d - \Psi_{\text{meas}}
\] (20)

such that Eq. (19) holds for \( \Psi_{\text{desired}} - \Psi_{\text{meas}} < 180^\circ \). Eq. (20) can be used to define a PD controller having the form:

\[
u_{\psi} = C_{\psi}(s) \cdot e_{\psi} = \text{sawtooth}(e_{\psi}) + \dot{e}_{\psi}
\] (21)

which gives a reference value that is passed through the filter \( F_{\psi}(s) \) in order to compute the roll reference \( \Phi_{\text{desired}} \). The function \( \text{sawtooth}(\bullet) \) serves as a wrapping function to fit the yaw angle between 0 and 360 [deg]. More precision on the calculation of \( \dot{e}_{\psi} \) can be found in [34]. To carry out the experiments, the control scheme depicted in Fig. 6 was adopted. In this respect, a constant setpoint \( Z_{\text{desired}} \) was given to the vehicle in order to keep its altitude constant by updating the flapping frequency command via the action of the PID controller \( C_{\psi}(s) \) and the information provided by the MoCap Vicon i.e. the position and the orientation. The directional closed loop control takes the desired yaw angle \( \Psi_{\text{desired}} \) computed with the vector field. Once this reference value has been computed, it is compared with the actual yaw angle \( \Psi_{\text{meas}} \) of the Metafly so that the PD controller \( C_{\psi} \) and the filter \( F_{\psi} \) will produce the appropriate desired roll angle, which is limited to between \(-35^\circ\) and \(35^\circ\) (to prevent the metafly from diving and crashing as observed in experiments), and this value is then compared with the actual roll angle. The resulting error is thus processed by the lead controller, \( C_{\Phi} \) which sends the \( t_{\text{deflection}} \) command activating the rudder. The altitude of the Metafly is also controlled concomitantly.

\section*{V. EXPERIMENTAL RESULTS}

Using the vector fields and the 3-state directional command, we established that the Metafly based on the model described as follows:

\[
F_{\psi}(s) = 1.28 \begin{bmatrix}
s + 0.1 \\
\frac{s}{s + 0.3}
\end{bmatrix}
\] (22)

\[
C_{\Phi}(s) = \frac{t_{\text{deflection}}}{e_{\Phi}} = 3 \begin{bmatrix}
s + 0.015 \\
\frac{s}{s + 0.055}
\end{bmatrix}
\] (23)

These were used for the automated trajectory tracking of the Metafly presented in the following section.
in section III and the controllers we developed, were able to perform circular trajectory tracking successfully. As shown in Fig. 8, the Metafly completed 21 circular trajectories with a radius of 1.75m (around 230m) successfully, while keeping a constant altitude. The radius error distribution was centered at 0.0548m and the standard deviation was 0.128m. Table II summarizes the circular trajectory tracking performances obtained with a mean error of almost 0 in the radius.

In Fig. 9, the Metafly is able to change the radius of its circular trajectory during flight: it performed 4 circular laps with a radius of 1.5m before changing its trajectory and performing 4 larger circular laps with a radius of 2.25m. With the vector field method, the Metafly is able to follow more complex trajectories, such as an 8-shaped circuit set up in several steps, each of which was a section of the 8-shaped trajectory obtained with vector fields converging with a circle (Fig. 11).

The use of the vector field method for flapping-wing MAVs is one of the main contributions of this study. As discussed in the introduction, the vector field method can be used to track a trajectory without imposing strict initial position and/or speed conditions. In Fig. 10, tests were performed with various initial conditions. The circular trajectory tracking task was performed successfully with initial speeds ranging from 2m/s to 3m/s and various positions along a 1.25m-wide starting line. So far, outside the present range of initial conditions, the initial error of the heading angle is such that the Metafly cannot track the circle.

TABLE II
PERFORMANCES DURING A CIRCULAR TRACKING TRAJECTORY IN FIG. 8

<table>
<thead>
<tr>
<th></th>
<th>std [m]</th>
<th>mean [m]</th>
<th>median [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius error</td>
<td>0.128</td>
<td>0.0548</td>
<td>0.0459</td>
</tr>
<tr>
<td>Altitude error</td>
<td>0.1040</td>
<td>0.2</td>
<td>0.2078</td>
</tr>
</tbody>
</table>

Fig. 11. Metafly tracking an 8-shape trajectory: A) Steps segmentation according to the position of the vehicle to create an 8-shape circuit with vector fields. B) Resulting 8-shape trajectory performed by the Metafly.

VI. CONCLUSION

In this study, it was established for the first time (to the best of our knowledge) that a vector field method could be used to make a micro FWAV perform trajectory tracking successfully over 230m. The vector field method, which was proved to be time-independent, enabled the Metafly to track trajectories for various initial position and initial velocity conditions. The results obtained here show that system identification based on enriched manual flight test data can provide an effective basis for designing FWMAV controllers. The 3-state directional command presented here enabled the Metafly to perform circular trajectory tracking and to track an 8-shaped circuit while keeping a constant altitude. In future studies, we would like to (i) to increase the automation of the Metafly using micro-GPSs, gyros, magnetometers or optic flow sensors while taking into account the SSWaP (Speed Size Weight and Power) constraints of micro aerial vehicles [39,40], and (ii) to improve their vector field-based trajectory tracking performances in the presence of wind and obstacles outdoors.

ACKNOWLEDGMENT

We thank Dr. J. Blanc for improving the English manuscript. We thank J.M. Ingargiola for the electronics, J. Diperi for the micro-mechanics, C. Coquet for the Vicon system, A. Negre, J. Dumond for the HF connection and S. Durand for proofreading. Financial support was provided by a RAPID Biofly grant to E.V.R. & F.R. from the AID/DGA. This work has been partially supported by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015).