

Optimal guidance and observer design for target tracking using bearing-only measurements [★]

Nicolas Merlinge^{*} Julien Marzat^{*} Léon Reboul^{*}

^{*} ONERA - The French Aerospace Lab
(e-mail: first name . last name @onera.fr).

Abstract: This paper addresses the issue of target tracking from a UAV equipped with an embedded passive seeker (e.g. a camera). The considered target is a non-cooperative ground vehicle whose trajectory is unknown. We propose a method combining monocular observation and Model Predictive Control (MPC) that estimates the target state vector while following the trajectory maximizing its observability. The proposed estimation method is a set-membership particle filter. Its numerical performances are compared to a typical extended Kalman filter (EKF) and result in an enhanced robustness.

Keywords: Particle Filter, Set-Membership Estimation, Model Predictive Control, Bearing-only tracking.

1. INTRODUCTION

Efficient characterization of the trajectory of a non cooperative target using remote embedded sensor requires the definition of guidance and estimation techniques. Possible applications are ground or sea vehicles monitoring in specific areas, vehicle pursuit, and ground target remote sensing data. The methods should be jointly selected to take into account the physical limitations of the seeker and the constraints linked to the feasibility of the carrier trajectory. The problem considered in this paper is as follows. A carrier vehicle called the chaser in what follows aims to estimate the state of a ground vehicle called the target, using a passive bearing-only embedded seeker (for example an optical sensor). Such a seeker presents many practical interests: low energy consumption, low-mass sensors, and stealth. However, the only available information about the target is its relative direction with respect to the chaser. As the range is not accessible, estimation of the target state (e.g. position and velocity) requires specific chaser maneuvers coupled with an appropriate estimation method. The guidance law aims to determine these maneuvers so as to enhance the observability of the target state. The state estimation method to be considered should be able to handle measurements depending non linearly on the state vector. Several approaches can be considered. The most commonly used methods are probabilistic estimators, such as the Extended Kalman Filter (EKF) or the Sequential Importance Resampling Particle Filter (SIRPF Ristic (2004)). However, these approaches rely on prior probabilistic descriptions of measurement and state errors. This is not suitable when these distributions are unknown, as can be the state evolution of an unknown non-holonomic target.

Another way of describing uncertainty is defining bounds

or intervals within which these values are assumed to remain (set-membership estimation Jaulin (2009)). However the bearing-only measurement is nonlinear, and most of the bounded-estimation filters are dedicated to linear models. For nonlinear estimation, a recent approach called Box Particle Filter (BPF Abdallah et al. (2007)) has been introduced. This method links the stochastic approach with the set-membership determinism. Indeed, the BPF draws a cloud of possible bounded sets in the state-space called Box Particles. The measurement is also considered as a known bounded set. As pointed out previously, the design of the guidance law for the chaser aims to increase the observability of the target state vector. This could be achieved by searching for the law that maximizes an observation criterion. A straightforward approach is using the estimate filter estimate covariance matrix, which is related to the Fisher information matrix (Aidala (1979)). Model predictive control (MPC) method (Morari and Lee, 1999) determines control inputs resulting from a minimization of a given cost criterion. However, the search for the optimal values may prove highly time consuming. A potential approach to avoid this drawback is computing a minimum of the criterion by evaluating a discrete set of possible values (Rochefort et al., 2014). This approach cannot guarantee the optimality of the result but presents the advantage of fixing the dedicated computation time. In order to drastically decrease the computation budget, it is also worthy to fix *a priori* the structure of the control law. A suitable candidate for the guidance law structure is the pursuit guidance law (Shneydor, 1998) which is known to be well-suited for target tracking. However, this law may result in loss of observability.

In this paper, the proposed guidance law combines weighted pursuit guidance and lateral maneuver. The weights are determined by a discretized MPC module to minimize the observation criterion. Section 2 presents the bearing-only tracking problem. Section 3 describes several

^{*} The authors would like to thank the COGENT Computing lab (Coventry University) for their financial support.

tracking estimation techniques. Section 4 describes the proposed MPC strategy used to enhance the estimation accuracy. Section 5 presents the simulation results.

2. PROBLEM STATEMENT

In this paper, a UAV (the chaser) equipped with an optical embedded sensor aims at following the trajectory of a ground vehicle (the target) while estimating the target position and velocity. The sensor provides the direction of the line-of-sight (i.e. the line connecting the centers of gravity of the ground vehicle and the UAV) as long as the vehicle remains within its field of view. In order to describe the dynamical equations and observations, the following frames will be used. The local reference frame (*LRF*) is the inertial frame. The assumption is made that the Earth can be locally approximated by a tangent plane. The reference frame *LRF* is then (${}^{LRF}\mathbf{e}_1, {}^{LRF}\mathbf{e}_2, {}^{LRF}\mathbf{e}_3$) along the East, North, and up directions. The *LRF* origin is set at altitude 0 (sea level). The body frame (*BF*) of the chaser has its origin at the chaser's center of mass and its attitude in the *LRF* is defined by the rotation matrix $\mathbf{B}_{LRF \rightarrow BF}$. The sensor frame *SF* is assumed to be fixed to the body frame. Its attitude in the body frame is defined by a constant rotation matrix $\mathbf{B}_{BF \rightarrow SF}$.

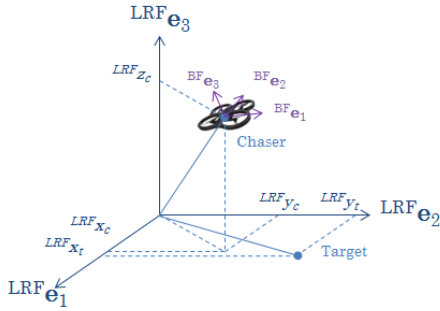


Fig. 1. Body frame and LRF frame

2.1 Dynamical model of the target

The target is represented by its center of mass. Its state vector in *LRF* consists of:

$\mathbf{X}_{t(k)} = [x_{t(k)} \ y_{t(k)} \ z_{t(k)} \ V_t^x(k) \ V_t^y(k) \ V_t^z(k)]^T$
with $(x_{t(k)}, y_{t(k)})$ the horizontal positions, $z_{t(k)}$ the altitude, and $V^{x,y,z}(k)$ the velocity components at time-step k .

The dynamical target model represents a random walk behavior and is as follows:

$$\mathbf{X}_{t(k)} = \mathbf{F}_k \mathbf{X}_{t(k-1)} \quad (1)$$

where \mathbf{F}_k is the state evolution matrix:

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 dt \\ \mathbf{O}_3 & \mathbf{R}_3(\alpha_k) \end{bmatrix} \quad (2)$$

and

$$\mathbf{R}(\alpha_k) = \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) & 0 \\ \sin(\alpha_k) & \cos(\alpha_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ with } \alpha_k \sim \mathcal{N}(0, \sigma_\alpha^2)$$

2.2 Chaser navigation and flight control

The chaser navigation module is based on an inertial navigation sensor (INS). As it derives, it has to be updated by additional measurements. The assumption is made that the navigation information is periodically available with a given uncertainty. In what follows, only the navigation filter output is considered (position, velocity, attitude, and their uncertainties). The chaser state vector at time-step k is noted $\mathbf{X}_{c(k)} = [x_{c(k)} \ y_{c(k)} \ z_{c(k)} \ V_c^x(k) \ V_c^y(k) \ V_c^z(k) \ \psi_{c(k)} \ \theta_{c(k)} \ \phi_{c(k)}]^T$ with $\psi_{c(k)}$, $\theta_{c(k)}$, and $\phi_{c(k)}$ the Euler angles defining the chaser attitude (yaw, pitch, roll). Let $d_c = 9$ be the chaser state dimension.

2.3 Target remote sensing

The target observation is performed using a strap-down embedded monocular optical sensor. The position of the camera (sensor frame s) in the BF frame is assumed to be fixed and known. It is expressed by the rotation matrix $\mathbf{B}_{SF \rightarrow BF}$. The chaser is assumed to fly at a fixed altitude. This is a simple and efficient way to prevent it to crash on ground obstacles. Then, it remains on a plane named tracking plane or chaser's plane. The measurements are the relative angular direction of the target with respect to the chaser. In what follows, the measured relative direction of the target is named the line of sight (LOS). The proposed strategy consists of a projection of the LOS in the chaser's tracking plane. If the chaser trajectory is not colinear with the projected LOS, the instantaneous crossing point of the successive projected LOS coincides with the projection of the target on the chaser's plane. This projection makes it possible to estimate the target geographical position and velocity. The geographical estimated state of the target is noted $\mathbf{X}_k \in \mathbb{R}^d$ ($d = 4$) and is defined as follows (3):

$$\mathbf{X}_k = [x_t(k) \ y_t(k) \ V_t^x(k) \ V_t^y(k)]^T \quad (3)$$

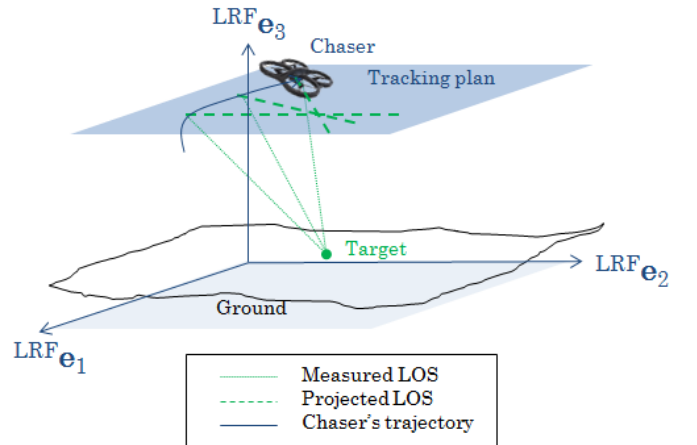


Fig. 2. Illustration of the target tracking principle

Real measured LOS Let $\chi_c = [x_c \ y_c \ z_c]^T$ and $\chi_t = [x_t \ y_t \ z_t]^T$ be respectively the chaser's position and the

target's real one. Likewise, \mathbf{V}_c and \mathbf{V}_t refer to their velocities.

