30th Congress of the International Council
of the Aeronautical Sciences
DCC, Daejeon, Korea ; September 25-30, 2016

NON-COOPERATIVE GROUND VEHICLE TRACKING AND INTERCEPTION BY MULTI-RPA COLLABORATION

Yoko Watanabe, Augustin Manecy, Alexandre Amiez, Charles Lesire, Christophe Grand
Department of Systems Control and Flight Dynamics
ONERA - The French Aerospace Laboratory
2 avenue Edouard Belin, 31055 Toulouse, France

Keywords: *multi-RPA operation, vision-based target tracking, interception guidance*

Abstract

This paper proposes a system for non-cooperative vehicle tracking and interception by collaboration of two Remotely Piloted Aircraft (RPA). In this system, one RPA visually detects and localizes a targeted vehicle from high altitude, and the other makes an interception with it at low altitude by using the target localization result sent by the first RPA. Compared to a mono-RPA operation for the same purpose, the proposed system has advantages of i) easily keeping a target in a field-of-view, ii) being able to make an agile maneuver to intercept with the target from any angle, iii) ensuring the communication relay to a ground control station, and iv) requiring less RPA payload capacity. The paper presents the proposed system design including vision-based target tracking and interception guidance, and flight experiment results using ONERA RPA platforms.

1 Introduction

Along with the popularization of low-cost camera-equipped drones, demands and interests of using RPAS (Remotely Piloted Aerial Systems) for civil or military services have been significantly increased in the last decade. The EU co-funded AEROCEPTOR project[1] *proposed a new concept of using RPAS to remotely and safely track and stop a non-cooperative vehicle in both land and sea scenarios. These scenarios assume police-type of operation such as border control, highway traffic control, drugs and weapons smuggling and crime scene escape. Tasks of the AEROCEPTOR project included; i) studies in different payloads for remotely stopping vehicles, ii) studies in RPA flight control system for vehicle interception, and iii) system integration and in-flight validation. The project investigated various types of payload systems for remotely stopping a vehicle; from a simple sign board, a caltrop-dropping system[1] to an electromagnetic antenna to stop an engine[2]. In order to achieve efficient and safe vehicle stopping intervention, requirements for the RPA's target interception differ in function of characteristics of such payload systems. For example, some payloads need to be deployed from the front of the target while some from the side.*

In this context, this paper presents a Master-Slave RPA collaboration system for tracking and intercepting a targeted vehicle at a desired impact angle. In this system, Master-RPA pursuits the target from high altitude and serves for communication relay as well as target detection and localization. Slave-RPA performs target interception and stopping intervention by using the target localization result sent by the Master. In function of distance, additional RPAs can be inserted between the Ground Control Station (GCS) and the Master-RPA for communication relay if necessary. The proposed system has the following advantages over a classical mono-RPA target track-

[1] EU-FP7(2012-2016) - http://www.aeroceptor.eu/

ing system (such as [3]):

- **Wide field-of-view for target detection:** *As Master-RPA stays at high altitude, its onboard camera has a wide view over ground/sea surface which is beneficial for vision-based target detection and localization.*
- **No maneuverability constraint for target interception:** *Since Slave-RPA does not need to "see" the target, it can take agile maneuver to track and intercept the target from any desired angle.*
- **Communication relay to GCS:** *Master-RPA stays at high altitude during the operation, and so can make a communication relay between Slave-RPA and GCS when the direct communication is difficult due to signal occlusion.*
- **Distribution of the payload:** *The payload for target stop intervention and that for target detection are distributed on the two RPAs. Consequently, payload capacity required for each RPA can be reduced.*

In particular, the last point justifies our choice of this Master-Slave collaboration configuration instead of multi-RPA parallel tracking, which is found often in literatures as it has advantages of making the tracking system tolerant to target occlusion, poor observability and the presence of obstacles[4]-[6].

2 System Overview

This paper proposes a system for non-cooperative vehicle tracking and interception by collaboration of two RPAs: Master- and Slave-RPA. A targeted vehicle, called POI (Point Of Interest), could be a car or a boat in land/sea scenarios. Figure 1 shows an overview of the proposed system. In this system, Master-RPA's main role is to detect and localize a non-cooperative POI by using onboard vision and send its localization state (position and velocity) to Slave-RPA. To this end, Master-RPA is required to have a mono-RPA POI tracking system which consists of algorithms of image processing for POI detection, POI state estimation and tracking guidance (Figure 1-right). Slave-RPA continuously receives POI's state from Master-RPA and makes an interception with POI in order to deploy a payload system for stopping it. It is considered that the payload system requires for Slave-RPA to intercept with POI at a specific position X_{imp} and angle ψ_{imp} for its efficient and safe deployment. This angle ψ_{imp} is called impact angle. For example, $\psi_{imp} = 180^\circ$ means frontal interception, and $\psi_{imp} = 90^\circ$ means interception from the left side of the POI. Slave-RPA needs a guidance law to achieve this desired interception by using the POI state information sent by the Master (Figure 1-left). The vision-based POI tracking system for the Master-RPA and the angle constrained interception guidance for the Slave-RPA will be described in the following two sub-sections.

