

## VISION-BASED ATTITUDE DETERMINATION USING A SLAM ALGORITHM DURING RELATIVE CIRCUMNAVIGATION OF NON-COOPERATIVE OBJECTS

Andrea Antonello\*, Panagiotis Tsiotras†

\*Centre of Studies and Activities for Space - G. Colombo, UNIVERSITY OF PADOVA, Italy

†School of Aerospace Engineering, GEORGIA INSTITUTE OF TECHNOLOGY, Atlanta GA - USA

**Abstract - We approach the problem of a chaser satellite circumnavigating a target object in a relative orbit. The objective is to obtain a map of the scenario and to measure the reciprocal position of the chaser-target pair, in order to subsequently perform proximity operations (active debris removal, rendezvous, servicing, etc.) more reliably. This work analyzes the case of a target-chaser scenario in a closed Clohessy-Wiltshire relative orbit. The chaser satellite has a vision sensor and observes a set of landmarks on the target satellite: the control acts on the yaw-rotation of the detector. By defining a probability distribution over a set of feasible control trajectories, we perform a search for a near-optimal solution. At the core of this approach lies the cross entropy minimization technique for estimating rare-event probabilities, which iteratively approximates the sampling distribution towards regions of progressively lower cost until converging to the optimum. We present simulations of a tracking scenario, demonstrating the validity of the proposed control technique. Performance of the proposed policy is compared with the case of a non controlled sensor by evaluating the time spent under observation and the residual uncertainty bounds on the landmarks. Results confirm the validity of the proposed approach.**

### I. INTRODUCTION

Vision-based navigation has been already demonstrated in several space missions that involved rendezvous and docking operations between satellites. While maneuvers between cooperative targets have been tested in orbit, limited experience is available for the case of non-cooperative objects, such as debris or malfunctioning spacecraft. The possibility of repairing and refurbishing out-of-order satellites with unmanned vehicles might give rise to a multi-million business opportunity:<sup>1</sup> nowadays, several space agencies and private companies are pushing in this direction, and are triggering the interest in researching close approach maneuvers and On-Orbit Service (OOS) operations.

In this paper we integrate planning and stochastic optimization with agent localization in order to control a vision sensor mounted on an autonomous spacecraft in orbit. Control here translates in the rotation of spacecraft or sensor about one of its axes so that it points to the location of the landmarks previously detected on the target satellite. The estimation accuracy of the detected features drives which feature should be selected next, and hence also drives the corresponding control action. The control and

estimation steps are therefore coupled. This active localization approach, when coupled with a mapping step of the initially unknown environment, is known in the terrestrial robotics community as Active SLAM (Simultaneous Localization and Mapping) and it has been studied extensively in the past.<sup>2,3</sup>

In order to capture estimation uncertainty in the optimization step, we construct a cost function which includes a term encapsulating the uncertainty of the state, provided by an Extended Kalman Filter (EKF). Since the resulting optimization problem is difficult to solve, we resort to the method of cross-entropy (CE) minimization to find the optimal control strategy. The CE strategy iteratively selects the best attitude trajectories to minimize the aforementioned cost. The method provides a framework to obtain near-optimal solutions for the orbital circumnavigation problem by jointly considering control, planning and estimation.

This paper is divided as follows: first, in Section II we introduce the Cross Entropy minimization mathematical preliminaries; in Sections III and IV we describe the problem under analysis and derive the state and measurement model; in Section V we proceed to the design of the algorithm and of the Extend Kalman Filter application; in Section VI we finally present the

simulation results.

## II. MATHEMATICAL PRELIMINARIES

### Cross Entropy Minimization

In this section we present the Cross Entropy minimization algorithm and show how it can be used to solve a certain class of stochastic optimal control problems. Assume the following stochastic dynamic system:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})\delta t + g(x)d\mathbf{w} \quad [1]$$

in which  $\mathbf{x} \in \mathbb{R}^n$  is the state of the system,  $\mathbf{u} \in \mathbb{R}^p$  is the control input, and  $\mathbf{w} \in \mathbb{R}^l$  is a zero-mean Gaussian process with covariance  $\Sigma_{\mathbf{w}}$ . We want to minimize a cost function as