The sensor is modeled as a typical pinhole camera with a bounded field of view. The field of view boundaries are modeled in the sensor frame as a domain delimited by a cone of rectangular section C_{FoV} . The camera's optical axis is the axis of symmetry of C_{FoV} and is noted ${}^{SF}\mathbf{e}_1$. The boundaries of C_{FoV} are $\pm\delta_1$ in azimuth and $\pm\delta_2$ in elevation.

As we use a monocular sensor, the information on the target is only its direction. We assume that the target detection is performed by an algorithm that provides the Line Of Sight (LOS) direction computed in the camera frame. The LOS between the target and the chaser is defined as the following normalized vector:

$${}^{LRF}\mathbf{LOS} = \frac{1}{\|\boldsymbol{\chi}_t - \boldsymbol{\chi}_c\|} (\boldsymbol{\chi}_t - \boldsymbol{\chi}_c) \quad (4)$$

The considered measurements are the angular orientation of the LOS in the sensor frame SF . We denote by m_1 the azimuth and m_2 the elevation angles. Azimuth and elevation at time k are defined as follows (see Fig 3):

$$\begin{bmatrix} m_1(k) \\ m_2(k) \end{bmatrix} = \begin{bmatrix} \text{atan} \left(\frac{{}^{SF}y_t(k) - {}^{SF}y_c(k)}{{}^{SF}x_t(k) - {}^{SF}x_c(k)} \right) \\ \text{atan} \left(\frac{{}^{SF}z_t(k) - {}^{SF}z_c(k)}{\sqrt{({}^{SF}y_t(k) - {}^{SF}y_c(k))^2 + ({}^{SF}x_t(k) - {}^{SF}x_c(k))^2}} \right) \end{bmatrix} \quad (5)$$

with ${}^{SF}\mathbf{X}_k = \mathbf{B}_{BF \rightarrow SF} \mathbf{B}_{LRF \rightarrow BF} \mathbf{X}_k$ ($\mathbf{B}_{LRF \rightarrow BF}$ is the rotation matrix describing the chaser attitude in the global frame LRF). In practice, this observation equation can be implemented with the $\text{atan2}()$ function.

As the angular measurements of the LOS are expressed in the sensor frame, the ${}^{LRF}\mathbf{LOS}$ vector can be written with respect to the measurements using equation (6).

$${}^{LRF}\mathbf{LOS} = \mathbf{B}_{body \rightarrow LRF} \mathbf{B}_{s \rightarrow body} \begin{bmatrix} \cos(m_2(k)) \cos(m_1(k)) \\ \cos(m_2(k)) \sin(m_1(k)) \\ \sin(m_2(k)) \end{bmatrix} \quad (6)$$

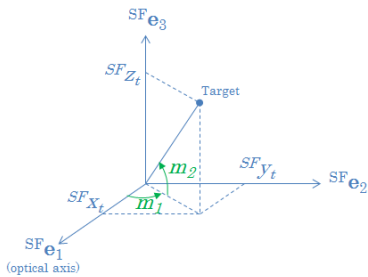


Fig. 3. Measurements in the sensor frame SF

Projected LOS The projected LOS is denoted by ${}^{LRF}\mathbf{d}$ in what follows.

$${}^{LRF}\mathbf{d} = \frac{1}{\|\prod_{\mathbf{e}_3} {}^{LRF}\mathbf{LOS}\|} \prod_{\mathbf{e}_3} {}^{LRF}\mathbf{LOS} \quad (7)$$

where $\prod_{\mathbf{e}_3}$ is the projector on the plane $z = z_c \in \mathbb{R}$.

Then, one can define the projected angular measurement that corresponds to the angle between the chaser speed and the projected LOS.

$$\mathbf{m} = [\arg({}^{LRF}\mathbf{d}, \mathbf{V}_c)] \quad (8)$$

Estimation algorithm The angular measurement \mathbf{m} of the LOS is the only available information about the target. The problem tackled in Section 3 is estimating the state vector (position and velocity) from this partial information. The assumption is made that the measurements are affected by an uncertainty that depends on the sensor resolution and the image processing algorithm precision.

Chaser guidance strategy The LOS measurement does not provide any information about the relative distance between the chaser and the target. As illustrated in Fig 2, the chaser speed must not be colinear with the projected LOS ${}^{LRF}\mathbf{d}$. Otherwise, the instantaneous crossing point of the successive projected LOS is not defined. In order to generate observability on the target position, the chaser has to maneuver. The maneuvers have to meet several constraints: keeping the target in the field of view, following it, and maximize the target observability. Section 4 presents an algorithm that aims to optimize the chaser's maneuvers while following the target.

Fig 4 gives an overview of the proposed system. An indication of the frequencies' orders of magnitude is provided. The chaser position, velocity, and attitude are assumed to be provided at 10Hz. The flight control is assumed to drive the actuators at 100Hz (also called low level control). The Target estimation frequency is set to 10Hz. The *tuning* and the *quality of estimation* will be defined in Section 4. The guidance update frequency can be adapted to the dynamics of the system (e.g. 1Hz).

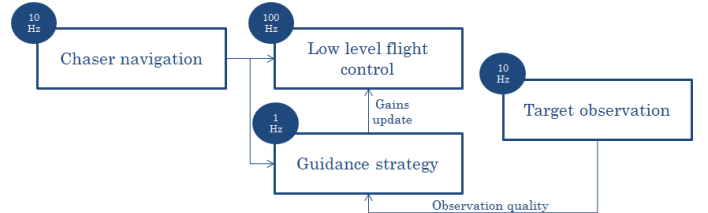


Fig. 4. System overview

3. TARGET TRACKING

In this section, two approaches to solve the target tracking problem are presented. The first one is the probabilistic approach. The most commonly used probabilistic estimator is the Extended Kalman Filter (EKF, Aidala (1979), Ristic (2004)). Although it is able to handle nonlinear models, it requires some strong hypotheses on the system uncertainties. Section 3.1 recalls the basics of the EKF. The second approach is set-membership estimation, which involves set manipulations. The advantage of this approach is that it ensures that the estimate belongs to a determined set. Section 3.2 recalls the basics of the set-membership estimation. However, these algorithms are often dedicated to linear systems, which makes them difficult to use in practice. Sections 3.3 and 3.4 present a mixed approach

that links the set-membership approach with the probabilistic estimation. In what follows, the state vector \mathbf{X} refers to the geographical state (3), and the measurement vector \mathbf{m} refers to the projected angular measurement of the LOS (8).

3.1 Classic probabilistic tracking estimation

The tracking algorithm aims to estimate the absolute state \mathbf{X}_k of the target. However, many aspects of the problem remain uncertain: The real target's evolution model is unknown, the optical sensor provides incomplete and noisy information on \mathbf{X}_k , and the chaser state (position, velocity, attitude) is estimated with a given uncertainty. Therefore, the tracking algorithm has to estimate \mathbf{X}_k from incomplete measurements, taking into account a variety of uncertainties. The estimate of \mathbf{X}_k at time-step k is usually written $\hat{\mathbf{X}}_k$. The most frequent approach is the probabilistic filters family (for example Kalman filters, Monte-Carlo Particle Filters). Those approaches often quantify the uncertainty on $\hat{\mathbf{X}}_k$ by a covariance matrix $\hat{\mathbf{P}}_k$.

Equation (1) describes the dynamical model of the target. However, this model is assumed to be unknown. Therefore, it has to be approximated with a simplified model. The probabilistic approach uses an additive noise called process noise to compensate the lack of knowledge. Let $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ be the assumed dynamical model. In our case, we will assume that $\forall \mathbf{X} \in \mathbb{R}^d$:

$$f(\mathbf{X}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 dt \\ \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{X} \quad (9)$$

Equation (5) describes the observation that can be performed with a given sensor. Let $g: \mathbb{R}^d \rightarrow \mathbb{R}^{d_{meas}}$ be the observation function (d_{meas} is the measurement dimension, here equal to one). It can be written as $\mathbf{m}_k = g(\mathbf{X}_{k-1})$. As the sensor and the detection algorithm are assumed to have a limited precision, one can use an additive measurement noise to model it.