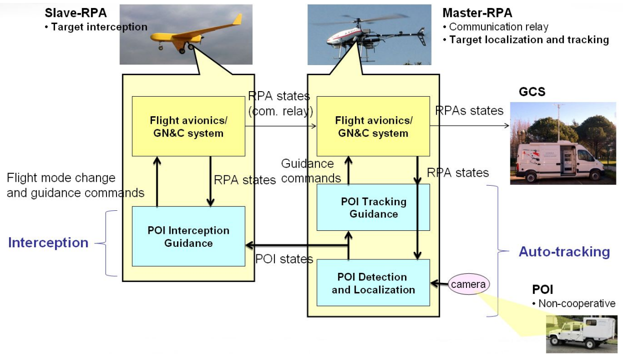


Fig. 1 Onboard systems on the Master- and the Slave-RPAs

2.1 Master-RPA: POI Localization and Tracking

2.1.1 Image processing for POI detection

To detect and track POI, a new vision-based tracking algorithm was developed. As the tracking task is performed with an RPA in outdoor conditions, the algorithm needs to be robust to strong illumination changes, fit with the real time requirements of the embedded computer, and be able to yield a confidence measurement. The developed POI image tracking algorithm is based on the color signature of the object, and the tracking task can be initialized by two different way: 1) a human operator can select the object to track in the video, thanks to a wireless video link, 2) a file can be used to load the object's description of one of the object saved in an object collection. We define the following notations (see also Figure 2):

- objectBox: minimal oriented rectangle containing the detected object,
- boundingBox: minimal rectangle containing the objectBox,
- searchArea: 1.5 times the previous bounding Box,

The visual tracker is able to track an object in a RGB or a Black&White vieo flow, as it is shown in 2. It consists in a variation of the well-knwon CAMSHIFT algorithm presented in [7]. In case of a RGB video flow, the images are first converted into HSV color space to be more robust o illumination variation. Then the target tracking algorithm uses a histogram of the target (extract form the operator selection or from the target reference file) to find the object in the image. First it computes the histogram of the full image, if there is similarities between the full image's histogram (i.e., the background) and the object's histogram, these similarities are ignored. A new object's histogram is built, containing only the object specific color, as shown in Figure 2b-c). For example, the background hue component (Figure 2c) are enforced to zero in the original object's histogram (Figure 2b). This simple trick make the tracking algorithm much more robust. Then a backprojection is computed on the searchArea and the CAMSHIFT algorithm is applied to find the object. Finally, to build the confidence measurement a new histogram is computed on the detected objectBox thanks to a cross-correlation between this new histogram and the original object histogram:

$$COV(t) = \frac{minCOV}{confidence(t)} \qquad (1)$$

Where $minCOV$ is the minimum covariance (precision of the visual tracking), set here to 10 pixels. The tracking algorithm is implemented in C++ using OpenCV, and runs in real-time onboard the payload computer embedded on a small helicopter. The tracking algorithm finally yields the objectBox (heading, width, length and pixel position) of the detected object and the confidence. The confidence parameter is useful to build artificially the covariance relative to the objectBox, needed by the EKF (Extended Kalman Filter) estimating the target states. During the flight test, the tracking task can be start and stop thanks from a graphical user interface (GUI) developed in this work. This GUI, shown in Figure 3, allows to visualize the current RPA position, the estimated POI position, and the visual tracking feedback.

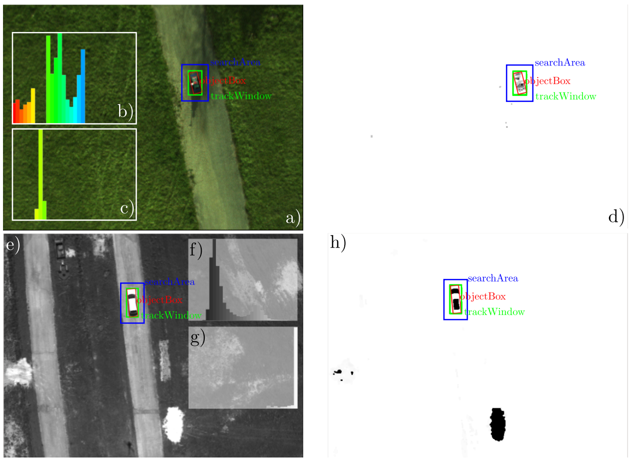
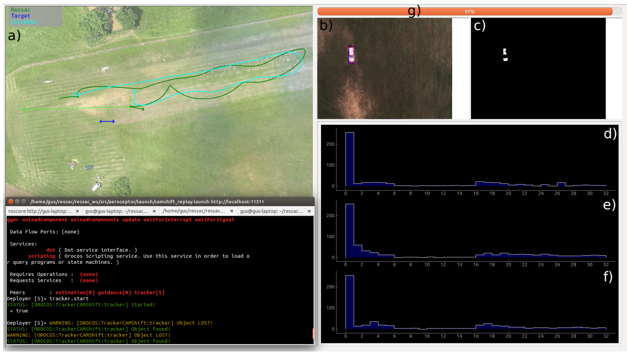


Fig. 2 a,e) Original RGB/grayscaled image with the tracking results, b,f) POI's histogram, c,g) The background histogram, d,h) The back projection and results of the CAMSHIFT algorithm.