$$\min \mathbb{E}_p[\mathcal{L}(\mathbf{x}, \mathbf{u})] \quad [2]$$

where the expectation in [2] is with respect to the trajectories of [1]. If  $\mathbf{u}(t)$  depends on a parameter vector  $\lambda \in \mathbb{R}^m$ , we can rewrite the control input as  $\mathbf{u}(t; \lambda)$ . The result of this parametrization is that we will minimize the cost function with respect to the finite dimensional parameters vector  $\lambda$ . According to the CE minimization method,<sup>4</sup> the cost function can be rewritten as follows:

$$J(\lambda) = \mathbb{E}_p[\mathcal{L}(\lambda)] = \int p(\lambda) \mathcal{L}(\lambda) d\lambda \quad [3]$$

where  $p(\lambda)$  is the probability density function corresponding to sampling trajectories based on [1]. By performing importance sampling from a proposal probability density  $q(\lambda)$ , the cost function is

$$\hat{J}(\lambda) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \left[ \frac{p(\lambda_i; \mu)}{q(\lambda_i)} \mathcal{L}(\lambda_i) \right] \quad [4]$$

in which  $N_s$  samples were drawn. The probability density that minimizes the variance of the estimator  $\hat{J}$  is:

$$q^*(\lambda) = \underset{q}{\operatorname{argmin}} \operatorname{Var} \left[ \frac{p(\lambda; \mu)}{q(\lambda)} \mathcal{L}(\lambda) \right] = \left[ \frac{p(\lambda; \mu) \mathcal{L}(\lambda)}{J(\lambda)} \right] \quad [5]$$

and it is the optimal, with respect to the variance, importance sampling density. As explained earlier, the goal of CE is to find the parameters  $\psi \in \Psi$  within the parametric class of pdfs  $p(\lambda; \psi)$ , such that the probability distribution  $p(\lambda; \psi)$  is close to the optimal distribution  $q^*(\lambda)$  given in [5]. Using the Kullback-Leibler divergence as the distance metric between

$q^*(\lambda)$  and  $p(\lambda; \psi)$ , the optimal parameters can be approximated numerically as:

$$\psi^* \approx \underset{\psi}{\operatorname{argmax}} \frac{1}{N_s} \sum \mathcal{L}(\lambda) \ln[p(\lambda; \psi)]. \quad [6]$$

Our goal is to compute  $\lambda$  that satisfies the following equation:

$$\mathbb{P}(\mathcal{L} \leq \epsilon) = \mathbb{E}_{p(\lambda; \mu)}[I_{\{\mathcal{L} \leq \epsilon\}}] \quad [7]$$

where  $\epsilon$  is a small constant and  $I$  is the indicator function. Using [4], this probability can be numerically approximated:

$$\hat{\mathbb{P}}(\mathcal{L} \leq \epsilon) \approx \frac{1}{N_s} \sum \left[ \frac{p(\lambda_i; \mu)}{p(\lambda_i; \psi)} I_{\{\mathcal{L}(\lambda_i) \leq \epsilon\}} \right]$$

where  $\lambda_i$  are i.i.d samples drawn from  $p(\lambda; \mu)$ . Based on [6], the goal is to find the optimal  $\psi^*$ , which is defined as:

$$\psi^* \approx \underset{\psi}{\operatorname{argmax}} \frac{1}{N_s} \sum I_{\{\mathcal{L}(\lambda_i) \leq \epsilon\}} \ln[p(\lambda_i; \psi)] \quad [8]$$

in which the samples  $\lambda_i$  are generated according to probability density  $p(\lambda; \mu)$ . In order to estimate the above probability, it would be computationally expensive to use a brute force method, such as Monte-Carlo<sup>5</sup> due to the rarity of the event  $\{\mathcal{L} \leq \epsilon\}$ . A solution is to start with  $\epsilon_1 > \epsilon$  for which the probability of the event  $\{\mathcal{L} \leq \epsilon_1\}$  is equal to some  $\rho > 0$ . Then, the value  $\epsilon_1$  is set to the  $\rho$ -th quantile of  $\mathcal{L}(\lambda)$  which means that  $\epsilon_1$  is the largest number for which:

$$\mathbb{P}(\mathcal{L}(\lambda) \leq \epsilon_1) = \rho.$$

The parameter  $\epsilon_1$  can be found by sorting the samples according to their cost in increasing order and setting  $\epsilon_1 = \mathcal{L}_{\rho N}$ . The optimal parameter  $\psi_1$  for the level  $\epsilon_1$  is calculated according to [8] using the value  $\epsilon_1$ . This iterative procedure terminates when  $\epsilon_k \leq \epsilon$ , in which case the corresponding parameter  $\psi_k$  is the optimal one and thus  $\psi^* = \psi_k$ .