The dynamical and observation equations considered in the tracking algorithm can be rewritten as follows:

$$\begin{cases} \mathbf{X}_k = f(\mathbf{X}_{k-1}) + \mathbf{W}_k \\ \mathbf{m}_k = g(\mathbf{X}_k) + \mathbf{V}_k \end{cases} \quad (10)$$

with $\mathbf{W}_k \sim \mathcal{L}_w$ being the process noise, and $\mathbf{V}_k \sim \mathcal{L}_v$ being the measurement noise. \mathcal{L}_w and \mathcal{L}_v are two probability distributions.

The state estimation consists in two steps: prediction and correction. Let $\mathbf{M}_k = \{\mathbf{m}_1, \dots, \mathbf{m}_k\}$ be the vector of measurements from time 1 to time k . The prediction step determines a predictive distribution $p(\mathbf{X}_k | \mathbf{M}_{k-1})$ with respect to the dynamical model uncertainty \mathcal{L}_w (also referred as $p(\mathbf{X}_k | \mathbf{X}_{k-1})$) and the previous conditional distribution (estimation density at time $k-1$ denoted by $p(\mathbf{X}_{k-1} | \mathbf{M}_{k-1})$). The correction step determines the posterior distribution (estimation density at time k denoted by $p(\mathbf{X}_k | \mathbf{M}_k)$) of the state with respect to the predictive distribution and the measurements \mathbf{m}_k and their uncertainty \mathcal{L}_v . Using $p(\mathbf{X}_k | \mathbf{M}_k)$, one can deduce the state estimation $\hat{\mathbf{X}}_k = E[\mathbf{X}_k | \mathbf{M}_k]$ where $E[\cdot]$ is the expectation (see Fig 5).

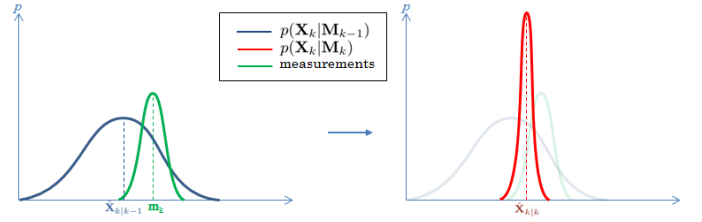


Fig. 5. Bayesian estimation

Extended Kalman Filter (EKF) The EKF relies on a local linearization of the state evolution and of the measurement equation. Therefore, system (10) has to be locally linearizable. Thus, a Jacobian computation of f and g functions is needed:

$$\begin{cases} \mathbf{J}_{f_k} = \frac{\partial f}{\partial \mathbf{X}} \Big|_{\mathbf{X}_{k-1}} \\ \mathbf{J}_{g_k} = \frac{\partial g}{\partial \mathbf{X}} \Big|_{\mathbf{X}_{k-1}} \end{cases} \quad (11)$$

In our case, $\mathbf{J}_{f_k} = \mathbf{F}$.

Table 1. EKF algorithm

Prediction step	
Predicted estimation	$\hat{\mathbf{X}}_{k k-1} = f(\hat{\mathbf{X}}_{k-1 k-1})$
Predicted covariance	$\hat{\mathbf{P}}_{k k-1} = \mathbf{J}_{f(k)} \hat{\mathbf{P}}_{k-1 k-1} \mathbf{J}_{f(k)}^T + \mathbf{Q}$
Correction step	
Innovation residual	$\psi_k = \mathbf{m}_k - g(\hat{\mathbf{X}}_{k k-1})$
Innovation covariance	$\mathbf{S}_k = \mathbf{J}_{g(k)} \hat{\mathbf{P}}_{k-1 k-1} \mathbf{J}_{g(k)}^T + \mathbf{R}$
Kalman gain	$\mathbf{K}_k = \hat{\mathbf{P}}_{k k-1} \mathbf{J}_{g(k)}^T \mathbf{S}_k^{-1}$
Updated estimation	$\hat{\mathbf{X}}_{k k} = \hat{\mathbf{X}}_{k k-1} + \mathbf{K}_k \psi_k$
Updated covariance	$\hat{\mathbf{P}}_{k k} = (\mathbf{I}_d - \mathbf{K}_k \mathbf{J}_{g(k)}) \hat{\mathbf{P}}_{k k-1}$

The main advantage of the EKF is its very low computation cost. The EKF solves the estimation problem assuming that uncertainties on the dynamical model and on the measurement model can be modeled by Gaussian noises (process noise and measurement noise). Fig 6 illustrates the estimation process. The assumption is also made that the posterior density $p(\mathbf{X}_k | \mathbf{m}_k)$ is also Gaussian. The uncertainty propagation and correction is performed using the Riccati equation. Although those conditions are met in a variety of situations, the EKF remains unstable in case of low observability, strong non-linearity, or high initial uncertainty. Table 1 summarizes the algorithm process, and Table 2 summarize EKF hypotheses and outputs.

Table 2. Extended Kalman Filter hypotheses and outputs

Hypotheses	
Dynamical model	Locally linearizable
Observation model	Locally linearizable
\mathcal{L}_w	Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{Q})$
\mathcal{L}_v	Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{R})$
Posterior	Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{P}_k)$
Outputs	
Estimation	$\hat{\mathbf{X}}_k$
Uncertainty covariance	$\hat{\mathbf{P}}_k$

3.2 Set-membership tracking

Another way to deal with system (10) is the set-membership state estimation approach. This approach

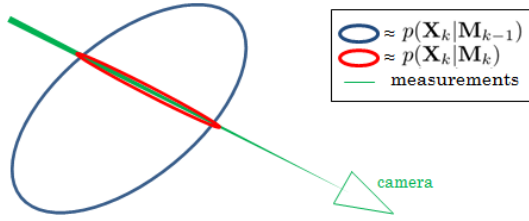


Fig. 6. Illustration of the EKF process

aims to include all the possible states \mathbf{X} and measurements \mathbf{m} into their respective including sets $\{\mathbf{X}\}$ and $\{\mathbf{m}\}$ (Jaulin, 2009).

Like a Bayesian filter, a set-membership state estimator is designed in two steps (Rassi et al., 2009): the prediction step propagates the set $\{\mathbf{X}_{k-1|k-1}\}$, containing all the possible states, using the dynamical model. Then, the correction uses the current measurement set $\{\mathbf{m}_k\}$ to correct the predicted set $\{\mathbf{X}_{k|k-1}\}$. Usually, the correction operation is performed using an inversion of the measurement equation: the set $\{\mathbf{Z}_k\} = g^{-1}(\{\mathbf{m}_k\})$ contains the possible states that can be explained by the current measurement (given its uncertainty) $\{\mathbf{m}_k\}$. Then, the contraction is made by an intersection operation:

$$\{\mathbf{X}_{k|k}\} = \{\mathbf{X}_{k|k-1}\} \cap \{\mathbf{Z}_k\}.$$

This approach makes it possible to describe a *guaranteed* set $\{\mathbf{X}_{k|k}\}$ that contains the real state \mathbf{X}_k . Fig 7 illustrates the estimation theory.

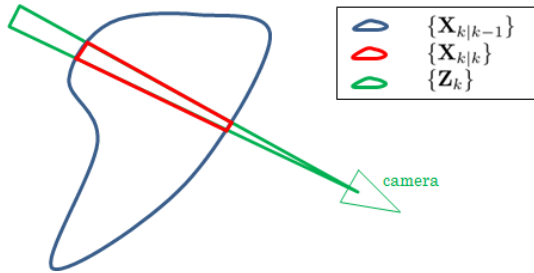


Fig. 7. Illustration of the Set-Membership estimation theory

3.3 Interval analysis

Among the different set-membership approaches, the Interval Analysis theory (Moore et al., 2009) is one of the more computationally convenient. Indeed, the choice is made to define uncertainties as bounded intervals. The simplicity of this approach makes it able to tackle numerical computing issues. The Interval Analysis theory defines several interval operations recalled in Table 3. In this paper, we will use the concept of box. A box $[\mathbf{a}] \in \mathbb{I}\mathbb{R}^\delta$ ($\delta \in \mathbb{N}$) is a vector of intervals that defines the set of all possible values in an hyperrectangle of the space \mathbb{R}^δ . Boxes can be defined in the state space as $[\mathbf{X}] \in \mathbb{I}\mathbb{R}^d$ as well as in the measurement space as $[\mathbf{m}] \in \mathbb{I}\mathbb{R}^{d_{meas}}$. However, this approach is very pessimistic, as every set, whatever be its shape, is roughly approximated by a hyperrectangular box. Therefore, the target dynamical equations and the observation equations can be written with this formalism. System (10) can be rewritten as follows:

$$\begin{cases} [\mathbf{X}_k] = [f]([\mathbf{X}_{k-1}]) + [\mathbf{W}_k] \\ [\mathbf{m}_k] = [g]([\mathbf{X}_k]) + [\mathbf{V}_k] \end{cases} \quad (12)$$

where $[f]$ and $[g]$ are inclusion functions. An inclusion function $[ψ]$ is defined as $[ψ] : \mathbb{I}\mathbb{R}^{\delta_1} \rightarrow \mathbb{I}\mathbb{R}^{\delta_2}$ the minimal box $[ψ]([\mathbf{a}]) \in \mathbb{I}\mathbb{R}^{\delta_2}$ containing the image $ψ([\mathbf{a}])$ of a box $[\mathbf{a}] \in \mathbb{I}\mathbb{R}^{\delta_1}$ by the function $ψ : \mathbb{R}^{\delta_1} \rightarrow \mathbb{R}^{\delta_2}$ (with $(\delta_1, \delta_2) \in \mathbb{N}^*$). The process and the measurement uncertainties are $[\mathbf{W}_k] \in \mathbb{I}\mathbb{R}^d$ and $[\mathbf{V}_k] \in \mathbb{I}\mathbb{R}^{d_{meas}}$. Like typical noises, their diameter has to be tuned using the sensor uncertainty and the level of dynamical model approximation.