2.1.2 POI localization

The POI position and velocity can be estimated by using the vision-based information provided by the image processor presented in Section 2.1.1. Let $X_{POI} = \begin{bmatrix} X_{POI} & Y_{POI} & Z_{POI} \end{bmatrix}^T$ denote the POI position in a locally fixed reference frame. Assuming a flat surface, the POI velocity can be expressed as $V_{POI} = V_{POI} \begin{bmatrix} \cos\psi_{POI} & \sin\psi_{POI} & 0 \end{bmatrix}^T$. Given the camera position X_{CAM} and orientation R_{CI} with respect to the reference frame, the POI center position appears on an image at

$$x_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} \frac{X^C}{Z^C} = K_{CAM} \frac{X^C}{Z^C} \qquad (2)$$



```
/home/gus/ressac/ressac_ws/src/aeroceptor/launch/camshift_replay.launch http://localhost:11311
roscore http://gus-laptop:...  gus@gus-laptop: ~/ressac...  /home/gus/ressac/ressac...  gus@gus-laptop: ~/ressac...
gger unloadComponent unloadComponents update waitForInterrupt waitForSignal

 Data Flow Ports: (none)

 Services:
            dot ( Dot service interface. )
      scripting ( Orocos Scripting service. Use this service in order to load o
r query programs or state machines. )

 Requires Operations :  (none)
 Requests Services   :  (none)

 Peers        : estimation[R] guidance[R] tracker[S]
Deployer [S]> tracker.start
STATUS: [OROCOS:TrackerCAMShift:tracker] Started!
 = true

Deployer [S]> WARNING: [OROCOS:TrackerCAMShift:tracker] Object LOST!
STATUS: [OROCOS:TrackerCAMShift:tracker] Object found!
WARNING: [OROCOS:TrackerCAMShift:tracker] Object LOST!
STATUS: [OROCOS:TrackerCAMShift:tracker] Object found!
```

Fig. 3 a) Local map with RPA and POI trajectories, b) Raw images with detected objectBox and boundingBox, c) Back projection of the image using the POI histogram, d) POI's reference histogram, e) POI histogram, f) Full image histogram, g) Confidence level.

where (f_x, f_y) is the camera's focal length in pixels, (c_x, c_y) is the image center in pixels, and

$$X^C = \begin{bmatrix} X^C & Y^C & Z^C \end{bmatrix}^T = R_{CI}\left(X_{POI} - X_{CAM}\right).$$

Let L and W be (unknown) length and width of POI. Then, the target size (b_l, b_w) in pixels on image will be

$$\begin{aligned} b_l &= \|x_{front} - x_{back}\| \\ &\simeq L\|M(X^C)R_{CI}\begin{bmatrix}\cos\psi_{POI}\\ \sin\psi_{POI}\\ 0\end{bmatrix}\| \\ &= L\|u(X_{POI},\psi_{POI})\| \qquad (3) \\ b_w &= \|x_{right} - x_{left}\| \\ &\simeq W\|M(X^C)R_{CI}\begin{bmatrix}-\sin\psi_{POI}\\ \cos\psi_{POI}\\ 0\end{bmatrix}\| \\ &= W\|v(X_{POI},\psi_{POI})\| \qquad (4) \end{aligned}$$

where

$$M(X^C) = \frac{K_{CAM} + \begin{bmatrix}0 & 0 & -x_c\end{bmatrix}}{Z^C}.$$

Let ψ_I be the target heading on the image. From the relation of $x_{front} - x_{back} = b_l \begin{bmatrix}\cos\psi_I & \sin\psi_I\end{bmatrix}^T$,

$$\psi_I = atan2\left(u_y(X_{POI},\psi_{POI}), u_x(X_{POI},\psi_{POI})\right) \qquad (5)$$

Then, the output model of the image processor at an instant $t = t_k$ is given by

$$z_k = \begin{bmatrix}x_c(t_k)\\ b_l(t_k)\\ b_w(t_k)\\ \psi_I(t_k)\end{bmatrix} + \xi_k = h(X_{POI}(t_k), \psi_{POI}(t_k), L, W) + \xi_k \qquad (6)$$

where ξ_k is the measurement error modeled as a zero-mean Gaussian noise with a covariance matrix R. In this work, this R is defined in function of the confidence level given by the image processor. The function $h(\cdot)$ is given by (2-5). Figure 4 shows the four vision-based measurements to be used for the POI localization.

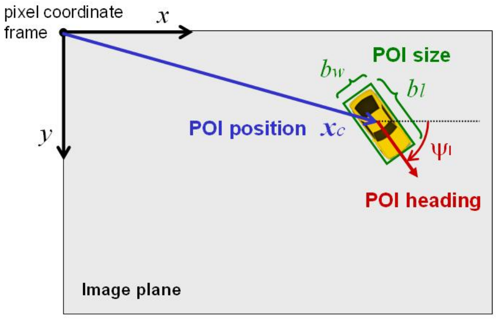


Fig. 4 Vision-based mesurements for POI localization

Now, our POI localization problem is to estimate the unknown POI state $x_{POI} = (X_{POI}, V_{POI}, \psi_{POI}, L, W)$ from the vision-based measurement (6). Assuming a non-maneuvering POI on a flat ground/sea surface, a dynamics of x_{POI} is given by

$$\dot{x}_{POI} = \begin{bmatrix}\dot{X}_{POI}\\ \dot{Y}_{POI}\\ \dot{Z}_{POI}\\ \dot{V}_{POI}\\ \dot{\psi}_{POI}\\ \dot{L}\\ \dot{W}\end{bmatrix} = \begin{bmatrix}V_{POI}\cos\psi_{POI}\\ V_{POI}\sin\psi_{POI}\\ 0\\ 0\\ 0\\ 0\\ 0\end{bmatrix} + \nu \qquad (7)$$

where ν is a process noise modeled as a zero-mean Gaussian noise with a covariance matrix Q. An Extended Kalman Filter (EKF) is applied to the process model (7) and the measurement model (6) to solve the POI localization problem.