In order to find the optimal trajectory  $\lambda^*$  and the corresponding optimal parameters  $\psi_k$ , the process is iterated until  $\epsilon \rightarrow \epsilon^*$ , where  $\epsilon^* = \min \mathcal{L}(\epsilon)$ . Since  $\epsilon^*$  is not known a-priori, we choose as  $\epsilon^*$  the value of  $\epsilon$  for which no further improvement within a predefined tolerance in the iterative process is observed. The overall algorithm is summarized in the table below.

## III. PROBLEM FORMULATION

### III.i Relative Navigation in Orbit

In reference,<sup>6</sup> the CE approach was applied to the problem of a two-dimensional circular orbit. In this

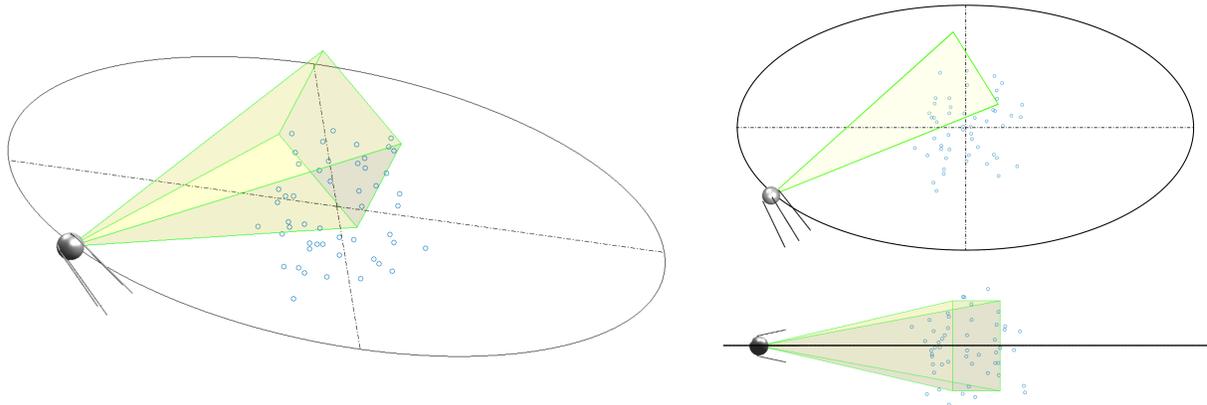


Fig. 1: Problem set up and simulated sensing scenario.

---

### Cross Entropy Algorithm

---

- 1: Draw  $N$  samples for  $\lambda$  from a probability distribution  $p(\lambda, \xi)$ , e.g. a Gaussian distribution  $\mathcal{N}(\mu_\xi, \Sigma_\xi)$ .
- 2: Compute the cost  $\mathcal{L}(\lambda)$  for each  $\lambda$  and sort them in ascending order.
- 3: Select the best performing  $\rho$ -th percentile and find the optimal parameters  $(\mu_\xi^*, \Sigma_\xi^*)$  which maximize

$$(\mu_\xi^*, \Sigma_\xi^*) = \operatorname{argmax}_{\mu_\xi, \Sigma_\xi} \frac{1}{|\mathcal{E}|} \sum_{e=1}^{|\mathcal{E}|} \ln p(\lambda, \xi)$$

- 4: Set  $(\mu_\xi, \Sigma_\xi) = (\mu_\xi^*, \Sigma_\xi^*)$
  - 5: Repeat from (1) until the variation of  $(\mu_\xi, \Sigma_\xi)$  is smaller than a predefined threshold.
- 

paper, we wish to extend that work to the general case of a target and a chaser satellite in a 3D orbital scenario. The relative orbit will be described by Clohessy Wiltshire's equations (CW).