Table 3. Interval Analysis operations

Description	notation
interval	$[a] = [\underline{a}, \bar{a}] = \{x \in \mathbb{R}, \underline{a} < x < \bar{a}\} \in \mathbb{I}\mathbb{R}$
addition	$[a] + [b] = [\underline{a} + \underline{b}, \bar{a} + \bar{b}]$
subtraction	$[a] - [b] = [\underline{a} - \bar{b}, \bar{a} - \underline{b}]$
multiplication	$[a] * [b] = [\min(\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}), \max(\underline{a}\bar{b}, \bar{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\bar{b})]$
division	$[a]/[b] = [\underline{a}, \bar{a}] * [1/\bar{b}, 1/\underline{b}]$ if $0 \notin [b]$
diameter	$ [a] = \bar{a} - \underline{a} \quad (\in \mathbb{R})$
box	$[\mathbf{a}] = [\underline{\mathbf{a}}, \bar{\mathbf{a}}] = [a_1] \times [a_2] \times \dots \times [a_n] \in \mathbb{I}\mathbb{R}^n$
volume	$ [\mathbf{a}] = \prod_{j=1}^n [a_j] \quad (\in \mathbb{R})$
center	$\mathbf{C} = \frac{1}{2}(\underline{\mathbf{a}} + \bar{\mathbf{a}}) \quad (\in \mathbb{R}^d)$

Although the Interval Analysis approach is a very deterministic way of designing an observer, a Bayesian interpretation can be formulated. Indeed, intervals can be described as uniform density kernels whose support is the interval itself. The reader will find some developments about the relations between set-membership filters and Bayesian filters in (Gning et al., 2013) and (DeFreitas et al., 2016). Furthermore, the interval formulation makes it possible to consider only the support bounds of a distribution. Therefore, there are very few hypotheses on the problem uncertainties: values are only assumed to remain into known bounds. However, this approach can mostly be used in linear applications. In the next subsection, we present a method that takes advantage of both the simplicity of set-membership estimation and of Bayesian filtering to tackle nonlinear problems.

3.4 Box Regularized Particle Filter: A stochastic set-membership estimator

A recent approach named Box Particle Filter (BPF) has been introduced in (Abdallah et al., 2007). The key idea is taking advantage of the simplicity of the set-membership approach as well as of a stochastic comprehensiveness. The method is inspired of Particle Filters (also known as Monte-Carlo filters) that try to provide an exhaustive description of the posterior density of the state space given the measurements. Applications to tracking have been presented in (Gning et al. (2012), Gning et al. (2013)). However, those applications examples do not deal with low-observability. Indeed, the proposed approaches assume that the range is observable (radar, lidar, sonar sensing methods, or triangulation measurements). The BPF is a translation of the conventional particle filters (PF) into the set-membership theory (Moore et al., 1979). In order to explore the state space more exhaustively, the BPF draws a cloud of weighted box particles that are propagated and contracted as explained in the previous subsection. However, because several box-hypotheses are considered,

a selection can be made to keep only the more likely boxes. This allows the algorithm to converge on the dimensions that are not explicitly linked to the measurements, as the velocity in our case. Each box particle is weighted by a weight that quantifies its likelihood. The authors have recently proposed another version of this filter, called Box Regularized Particle Filter (BRPF, Merlinge et al. (2016)).

Initialization The BRPF is initialized performing a sub-paving of a box area delimited by the initial uncertainty box $[\mathbf{P}_0]$, that delimits the possible initial target space. Each box particle $i \in \llbracket 1, N \rrbracket$, as part of the sub-paving, is associated with a weight w_k^i . At the initialization step ($k = 0$), all the particles have the same weight: $w_k^i = \frac{1}{N} \forall i \in \llbracket 1, N \rrbracket$.

Prediction step The prediction step is performed using the first equation of (12) for each particle: $[\mathbf{X}_k^i] = [f](\llbracket \mathbf{X}_{k-1}^i \rrbracket) + [\mathbf{W}_k]$. Since the box particles are vectors of intervals, the propagation using interval analysis propagates the velocity uncertainties on the position dimensions.

Correction step In the box particle filter, the correction step is performed in two parts. To begin, each particle is intersected with a set of the state space explained by the measurements. This set is obtained by a current measurement set inversion. This leads to new box particles: $[\mathbf{X}_k^i]^{new} = [\mathbf{X}_k^i] \cap [g^{-1}](\llbracket \mathbf{m}_k \rrbracket)$. Then, the weights are updated with respect to the measurements. The weights are updated using the contraction ratio of each particle (Gning et al., 2013): $\mathbf{w}_k^i = \frac{|\llbracket \mathbf{X}_k^i \rrbracket^{new}|}{|\llbracket \mathbf{X}_k^i \rrbracket|} \mathbf{w}_{k-1}^i$ and then normalized.

Resampling step Particle Filters in general suffer a degeneracy problem (Ristic, 2004). After several iterations, all but one particle remain with a negligible weight. In order to enhance the approximation of the posterior density $p(\mathbf{x}_k | \mathbf{M}_k)$, one can replace the low-weighted particles with new particles cloned from the high-weighted particles. This operation can be triggered when too many particles are low-weighted. Several numerical criteria can be used for that purpose. The N_{eff} criterion is used here. It consists in computing the number of efficient particles N_{eff} with respect to their weights, $N_{eff} = \sum_i \frac{1}{w_k^i}$. If $N_{eff} < \theta_{eff} N$ with $\theta_{eff} \in [0, 1]$ a tuned coefficient, the resampling operation is triggered. For the BRPF, the resampling step is performed as in a conventional particle filter. However, instead of cloning the particles that are the most consistent with the current measurement, these box particles are subdivided along one dimension (Fig 9). This makes it possible to increase the resolution of the boxes that have the higher likelihood (related to weights, illustrated in bluescale). However, a subdivision of a box particle implies a choice of the cutting dimension $d_{cut} \in \llbracket 1, d \rrbracket$. An efficient way to select the cutting dimension d_{cut} consists in determining the one that corresponds to the maximal local uncertainty. Thus, the selected dimension is the one along which the box particle is the longest. In order to take into account inhomogeneity of state components, it is necessary to normalize the box diameter vector. Hence, we work with a set of homogeneous and normalized box diameters. The normalization is based on the physical

dimension of each state dimension. A state vector can be mapped with several sub-vectors defined by their physical dimensions (position, speed,...). We have chosen to normalize a state vector diameter with the Euclidean norm of its sub-vectors. Fig 8 illustrates the estimation process. The reader will notice that the BRPF representation can be interpreted either as a set intersection approximation (Fig 7) or as a Bayesian density convolution (Fig 6).

Regularization operation The discretization of the approximation of the posterior density $p(\mathbf{X}_k | \mathbf{M}_k)$ can lead to an exploration of a wrong area and an erroneous estimate. One can explore the state space with a more regular description by providing a kernel to each particle. Numerically, this kernel is obtained applying a bounded random noise on the particles. This process, called regularization, was presented in (Musso et al., 2001) to improve the robustness of conventional particle filters. The authors have proposed an adaptation of the regularization method to the Box Particle Filter in (Merlinge et al., 2016). The Box Particle Filter process, hypotheses and outputs are recalled in Table 4 and Table 5. The center of a box $[\mathbf{X}_k^i]$ is denoted by \mathbf{C}_k^i .