2.1.3 POI tracking guidance

In order to maintain the POI in a field-of-view of its onboard camera, Master-RPA needs to track the moving POI. In this paper, a VTOL-type of vehicle is considered for the Master-RPA. The control objectives are;

- To horizontally track the POI by keeping a desired relative position (X_{track}, Y_{track}),
- To maintain a given constant altitude Z_{Md}, and
- To align the RPA heading to that of the POI ψ_{POI}.

The desired position (X_{Md}, Y_{Md}) for Master-RPA can be given by

$$\begin{bmatrix} X_{Md} \\ Y_{Md} \end{bmatrix} = \begin{bmatrix} \hat{X}_{POI} \\ \hat{Y}_{POI} \end{bmatrix} + \begin{bmatrix} \cos\hat{\psi}_{POI} & -\sin\hat{\psi}_{POI} \\ \sin\hat{\psi}_{POI} & \cos\hat{\psi}_{POI} \end{bmatrix} \begin{bmatrix} X_{track} \\ Y_{track} \end{bmatrix}$$

where $\hat{\cdot}$ denotes the estimated POI state provided by the estimator described in Section 2.1.2. The paper uses the previously developed ONERA flight controller[8] which takes command inputs of desired horizontal velocity, altitude and heading. A simple linear controller is applied to calculate those commands to track the POI.

$$\begin{bmatrix} V_{X_{Md}} \\ V_{Y_{Md}} \\ Z_{Md} \\ \psi_{Md} \end{bmatrix} = \begin{bmatrix} -K_p \begin{bmatrix} X_M - X_{Md} \\ Y_M - Y_{Md} \end{bmatrix} - K_d \begin{bmatrix} V_{X_M} \\ V_{Y_M} \end{bmatrix} \\ Z_{Md} \\ \psi_M + K_\psi(\hat{\psi}_{POI} - \psi_M) \end{bmatrix} \tag{8}$$

where X_M, V_M and ψ_M are the known position, velocity and heading of the Master-RPA.

2.2 Slave-RPA: POI Interception

Slave-RPA continuously receives the estimated POI state $\hat{x}_{POI}$ from the Master-RPA's estimator, and makes use of it to achieve an interception with the POI at a specific position X_{imp} and impact angle ψ_{imp}. Those parameters X_{imp} and ψ_{imp} are defined by requirements of the payload system for its safe and efficient stop intervention of the POI. At each time instant, the desired interception position and angle can be estimated as

$$\hat{X}_t(t) = \hat{X}_{POI}(t) + R_3(-\hat{\psi}_{POI}(t))X_{imp} \tag{9}$$

$$\hat{\psi}_t(t) = \hat{\psi}_{POI}(t) + \psi_{imp} \tag{10}$$

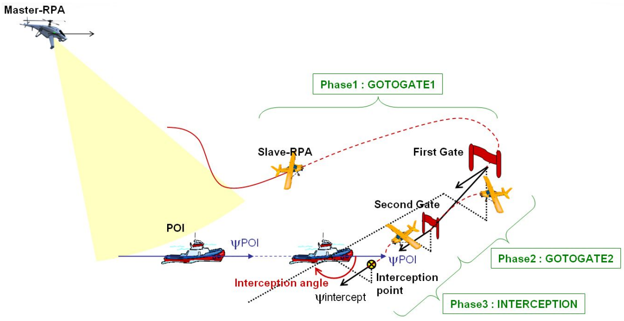


Fig. 5 Three-step target interception guidance procedure

where $R_3(\cdot)$ is a Z-axis rotation matrix. In this paper, a fixed-wing RPA with a high speed flying capability is assumed for Slave-RPA to catch up with POI. In order to ensure the interception precision, this paper proposes a three-step interception guidance procedure as illustrated in Figure 5. Suppose that Slave-RPA is loitering somewhere before receiving interception request from an operator on ground. Triggered by this operator's request, three different flight phases (GOTOGATE1, GOTOGATE2, INTERCEPTION) will be consecutively executed to achieve the final interception. This section describes a guidance law for each of those three flight phases.