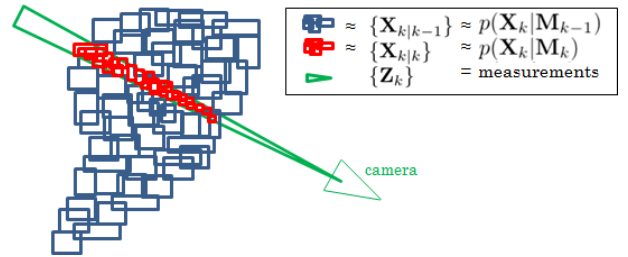


Fig. 8. Illustration of the BRPF process

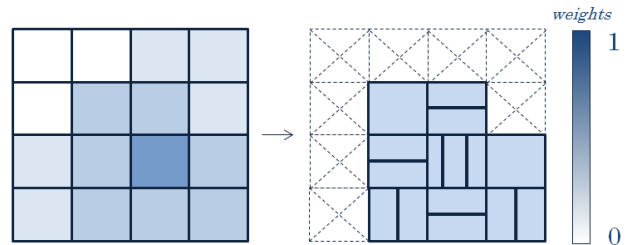


Fig. 9. Illustration of resampling by subdivision

An advantage of the BRPF over the EKF is that it explicitly takes into account all the system's uncertainties and their relationships: chaser's position, attitude, and measurements. The measurement noise only represents the observation uncertainty. The EKF, for its part, builds a predicted measurement as if there were no other uncertainty than the measurement noise. As the measurements depend on the chaser's position and attitude, their uncertainties are taken into account by an empirical increase of the measurement noise. The interval analysis makes it possible to take them into account in a deterministic way.

4. CHASER GUIDANCE USING MODEL PREDICTIVE CONTROL

In this section, the chaser's control input is first defined. The design of the guidance law aims to increase the

Table 4. Box Regularized Particle Filter algorithm

Prediction step	$\forall i \in \llbracket 1, N \rrbracket$
Particles propagation	$[\mathbf{X}_k^i] = [f](\llbracket \mathbf{X}_{k-1}^i \rrbracket) + [\mathbf{W}_k^i]$
Correction step	
Contraction	$[\mathbf{X}_k^i]^{new} = [\mathbf{X}_k^i] \cap [g^{-1}](\llbracket \mathbf{m}_k \rrbracket)$
Likelihood	$\lambda_k^i = \frac{ \llbracket \mathbf{X}_k^i \rrbracket^{new} }{ \llbracket \mathbf{X}_k^i \rrbracket }$
Updated weights	$\mathbf{w}_k^i = \lambda_k^i \tilde{\mathbf{w}}_{k-1}^i$
Normalized weights	$\tilde{\mathbf{w}}_k^i = \frac{\mathbf{w}_k^i}{\sum_{j=1}^N \mathbf{w}_k^j}$
Resampling step	if $N_{eff} < \theta_{eff} N$
Subdivision	Resampling by subdivision on a determined dimension for each box particle d_{cut}
Regularization	Randomly moving the box particles using bounded kernels.

Table 5. BRPF outputs and hypotheses

Hypotheses	
Dynamical model	Nonlinear
Observation model	Nonlinear
\mathcal{L}_w	Only a bounded support
\mathcal{L}_v	Only a bounded support
Posterior	Any (discrete approximation)
Outputs	
Estimation	$\hat{\mathbf{X}}_k = \sum_{i=1}^N w_k^i \mathbf{C}_k^i$
Uncertainty covariance	$\hat{\mathbf{P}}_k = \sum_{i=1}^N w_k^i (\hat{\mathbf{X}}_k - \mathbf{C}_k^i)(\hat{\mathbf{X}}_k - \mathbf{C}_k^i)^T$
Other moments	$\mathbf{M}_{3(k)}, \mathbf{M}_{4(k)} \dots$
Uncertainty hull	$[\mathbf{P}_k] = [\underline{\mathbf{U}}_k, \overline{\mathbf{U}}_k]$ with $\mathbf{U}_k = \bigcup_{\forall i w_k^i \neq 0} [\mathbf{X}_k^i]$

observability of the target state vector. This could be achieved by searching for the control input that maximizes an observation criterion. This criterion must quantify the estimation efficiency. A straightforward approach is using the filter estimate covariance matrix, which is related to the Fisher information matrix (Aidala (1979)). Model predictive control (MPC) method (Morari and Lee, 1999) determines control inputs resulting in a minimization of a given cost criterion. However, the search for the optimal values using a typical iterative algorithm may prove highly time consuming. A potential approach to avoid this drawback is computing a minimum of the criterion by evaluating a discrete set of possible values (Rochefort et al., 2014). This approach cannot guarantee the optimality of the result but presents the advantage of fixing the dedicated computation time. Furthermore, the choice has been made to fix *a priori* the structure of the control law. This makes it possible to guarantee the tracking behavior whatever be the MPC outputs. A suitable candidate for the guidance law structure is the pursuit guidance law (Shneydor, 1998) which is known to be well-suited for target tracking. However, this law results in loss of observability. In this paper, the proposed guidance law combines weighted pursuit guidance and lateral maneuver. The weights are determined by a discretized MPC module to minimize the observation criterion.

4.1 Control input

The chaser's goal is twofold: following the target and estimating its state vector \mathbf{X} as precisely as possible. This implies several constraints that have to be taken

into account in the control law: keeping the target in the sensor's field of view, which implies distances constraints, and maneuver to reduce the magnitude of the error covariance. The UAV control input requires two components: The desired acceleration and the desired yaw angle. Let $\mathbf{u} \in \mathbb{R}^3$ and $\psi_d \in \mathbb{R}$ respectively be the acceleration input and the angular input. Let $U = [\mathbf{u}^T \psi_d]^T \in \mathbb{R}^{d_{cont}}$ be the whole control input with $d_{cont} = 4$ the control dimension.

Following control input The first constraint to be dealt with is keeping the target in the field of view. One can define the desired angle ${}^{LRF}\psi_d$ to track the line of sight ${}^{LRF}\mathbf{LOS}$ in the LRF frame.

$${}^{LRF}\psi_d = \arg({}^{LRF}\mathbf{d}, {}^{LRF}\mathbf{e}_1) \quad (13)$$

Then, the acceleration has to be designed to follow the target while keeping a constant altitude. A possible design inspired from the typical missile pure pursuit control law is (Shneydor, 1998):

$$\mathbf{u}_1 = \kappa_1 \mathbf{V}_c \times ({}^{LRF}\mathbf{d} \times \tilde{\mathbf{V}}_c) \quad (14)$$

where ${}^{LRF}\mathbf{d}$ is the projection of ${}^{LRF}\mathbf{LOS}$ in a constant altitude plane (\times refers to the cross product) and $\kappa_1 \in \mathbb{R}^+$ is a gain. $\tilde{\mathbf{V}}_c = \frac{1}{\|\mathbf{V}_c\|} \mathbf{V}_c$ is the normalized chaser velocity in the inertial frame. This control input tends to maintain \mathbf{V}_c colinear with ${}^{LRF}\mathbf{d}$.

Lateral maneuvering control input As the observation is only performed on the *LOS* direction, there is no observation on the range value. Therefore, the chaser has to maneuver to enhance the observability on the range. For safety reasons, we force the chaser trajectory to remain into a constant altitude plane. The direct orthogonal vector to ${}^{LRF}\mathbf{d}$ contained in this plane is:

$${}^{LRF}\mathbf{d}^\perp = {}^{LRF}\mathbf{e}_3 \times {}^{LRF}\mathbf{d} \quad (15)$$

Therefore, one can define the following control input:

$$\mathbf{u}_2 = \kappa_2 \mathbf{V}_c \times ({}^{LRF}\mathbf{d}^\perp \times \tilde{\mathbf{V}}_c) \quad (16)$$

This control input tends to maintain \mathbf{V}_c colinear with ${}^{LRF}\mathbf{d}^\perp$ (with $\kappa_2 \in \mathbb{R}$). In practice, κ_2 is bounded by the chaser dynamics.

Speed maneuvering control input As the target speed is unknown at the beginning, the chaser velocity norm has to be adjusted. This requires a third control input that aims to change the chaser velocity norm (tuned by a gain $\kappa_3 \in \mathbb{R}$):

$$\mathbf{u}_3 = \kappa_3 \tilde{\mathbf{V}}_c \quad (17)$$

In practice, κ_3 is bounded by the chaser dynamics.

As a result, the acceleration control input is:

$$\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_3 \quad (18)$$

4.2 Model Predictive Control (MPC)

We have defined a control input $\mathbf{u}(\kappa_1, \kappa_2, \kappa_3)$. The κ_i gains have to be tuned to make the control input efficient to achieve the tracking. In this control architecture, the control input will be periodically updated using MPC every T_{MPC} seconds. Those three control components are equivalent to three distinct behaviors:

- κ_1 sets the pursuit behavior of the chaser (without changing the velocity)
- κ_2 sets a lateral motion behavior that aims to enhance the target observability (without changing the velocity)
- κ_3 sets the chaser speed norm behavior

In our application, the pursuit behavior has to be maintained during the whole mission. Therefore, κ_1 can be fixed and does not require to be periodically updated. However, the lateral motion and the speed norm have to be adapted to the situation. Therefore, coefficients κ_2 and κ_3 will be periodically updated. This section describes the proposed strategy to optimize the couple of gains (κ_2, κ_3).

Model Predictive Control MPC relies on an optimization of a cost function J with respect to possible control sequences. In our case, the MPC module will choose the couple $(\kappa_2, \kappa_3) \in \mathbb{R}^2$ that minimizes $J(\kappa_2, \kappa_3)$ over a prediction period $k_p \in \llbracket k+1, k+K_h \rrbracket$, with K_h the prediction horizon. The time-step used in the prediction loop is the same as the observer (dt). Figure 10 illustrates the MPC process. A control sequence is periodically chosen among a set of candidate trajectories. The number N_u of candidate control sequences (κ_2, κ_3) is set offline. Therefore, the computation cost is guaranteed to be known and constant for the whole optimization process.

Proposed MPC architecture The chosen control sequence has to meet several constraints on the chaser's state. The constraints considered in this article are listed in Table 6.

Table 6. Model Predictive Control constraints

Constraints	Associated numerical criterion	Example
Chaser's flight domain	Acceptable velocity $\ \mathbf{V}_c\ \in [V_c^{accept}]$	[20, 100] km/h
Target kept into the FoV	$\begin{bmatrix} m_1(k) \\ m_2(k) \end{bmatrix} \in [\mathbf{C}_{FOV}]$	$\begin{bmatrix} [-40^\circ, 40^\circ] \\ [-40^\circ, 40^\circ] \end{bmatrix}$
Estimated range	$\hat{r} = \ \hat{\mathbf{x}}_t - \mathbf{x}_c\ \in [r^{accept}]$	[50, 300]m

To meet those constraints, the following algorithm is proposed:

To begin, the target state is predicted over the prediction horizon (equation 19). The predicted target trajectory is independent from the candidate trajectory. Therefore, it can be computed once before the MPC optimization loop. However, for clarity reasons, Algorithm 1 presents this computation inside the MPC loop. When MPC is triggered, the current time-step is k . The prediction temporal loop is performed between time-step $k_p = k+1$ and time-step $k_p = k+K_h$.

$$\hat{\mathbf{X}}_{k_p} = f(\hat{\mathbf{X}}_{k_p-1}) \quad (19)$$

The associated uncertainty can be propagated using the Riccati equation (20). \mathbf{Q}_p is the covariance of the MPC prediction process noise.

$$\hat{\mathbf{P}}_{k_p|k_p} = \mathbf{F}\hat{\mathbf{P}}_{k_p-1}\mathbf{F}^T + \mathbf{Q}_p \quad (20)$$

For each i candidate control sequence, the chaser's state can be propagated. Let $f_c : \mathbb{R}^{d_c} \times \mathbb{R}^{d_{cont}} \rightarrow \mathbb{R}^d$ be the chaser's prediction model. f_c is a simplified model of the chaser's dynamical model. Indeed, the prediction of the

chaser's space has to be very fast to be performed $N_u \times K_h$ times in a minimum amount of time. The i^{th} candidate control sequence is $\mathbf{U}_i = \mathbf{U}_i(\kappa_2, \kappa_3)$.

$$\hat{\mathbf{X}}_{c(k_p)} = f_c(\hat{\mathbf{X}}_{c(k_p-1)}, \mathbf{U}_i) \quad (21)$$

The approximated dynamical model is:

$$f_c(\hat{\mathbf{X}}_{c(k_p-1)}, \mathbf{U}_i) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 dt & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{O}_3 & \mathbf{I}_3 \end{bmatrix} \hat{\mathbf{X}}_{c(k_p-1)} + \begin{bmatrix} \mathbf{O}_{3 \times 1} \\ \mathbf{u} dt \\ \Xi \end{bmatrix} \quad (22)$$

with $\Xi = \begin{bmatrix} dt K_\psi (LRF \psi_d - \hat{\psi}_{c(k_p-1)}) \\ 0 \\ 0 \end{bmatrix}$ where K_ψ is a gain comparable with the actual yaw flight control law.

Having computed the predicted state of the two objects, one can compute a predicted measurement $\hat{\mathbf{m}}_{k_p}$ using equation (5):

$$\hat{\mathbf{m}}_{k_p} = g(\hat{\mathbf{X}}_{k_p-1}, \hat{\mathbf{X}}_{c(k_p-1)}) \quad (23)$$

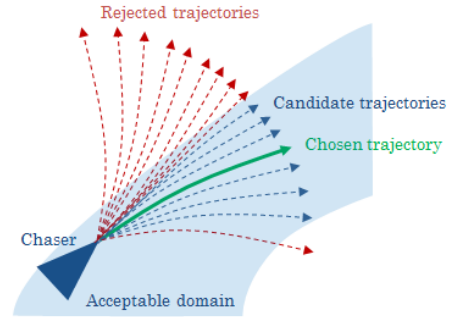


Fig. 10. Illustration of the Boolean domain conditions

The MPC algorithm has to choose the best trajectory among a given set of trajectories. The proposed strategy consists in eliminating the unacceptable trajectories using Boolean conditions (Figure 10). The acceptable trajectories are then evaluated with a numerical criteria. The trajectory associated to the minimal criteria is chosen. If all the trajectories have been rejected, a control input is set by default with respect to the cause of the whole rejection.

A first constraint that must be met by the trajectory is keeping the target in the chaser's field of view (FoV). The FoV is defined by a cone that can be written in the sensor frame with the interval analysis formalism: $[\mathbf{C}_{FOV}] = \begin{bmatrix} [-\delta_1 & \delta_1] \\ [-\delta_2 & \delta_2] \end{bmatrix}$ (see Section 2.3). Then, a Boolean condition can be defined, verified if the predicted measurement is in the field of view:

$$\hat{\mathbf{m}}_{k_p} \in [\mathbf{C}_{FOV}] \quad \forall k_p \in \llbracket k+1, k+K_h \rrbracket \quad (24)$$

For safety reasons, the chaser should remain in its flight domain during the whole trajectory. Therefore, the following Boolean condition on the predicted velocity can be defined:

$$\|\hat{\mathbf{V}}_{c(k_p)}\| \in [V_c^{accept}] \quad \forall k_p \in \llbracket k+1, k+K_h \rrbracket \quad (25)$$

A simple way to guarantee that the chaser will stay in a delimited domain around the target (assuming that the measurements are available) is eliminating trajectories that will drive it out of this domain. The proposed formulation relies on an acceptable range interval $[r^{accept}]$ within which the estimated range must remain. The trajectory

is eliminated if it does not satisfy the following Boolean condition:

$$\hat{r} \in [r^{accept}] \quad \forall k_p \in \llbracket k+1, k+K_h \rrbracket \quad (26)$$

If the previous Boolean conditions are met, one can define a refined selection criterion. This criterion is based on the target observability. It can be quantified using the predicted estimation uncertainty using the Riccati equation (Kalman formulation, see Table 2).

$$\begin{cases} \mathbf{S}_{k_p} = \mathbf{J}_{g(k_p)} \hat{\mathbf{P}}_{k_p-1|k_p-1} \mathbf{J}_{g(k_p)}^T + \mathbf{R} \\ \mathbf{K}_{k_p} = \hat{\mathbf{P}}_{k_p|k_p-1} \mathbf{J}_{g(k_p)}^T \mathbf{S}_{k_p}^{-1} \\ \hat{\mathbf{P}}_{k_p} = (\mathbf{I}_d - \mathbf{K}_{k_p} \mathbf{J}_{g(k_p)}) \hat{\mathbf{P}}_{k_p|k_p-1} \end{cases} \quad (27)$$

with $\mathbf{J}_{g(k_p)}$ obtained from equation (11) and \mathbf{R} a covariance matrix modeling the predicted observation uncertainty.

From this predicted covariance matrix $\hat{\mathbf{P}}_{k_p}$, one can quantify the observability. The proposed approach is based on the spectral norm of the covariance matrix. Numerically, it consists of a diagonalization strategy using the Singular Values Decomposition (SVD) method. The instantaneous observation criterion is noted j_{k_p} and is defined as the maximum of the covariance matrix's singular values. Indeed, the larger singular value of $\hat{\mathbf{P}}_{k_p}$ quantifies the total maximum uncertainty. This can be illustrated by the major axis of an ellipsoid defined by $\hat{\mathbf{P}}_{k_p}$ (Fig 11).