2.2.1 Phase1-GOTOGATE1

The first step of the POI interception procedure is to bring Slave-RPA on the interception course. When received an interception request, a predicted interception point is first identified by assuming a non-maneuvering POI. Let t_0 and t_f be the current and predicted interception time. Then the predicted interception point is given by

$$\hat{X}_t(t_f) = \hat{X}_t(t_0) + \hat{V}_{POI}(t_f - t_0) \begin{bmatrix} \cos\hat{\psi}_{POI} \\ \sin\hat{\psi}_{POI} \\ 0 \end{bmatrix} \tag{11}$$

Then GATE-1 is defined at a point d_{gate1} short of and $\Delta Z_{gate1} (> 0)$ above this interception point. For flight safety, ΔZ_{gate1} is set still higher than *the intervention altitude.*

$$X_{gate1} = \hat{X}_t(t_f) - \begin{bmatrix} d_{gate1}\cos\hat{\psi}_t \\ d_{gate1}\sin\hat{\psi}_t \\ \Delta Z_{gate1} \end{bmatrix} \tag{12}$$

The SNAKE guidance algorithm[8], previously developed at ONERA for waypoint tracking guidance, is applied to attain X_{gate1} with the desired heading $\hat{\psi}_t$ and speed V_{gate1}. This SNAKE algorithm treats horizontal and vertical guidance separately. For vertical guidance, a linear trajectory with a constant flight path angle is set as a reference. The horizontal guidance uses a minimum-time trajectory which results in an arc-line-arc trajectory as shown in Figure 6. The arc radii (R_1 and R_2) are determined by the current and desired waypoint speed respectively. The arc angles (α_1 and α_2) and the line length (d_{12}) are calculated from a geometry of the current RPA position and the waypoint. Given a gudance horizon d_{guid}, the heading rate command is calculated as follows.

$$\Delta\psi_{com} = \begin{cases} \omega_1, & \text{if } R_1|\alpha_1| \geq d_{guid} \\ \frac{R_1|\alpha_1|}{d_{guid}}\omega_1, & \text{else if } R_1|\alpha_1| + d_{12} \geq d_{guid} \\ \frac{R_1|\alpha_1|}{d_{guid}}\omega_1 + \frac{d_{guid}-R_1|\alpha_1|-d_{12}}{d_{guid}}\omega_2, & \text{otherwise} \end{cases}$$

where $\omega_1 = sign(\alpha_1)V_S(t_0)/R_1$ and $\omega_2 = sign(\alpha_2)V_{gate1}/R_2$ are the turning rate on the arcs.

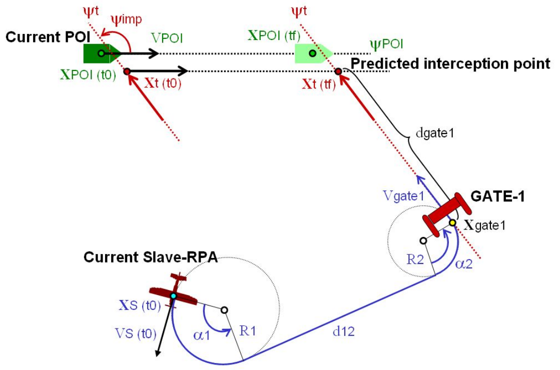


Fig. 6 Predicted interception point and GATE-1 definition

In our guidance design, we set the GATE-1 speed equal to the current horizontal speed of Slave-RPA: $V_{gate1} = V_S(t_0)$. Then, the time when Slave-RPA attains GATE-1 can be estimated as

$$t_{gate1}(t_f) = t_0 + \frac{R_1|\alpha_1| + d_{12} + R_2|\alpha_2|}{V_S(t_0)}$$

It should be recalled that the arc radii, angles and a length of the line segment are functions of t_f, and hence this t_{gate1} is also a function of t_f. After achieving GATE-1, Slave-RPA flies straight towards the interception point $\hat{X}_t(t_f)$ while deccelerating (or accelerating) up to the desired interception speed V_{imp}. Let $a_{max}(>0)$ be a maximum acceleration limit defined in the flight controller. Supposing long enough d_{gate1}, the time to intercept POI from GATE-1 can be given by

$$\Delta t_{imp} = \frac{d_{gate1}}{V_{imp}} + \frac{s_0(V_{imp} - V_S(t_0))^2}{2a_{max}V_{imp}} \tag{13}$$

where $s_0 = sign(V_{imp} - V_S(t_0))$. This Δt_{imp} does not depend on t_f. Then, the time of interception becomes

$$t_{imp}(t_f) = t_{gate1}(t_f) + \Delta t_{imp}.$$

The predicted interception time t_f and hence $\hat{X}_t(t_f)$ are identified by solving an equation $t_{imp}(t_f) = t_f$. Since the minimum-time trajectory parameters (R_1, R_2, α_1, α_2 and d_{12}) are nonlinear to t_f, an iterative search method based on linear interpolation is applied to find the solution. Once the solution t_f is obtained, the GATE-1 position X_{gate1} given in (12) is set as a waypoint until the SNAKE guidance achievs it within a given miss-distance threshold.

2.2.2 Phase2-GOTOGATE2

The second step after achieving the GATE-1 is to refine Slave-RPA's trajectory on the interception course, as well as to adjust Slave-RPA's speed and altitude to their desired value of the interception. During this GOTOGATE2 phase, the predicted interception point is continuously updated by the current Slave-RPA and POI states. Let t_1 and t_f be the current and predicted interception time, and the interception point $\hat{X}_t(t_f)$ can be predicted as in (11). Then GATE-2 is defined *at a point $d_{gate2}(< d_{gate1})$ short of this interception point as shown in Figure 7.*

$$X_{gate2} = \hat{X}_t(t_f) - \begin{bmatrix} d_{gate2}\cos\hat{\psi}_t \\ d_{gate2}\sin\hat{\psi}_t \\ 0 \end{bmatrix} \qquad (14)$$

The distance d between $\hat{X}_t(t_f)$ and the current Slave-RPA position $X_S(t_1)$ in direction of $\hat{\psi}_t$ is

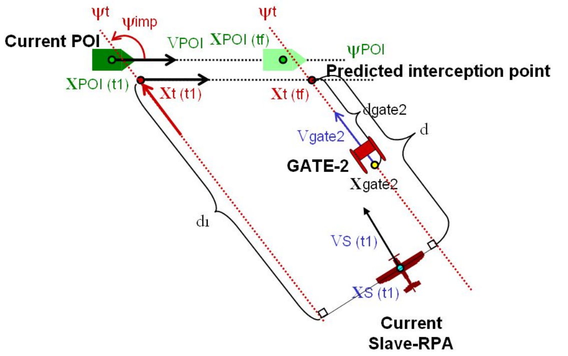


Fig. 7 Predicted interception point and GATE-2 definition

$$d = d_1 + \hat{V}_{POI}(t_f - t_1)\cos\psi_{imp}$$

where $d_1 = \begin{bmatrix}\cos\hat{\psi}_t & \sin\hat{\psi}_t & 0\end{bmatrix}\left(\hat{X}_t(t_1) - X_S(t_1)\right)$. In the same manner as (13), the time to interception from the current state can be calculated as

$$\Delta t_{imp} = \frac{d}{V_{imp}} + \frac{s_1(V_{imp} - V_S(t_1))^2}{2a_{max}V_{imp}}$$

where $s_1 = sign(V_{imp} - V_S(t_1))$. Then, the predicted interception time t_f can be solved from the relation $\Delta t_{imp} = t_f - t_1$, resulting in

$$t_f = t_1 + \frac{d_1 + \frac{s_1(V_{imp} - V_S(t_1))^2}{2a_{max}}}{V_{imp} - \hat{V}_{POI}\cos\psi_{imp}}$$

with a condition of $V_{imp} > \hat{V}_{POI}\cos\psi_{imp}$. Once t_f is identified, the GATE-2 position is defined according to (14) and set as a waypoint with the desired heading $\hat{\psi}_t$ and speed $V_{gate2} = V_{imp}$.

Unlike the GOTOGATE1 phase, the control objective of this GOTOGATE2 phase is not just to reach the GATE-2 with the desired heading and speed but to track a straight line which goes through the GATE-2 in a direction of the desired heading (a red dotted line in Figure 7). As in the SNAKE guidance, the vertical guidance uses a reference trajectory with constant flight path angle. The velocity vector field approach[9] is applied for the horizontal guidance. This approach calculates the velocity direction (hence the heading angle in a fixed-wing RPA case) command in function of the distance of the RPA to the reference trajectory line.Let

$$\varepsilon = \begin{bmatrix} -\sin\hat{\psi}_t \\ \cos\hat{\psi}_t \\ 0 \end{bmatrix}^T (X_{gate2} - X_S(t_1))\,.$$

Then, the velocity direction command is given by

$$\psi_{Sd} = \begin{cases} \hat{\psi}_t + sign(\varepsilon)\frac{\pi}{2}, & |\varepsilon| > E \\ \hat{\psi}_t + sign(\varepsilon)(\frac{\pi}{2} - \arccos\frac{|\varepsilon|}{E}), & |\varepsilon| \le E \end{cases}$$

where $E > 0$ is a given bound limit. GATE-2 is validated when $t_f < 0$ and the initial geometry condition required by the interception guidance law ((18) in the next subsection) is satisfied.

2.2.3 Phase3-INTERCEPTION:

When achieving GATE-2, Slave-RPA should be almost on the final interception course with the desired interception speed and altitude. The last step of the interception proceudre is to make fine adjustment of the interception course and achieve the precision interception with POI. For this purpose, the Biased Proportional Navigation Guidance (BPNG) law[10] is applied. This BPNG is an extention of the conventional Proportional Navigation Guidance (PNG) [11] widely used for homing missile guidance. It adds a time-varying bias to the PNG in order to control the angle of impact.

Let X_S, V_S and ψ_S be the current position, horizontal speed and horizontal velocity direction of the Slave-RPA. In BPNG law, it is assumed that the speed is already regulated to the desired one. Hence, only the speed direction is controlled by a lateral acceleration $a_S = V_S\dot{\psi}_S$. Define the relative distance r and the Line-of-Sight (LoS) angle

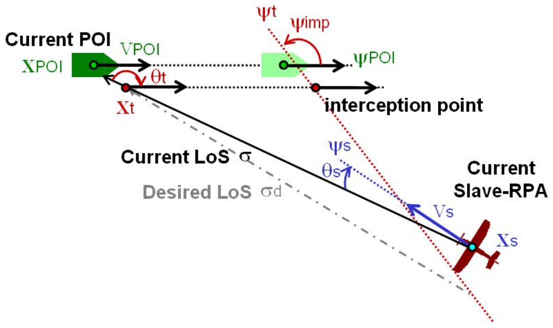


Fig. 8 BPNG geometry

σ of the targeted point X_t from the Slave-RPA as

$$\begin{bmatrix} X_t - X_S \\ Y_t - Y_S \end{bmatrix} = r \begin{bmatrix} \cos\sigma \\ \sin\sigma \end{bmatrix}.$$