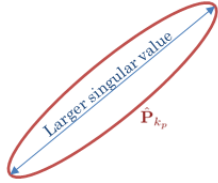


Fig. 11. Illustration of larger singular value of $\hat{\mathbf{P}}_{k_p}$

$$\begin{cases} (\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = \text{SVD}(\hat{\mathbf{P}}_{k_p}) \\ j_{k_p} = \max_i(\mathbf{\Sigma}_{i,i}) \end{cases} \quad (28)$$

If the trajectory has not been rejected, one can compute a criterion for the whole trajectory i :

$$J_i = \sum_{k_p} j_{k_p} \quad (29)$$

This criterion makes it possible to compare the i^{th} trajectory with the others acceptable trajectories. The following discrete optimization can be performed to find the trajectory that minimizes J_i :

$$(\kappa_2, \kappa_3) = \arg \min_{i=\{1, \dots, N_u\}} (J_i(\kappa_2, \kappa_3)) \quad (30)$$

However, the chaser and the target model used for the MPC prediction are very simplified models. While it is possible to monitor the chaser velocity at low level to prevent it to go out of its flight domain, one cannot guarantee that the other conditions are always met (unpredictable loss of measurement or too important predicted range). In such a situation, their predicted states can diverge from their real values and lead to a rejection of all the trajectories. This issue can be tackled using default control sequences that will be applied with respect to the cause of the whole set of trajectories rejection. Therefore, the following strategy is proposed. If the trajectories have all been rejected because

Result: Sub-optimal (κ_2, κ_3)

$\hat{\mathbf{P}}_k$ is the current covariance estimation

```

for  $i \in \llbracket 1, N_k \rrbracket$  do
  for  $k_p \in \llbracket k+1, k+K_h \rrbracket$  do
    Predict target state (19)
    Predict covariance (20)
    Predict chaser state (21)
    Predict measurement (23)
    if FoV condition (24) then
      if Speed condition (25) then
        if Range condition (26) then
          Compute the instantaneous MPC
            criterion (28)
        else
          Candidate trajectory  $i$  rejected
        end
      else
        Candidate trajectory  $i$  rejected
      end
    else
      Candidate trajectory  $i$  rejected
    end
  end
  Criterion for the whole trajectory  $i$  if not rejected (29)
end

```

Find the trajectory i associated with the corresponding (κ_2, κ_3) that minimizes the MPC criterion (30).

Algorithm 1: MPC architecture

of the FoV condition, the control input is automatically set to a research mode sequence that aims to recover the measurements.

$$\begin{cases} \mathbf{U} = \mathbf{0} \\ {}^{LRF} \psi_d = \delta_\psi k \end{cases} \quad (31)$$

If the trajectories have all been rejected because of the range condition, a following mode is set to take back the chaser into the acceptable range set.

$$\begin{cases} \kappa_2 = 0 \\ \kappa_3 = \pm \kappa_{3_{fm}} \\ {}^{LRF} \psi_d = \delta_\psi k \quad \forall k \in \mathbb{N} \end{cases} \quad (32)$$

where δ_ψ and $\kappa_{3_{fm}}$ are increasing coefficients to be set. If $\hat{r} < r^{accept}$, $\kappa_3 = \kappa_{3_{fm}}$ and $\kappa_2 = -\kappa_{3_{fm}}$ else.

5. NUMERICAL RESULTS

In this section, the coupled observation-guidance MPC strategy is tested in different configurations. The considered chaser is a typical quadrotor UAV (see Table 7). In our numerical simulations, the chaser is modeled with a realistic dynamical model and flight control.

Table 7. Chaser's features

Constructor	Parrot
Model	AR DRONE 2.0
Propulsion	Rotary wing
Number of motors	4
Type	electrical brushless 28,000 rpm
Maximum speed	18 km/h
Dimensions	451 × 451 mm
Mass	420 g
Optical sensor resolution	720 p
Field of view ($2\delta_1 \times 2\delta_2$)	92° × 92°

In order to compare the possible strategies and algorithms, the following criterion is used, evaluated for $N_{MC} \in \mathbb{N}$ Monte-Carlo simulations: the Root Mean Square Error (RMSE) between the estimation $\hat{\mathbf{X}}_k$ and the real target's state \mathbf{X}_k .

$$\mathbf{RMSE}_k = \sqrt{\frac{\sum_{i=1}^{N_{MC}} (\hat{\mathbf{X}}_k - \mathbf{X}_k)^2}{N_{MC}}} \quad (33)$$

The mean RMSE is $\mathbf{RMSE} = \frac{1}{N_k} \sum_{k=1}^{N_k} \mathbf{RMSE}_k$ with N_k the number of iterations of a simulation.

5.1 Interest of the MPC guidance strategy

In this paragraph, the following scenario is considered: the target moves in a straight line at constant speed and constant altitude. Three strategies are compared: the pure pursuit mode ($\kappa_1 = 1, \kappa_2 = 0, \kappa_3 = 0$), the random lateral mode ($\kappa_1 = 1, \kappa_2 \sim \mathcal{U}[\underline{\kappa}_2, \overline{\kappa}_2], \kappa_3 = 0$), and the MPC mode described in Section 4. In practice, κ_2 and κ_3 are bounded by the chaser dynamics. For the considered chaser, $\kappa_2 \in [\underline{\kappa}_2, \overline{\kappa}_2] = [-5, 5]$ and $\kappa_3 \in [\underline{\kappa}_3, \overline{\kappa}_3] = [-0.1, 0.1]$. Table 8 describes the simulation parameters. The estimation algorithm used in this test is the Box Regularized Particle Filter (BRPF).

Table 8. Simulation parameters

Initial chaser's state $[\mathbf{x}_{c(0)}^T \mathbf{V}_{c(0)}^T]^T$	$[0, 0, 10, 1, 0, 0]^T$ (m, m/s)
Initial target's state \mathbf{X}_0	$[5, 5, 1, 1]^T$ (m, m/s)
Initial state guest $\hat{\mathbf{X}}_0$	$[\mathbf{x}_{c(0)}^T, \mathbf{0}_{1 \times 3}]^T$
Initial estimation uncertainty $\hat{\mathbf{P}}_0$	$\pm[6\text{m}, 6\text{m}, 3\text{m/s}, 3\text{m/s}]^T$
Measurement uncertainty	$\pm[3^\circ, 3^\circ]$ (azimuth, elevation)
Chaser's position uncertainty	$\pm[0.3, 0.3, 0.3]\text{m}$
Chaser's attitude uncertainty	$\pm[3^\circ, 3^\circ, 3^\circ]$
Simulation duration	40s
MPC update frequency	0.33Hz
MPC horizon	6s
Observation frequency	10Hz
Admissible MPC chaser's range	$[5, 100]\text{m}$
Admissible MPC chaser's speed	$[0.1, 5]\text{m/s}$

Table 9 shows the mean RMSE errors obtained with the three strategies (100 Monte-Carlo runs). The MPC guidance strategy appears to perform far better than the pure pursuit strategy. The random lateral mode shows that a random choice of κ_2 is not sufficient to make the estimator converge. This underlines the necessity of an optimized control strategy.

Table 9. RMSE results: control strategies comparisons (100 simulations)

	Pure pursuit	random κ_2	proposed MPC strategy
RMSE x (m)	18.3	12.8	3.75
RMSE y (m)	18.6	13.7	2.73
RMSE Vx (m/s)	0.96	0.75	0.68
RMSE Vy (m/s)	0.98	0.78	0.73

5.2 Filter robustness comparison

This section compares the estimation performances of the Extended Kalman Filter (EKF) and the Box Regularized Particle Filter (BRPF) presented in section 3 using MPC

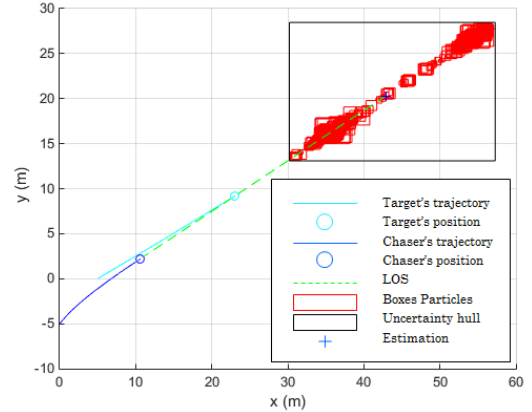


Fig. 12. Pure pursuit - estimation behavior

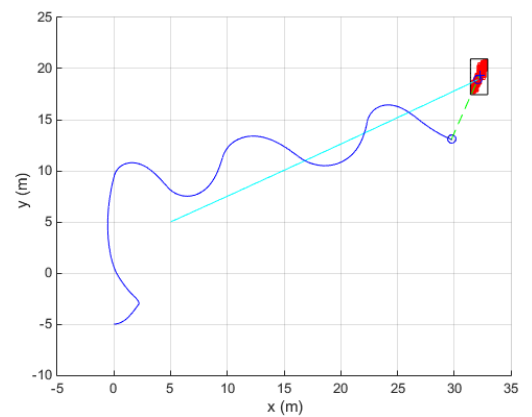


Fig. 13. MPC strategy - estimation behavior

guidance law. The proposed scenario is as follows: the target performs a random walk according to model (2). The random angle's standard deviation is $\sigma_\alpha = 10^{-2}$. The same trajectory has been used for all the simulations (Fig 14). For each simulation i , the mean error E_i is obtained as the average of all estimation errors $|\hat{\mathbf{X}}_k - \mathbf{X}_k|$ where k is the estimation time. Table 10 presents the mean of the errors obtained for the two filters for 100 simulations. The *best* column contains the lowest value of E_i . Likewise, the *worst* column contains the highest value of E_i . The RMSE value is the mean of the Root Mean Square Errors.

The main advantages of the EKF are a very low computation cost and an implementation simplicity. Furthermore, its Bayesian approach theoretically allows it to converge accurately on the maximum of likelihood. However, it suffers instabilities for bearing-only tracking applications (Aidala, 1979). This can result in a sudden and huge divergence that is often impossible to catch up. In particular, abrupt course alterations can make it diverge. In our results, the EKF RMSE is strongly degraded by those sudden divergences.