Their dynamics are given by

$$\begin{aligned} \dot{r} &= V_{POI}\cos\theta_t - V_S\cos\theta_S \\ r\dot{\sigma} &= V_{POI}\sin\theta_t - V_S\sin\theta_S \end{aligned}$$

where $\theta_t = \psi_{POI} - \sigma$ and $\theta_S = \psi_S - \sigma$ as indicated in Figure 8. From the Slave-RPA's ideal speed V_{imp} and direction ψ_t, the desired LoS angle σ_d is given as a solution of

$$V_{POI}\sin(\psi_{POI} - \sigma_d) - V_{imp}\sin(\psi_t - \sigma_d) = 0.$$

The BPNG law regulates the LoS angle σ to this desired value σ_d, while the classic PNG aims only at constant σ. The heading rate command $\dot{\psi}_S$ is set as

$$\dot{\psi}_S = N\dot{\sigma} - \eta \frac{V_S(\sigma_d - \sigma)}{r \max\left(\cos\theta_S, \sqrt{1-(\rho+\beta)^2}\right)} \quad (15)$$

where $\rho = V_S/V_{POI}$, $\beta \in (0, 1-\rho)$, $\eta > 0$ and

$$N > 1 + \frac{\rho}{\sqrt{1-(\rho+\beta)^2}}. \quad (16)$$

According to [10], this BPNG law (15) guarantees the interception when the following initial geometry condition satisfies.

$$\begin{cases} |\theta_S| < \frac{\pi}{2} \\ \sqrt{(\rho\sin\theta_t - \sin\theta_S)^2 + \eta(\sigma_d - \sigma)^2} < \beta \end{cases} \quad (17)$$

In our design, $N = 4$ and $\eta = 2$ are used since they coincide with the optimal gain which minimizes a quadratic cost of the acceleration when Slave-RPA and POI are on the near interception course. For this fixed gains, one can find $\beta \in (0, 1-\rho)$ which satisfies the condition (16, 17) when the following initial geometry condition is met.

$$\begin{aligned} &\sqrt{(\rho\sin\theta_t - \sin\theta_S)^2 + \eta(\sigma_d - \sigma)^2} \\ &< \sqrt{1 - \frac{\rho^2}{(N-1)^2}} - \rho = \tilde{\rho} \end{aligned} \quad (18)$$

Therefore, one of the criteria to validate GATE-2 is this condition. During this INTERCEPTION phase, the BPNG heading rate command (15) is calculated based on the current POI estimated state. $\beta = (1-\mu)\tilde{\rho}$ is used with small $\mu > 0$. Then this is integrated to obtain a heading commend $\psi_{S_d} = \psi_S + \dot{\psi}_S \Delta t$ for the guidance horizon. It should be noted that unlike the first two phases this phase applies a direct interception guidance without predicting explicitly the interception time and point. If the interception is achieved with allowable miss-distance and miss-angle, the payload deployment will be triggered for remotely stopping POI.

3 Flight Experiment Results

The POI localization and tracking guidance of Master-RPA and the POI interception guidance of Slave-RPA, developed in this paper, have been tested on ONERA RPA platforms.

3.1 POI localization and tracking

By using the POI image tracker and the POI state estimator described in Section 2.1, a closed-loop vision-based tracking of a non-cooperative vehicle has been achieved. Figure 9 compares the vision-based measurements output by the POI image tracker with those calculated by using GPS-based POI and RPA states. The results show a good match between the vision-based and GPS-based measurements except a slight offset which might be due to a heading angle estimation error in the RPA state. Between $t = 80$ and $100(sec)$, *the tracker lost POI from the image. It is well appeared in a drop of the tracker's confidence level. Figure 10 shows the POI position and velocity estimation results by using the vision-based measurements, being compared to their GPS-based ground truth. Those states are estimated accurately except the period of image tracking failure. The 3D trajectory plot in Figure 10 also includes the trajectory of the RPA which automatically pursuit the POI by using the estimation results. Even with the POI loss period, the RPA succeeded to continue to track the vehicle.*

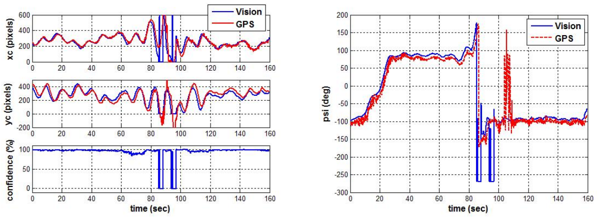


Fig. 9 Vision-based measurements: (left) pixel coordinates and confidence level, (right) heading angle