The BRPF, since it performs interval analysis operations, may be more pessimistic than a Bayesian filter. However, its mean *best* errors are comparable to the EKF's ones. Furthermore, it has been shown to be robust to strong non-linearities (Merlinge et al., 2016). This is confirmed in these results.

Table 10 also presents the mean computation time for each filter (for one estimation call). Those values have

been obtained with a desktop computer running Matlab. The *time* line underlines the EKF low computation cost (0.35ms) with respect to the BRPF (2.5ms). However, since the observation filter frequency is 10Hz, the real-time condition is met for both filters.

Table 10. Mean error results: EKF and BRPF comparison (MPC strategy, 100 simulations)

	EKF			BRPF		
	best	RMSE	worst	best	RMSE	worst
x (m)	0.609	37.9	125	0.576	3.53	15.6
y (m)	0.988	36.4	136	1.17	3.91	14.6
V _x (m/s)	0.299	9.28	28.5	0.417	0.844	2.47
V _y (m/s)	0.372	9.90	30.8	0.329	0.780	2.34
Time (ms)	0.35			2.5		

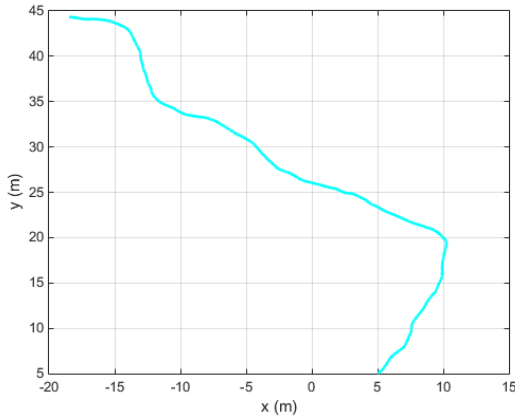


Fig. 14. Random walk target trajectory

5.3 Criterion evolution

This section illustrates the MPC behavior. Fig 15 shows a simulation of one MPC update step. The chaser has to choose one trajectory among a set of candidate trajectories. For this MPC update, several trajectories have been rejected because of the distance Boolean criterion. Indeed, they would have led the chaser too close to the target. Every uncertainty has been represented by a set: the chaser position, the target state estimation, and the measurements (the LOS, as it is a cone, can be formalized by an angular box). Fig 16 shows the criterion values used to perform the control update. On the left side is plotted the chaser and target trajectories, and the boxes used to perform the estimation (red rectangles) for the three first MPC updates. On the right side is plotted the corresponding criterion and its minimum (with respect to κ_2). This is the result of the simulation of all the trajectories using the target state estimation at time t . Applying the trajectory that minimizes the observation criterion until the next MPC update, one obtains the smallest possible uncertainty estimation. At time $t=3s$, the estimator has very few knowledge on the target state. Performing maneuvers perpendicular to the LOS helps it to reduce its uncertainty around the target real state.

6. CONCLUSION AND FUTURE WORK

This paper has presented a tracking system made of two coupled algorithms. The first algorithm, based on a set-membership particle filter (BRPF), aims to estimate the

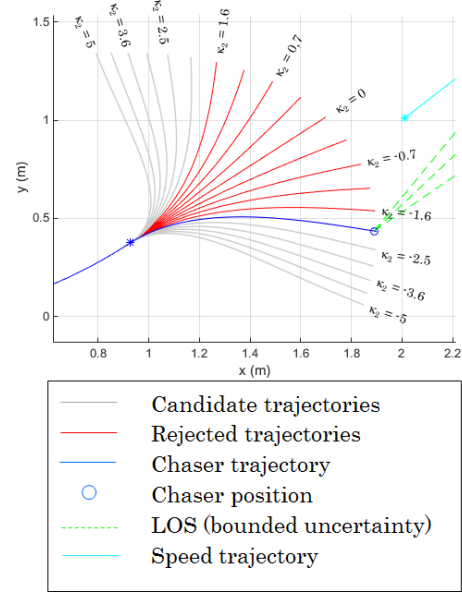


Fig. 15. MPC trajectories

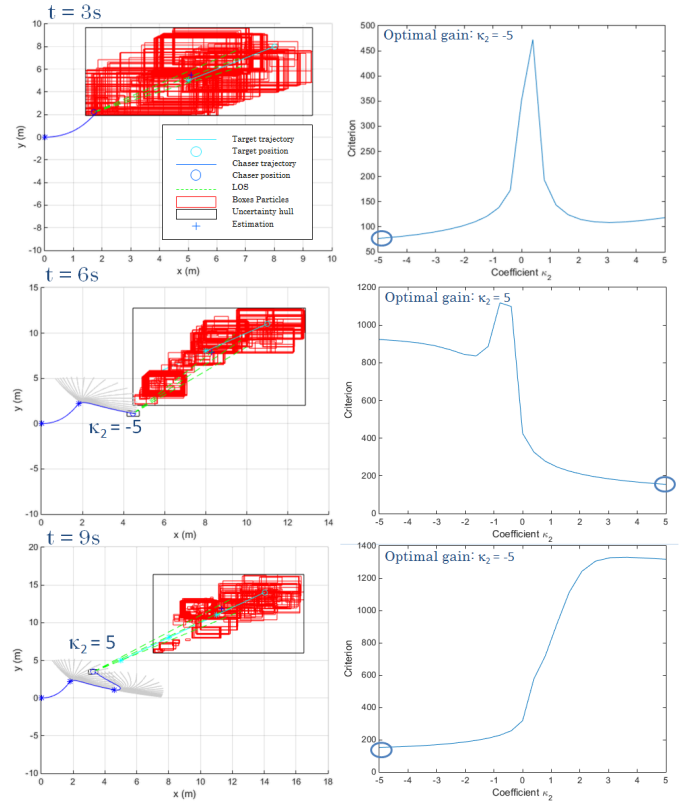


Fig. 16. MPC simulation

state of a distant ground target (geographical position and velocity). The second algorithm, based on Model Predictive Control (MPC), aims to optimize the control input to maximize the estimator's precision. The MPC algorithm is shown to be efficient in maximizing the target observability. Since the structure of the control law has been fixed *a priori*, the chaser behavior is guaranteed to remain in a suitable flight domain. Furthermore, the discretization of the possible control values guarantees a constant computation cost. The set-membership particle

filtering approach is shown to be more robust than a Bayesian EKF. Furthermore, it respects real-time computational constraints..

In the short term, the authors are working on an experiment to quantify more accurately the algorithm behavior and the associated computation costs in a real situation. Future work will tackle the issue of multi-target tracking and multi-chaser surveillance.

ACKNOWLEDGEMENTS

The authors would like to thank H. Piet-Lahanier, S. Reynaud, S. Bertrand, and K. Dahia for their advice.

REFERENCES

- Abdallah, F., Gning, A., and Bonnifait, P. (2007). Box particle filtering for non linear state estimation using interval analysis. *Automatica*.
- Aidala, V. (1979). Kalman filter behavior in bearings-only tracking applications. *IEEE transactions on aerospace and electronic systems*, 15, 29–39.
- DeFreitas, A., Mihaylova, L., Gning, A., Angelova, D., and Kadiramanathan, V. (2016). Autonomous crowds tracking with box particle filtering and convolution particle filtering. *Automatica, to appear*.
- Gning, A., Mihaylova, L., and Abdallah, F. (2012). Particle filtering combined with interval methods for tracking applications. *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*.
- Gning, A., Ristic, B., Mihaylova, L., and Abdallah, F. (2013). An introduction to box particle filtering. *IEEE Signal Processing Magazine*, 166–171.
- Jaulin, L. (2009). Robust set-membership state estimation. *Automatica*, 45, 202–206.
- Merlinge, N., Dahia, K., and Piet-Lahanier, H. (2016). A box regularized particle filter for terrain navigation with highly non-linear measurements. In *IFAC ACA 2016 Conference*. Sherbrooke, Canada.
- Moore, R., Bierbaum, F., and Schwartz, K. (1979). *Methods and Applications of Interval Analysis*. SIAM, Philadelphia.
- Moore, R., Kearfott, R.B., and Cloud, M. (2009). *Introduction to Interval Analysis*. SIAM, Philadelphia.
- Morari, M. and Lee, J. (1999). Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23, 667–682.
- Musso, C., Oudjane, N., and LeGland, F. (2001). Improving regularized particle filters. *Sequential Monte Carlo Methods in Practice*, 12, 247–271.
- Rassi, T., Ramdani, N., and Candau, Y. (2009). Set membership state and parameter estimation for systems described by nonlinear differential equations. *Automatica*, 40, 1771–1777.
- Ristic, B. (2004). *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, Boston.
- Rocheffort, Y., Piet-Lahanier, H., Bertrand, S., Beauvois, D., and Dumur, D. (2014). Model predictive control of cooperative vehicles using systematic search approach. *Control Engineering Practice*, 32, 204–217.
- Shneydor, N. (1998). *Missile Guidance and Pursuit*. Horwood Publishing, Chichester.