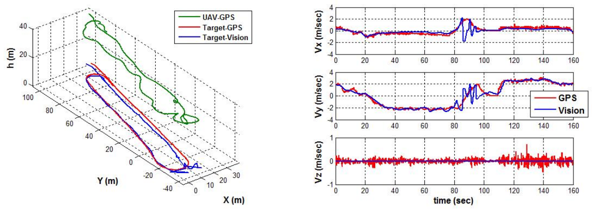


Fig. 10 POI localization results: (left) 3D trajectory, (right) velocity

3.2 POI interception

The three-step POI interception guidance law proposed in this paper has been implemented on the airplane RPA, and its performance was tested in flight by using the GPS-based POI state sent from the POI on ground. Due to the limited flight authorized zone, a stationary POI was used in the experiment. Figure 11 shows an example of the resulting RPA trajectory with a zero impact angle setting. The three different flight phases introduced in Section 2.2 are distinguished by colors on the trajectory, and the gate positions are also shown. A very good interception precision was obtained, with 2 meters of final miss distance. In order to show the interception capability at an arbitrary impact angles, Figure 12 illustrates the RPA interception trajectories obtained in the 6 DoF flight simulations. In these simulations, the impact angle ψ_{imp} was increased by 30 degrees from $\psi_{imp} = -180$ (deg). The precise interception with less than 1 meter miss distance was succeeded in all the cases except $\psi_{imp} = -30$ (deg) where the RPA intercepted at POI with 35 degrees of miss impact angle. This failure is due to a too-close configuration of the RPA and the POI. When $\psi_{imp} = -90$ (deg), the GATE-1 was not validated for the first time but the guidance procedure re-tried and succeeded the final interception. Those cases were shown in green in Figure 12.

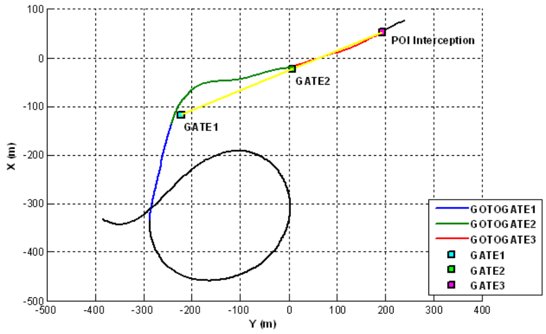


Fig. 11 Interception flight trajectory

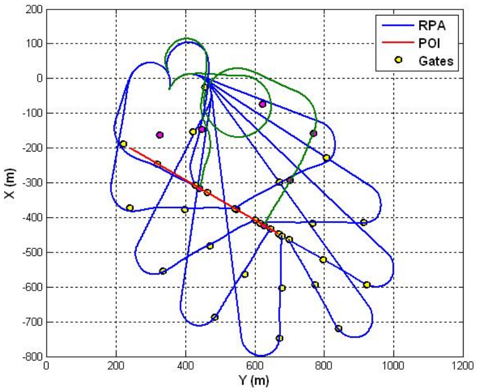


Fig. 12 Interception trajectories with different impact angles

4 Conclusion

This paper presented a concept of Master-Slave RPA collaborative tracking and interception of a non-cooperative vehicle which has many advantages over mono-RPA operation and multi-RPA parallel tracking. Flight guidance systems for POI tracking (Master-RPA) and interception (Slave-RPA) have been developed and validated onboard real RPA platforms. However, the entire scenario of collaborative tracking and interception by two RPA has not yet been realized. This is our immediate future work. For further development, we would like to modify the tracking and interception guidance law by considering a maneuvering POI. It can be done by using some additional information such as a road map for a ground vehicle or a probablistic POI maneuver model. We are also interested in taking into account an environmental condition (obstacles, no-fly zone) in the low-altitude Slave-RPA interception guidance.

Acknowledgement

The research leading to these results has received funding from the European Union (EU)'s Seventh Programme for research, technological development and demonstration under grant agreement No. 285144.

References

[1] A. Spronska, A. Woloszczuk, P. Salek, M. Macias and J. Glowka. "Concept Analysis and Development of an Innovative Remotely Controlled Portable Tyre Puncturing Device (R-TPD)", Challenges in Automation, Robotics and Mesurement Techniques, Springer, 2016.

[2] F. Christophe, D. Prost, L. Guibert, C. Martel and J-P. Parmantier. "Car-stopping concept based on an airborne HPM source", European Electromagnetics Symposium, 2016.

[3] Y. Watanabe, C. Lesire, A. Piquereau and P. Fabiani. "System Development and Flight Experiment of Vision-based Simultaneous Navigation and Tracking", AIAA Infotech@Aerospace, 2010.

[4] B. Bethke, M. Valenti and J. How. "Cooperative Vision Based Estimation and Tracking using Multiple UAV", Springer, 2007.

[5] S.A.P. Quintero, F. Papi, D.J. Klein, L. Chisci and J.P. Hespanha. "Optimal UAV coordination for target tracking using dynamic programming", IEEE Conference on Decision and Control, 2010.

[6] K. Hausman, G. Kahn, S. Patil, J. Muller, K. Goldberg, P. Abbeel and G.S. Sakhatme. "Cooperative Occlusion-Aware Multi-Robot Target Tracking using Optimization", IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015.

[7] G.R. Bradski. "Computer Vision Face Tracking For Use in a Perceptual User Interface", Intel Technology Journal Q2, 1998.

[8] P. Fabiani, A. Piquereau, V. Fuertes and P.M. Basset, "The ReSSAC Autonomous Rotorcraft: from autonomy demonstrations to out-of-sight flight", AHS Specialist's Meeting on Unmanned Rotorcraft, 2005.

[9] D.R. Nelson, D.B. Barber, T.W. McLain and R.W. Beard, "Vector Field Path Following for Miniature Air Vehicles", IEEE Transactions on Robotics, 2007.

[10] B.S. Kim, J.G. Lee and H.S. Han, "Biased PNG Law for Impact with Angular Constraint", IEEE Transactions on Aerospace and Electronic Systems, 1998.

[11] P. Zarchan, "Tactical and Strategic Missile Guidance", AIAA, 2002.

Copyright Statement