The objective of the chaser is to obtain a map of a set of landmarks located on the target satellite. These can be features such as edges, patches, arrays of LEDs, etc. For the sake of simulations, we consider the landmarks as single 3D points randomly located in a bounded box inside the relative orbit. The process of gathering information about the landmarks positions is achieved through the application of a Simultaneous Localization and Mapping (SLAM) algorithm, which also allows for a simultaneous improvement of the chaser localization.

The satellite has an onboard sensor, which is free to rotate around the axis normal to the orbital plane.

Frame  $\{G\}$  denotes the Global frame\*,  $\{R\}$  the Local non Rotating Frame attached to the chaser and  $\{S\}$  the Local Rotating Frame attached to the satellite sensor. In addition, we define the angles  $\phi$  and  $\theta$ , which respectively represent the heading direction of the satellite and the sensor bearing. Note that in our setup, frame  $\{S\}$  has a positive  $\pi/2$  angular offset with respect to  $\{R\}$ .

In order to describe the relative motion between the chaser and the target, we start by analyzing Hill's equations.

### III.ii Clohessy-Wiltshire Reference Frame

The Clohessy-Wiltshire framework allows for the description of orbital relative motion, in which the target is in a circular orbit, and the chaser is in an elliptical (or circular) orbit. This model is a first-order approximation of the actual chaser's motion in a target-centered coordinate system.

Hill's differential equations in Cartesian coordinates and in the non homogeneous form can be written as follows:<sup>7,8</sup>

$$\begin{cases} \ddot{x} - 3n^2x - 2n\dot{y} & = f_x \\ \ddot{y} + 2n\dot{x} & = f_y \\ \ddot{z} + n^2z & = f_z \end{cases}$$

The CW equations can be obtained solving Hill's differential unforced equations with the standard

---

\*For example,  $\{G\}$  could represent the base frame of a Clohessy-Wiltshire transformation for a relative navigation problem.

Laplace transform, which yields:

$$\begin{cases} x(t) = x_0[4 - 3 \cos(nt)] + \frac{\sin(nt)}{n} \dot{x}_0 + \frac{2\dot{y}_0}{n} [1 - \cos(nt)] \\ y(t) = y_0 - \frac{2\dot{x}_0}{n} - 3(2nx_0 + \dot{y}_0)t + 2(3x_0 + \frac{2\dot{y}_0}{n}) \sin(nt) \\ \quad + \frac{2\dot{x}_0}{n} \cos(nt) \\ z(t) = z_0 \cos(nt) + \frac{\dot{z}_0}{n} \sin(nt) \end{cases}$$

One interesting property of these equations is that, although the equations describing the in-plane motion are coupled, the out-of-plane motion is decoupled.

Even though the chaser does not actually orbit around the target satellite, the instantaneous motion is somewhat elliptical.<sup>9</sup> The term  $(2nx_0 + \dot{y}_0)t$  in the  $y$ -equation represents the secular drift between the chaser and the target due to differences in the orbital periods. If this term is set to zero by choosing the appropriate initial conditions,  $\dot{y}_0 + 2nx_0 = 0$ , then the linearized relative orbit will have a bounded motion.

Assuming this constraint is satisfied, then the HCW equations can be rewritten as follows:<sup>10</sup>

$$\begin{cases} x(t) = \alpha \sin(nt + \phi_1) \\ y(t) = 2\alpha \cos(nt + \phi_1) + \Delta y \\ z(t) = \beta \sin(nt + \phi_2) \end{cases}$$

where the parameters  $\alpha$ ,  $\beta$ ,  $\Delta y$ ,  $\phi_1$  and  $\phi_2$  are determined through the relative orbit initial conditions:

$$\alpha = \sqrt{x_0^2 + \frac{\dot{x}_0^2}{n^2}}, \quad \beta = \sqrt{z_0^2 + \frac{\dot{z}_0^2}{n^2}} \quad [9]$$

$$\Delta y = y_0 - 2\frac{\dot{x}_0}{n} \quad [10]$$

$$\phi_1 = \tan^{-1}\left(\frac{nx_0}{\dot{x}_0}\right), \quad \phi_2 = \tan^{-1}\left(\frac{nz_0}{\dot{z}_0}\right) \quad [11]$$

In order to simplify the analysis, we may impose a target-centered orbit by driving  $\Delta y$  to zero, that is let  $y_0 = 2(\dot{x}_0/n)$ . The case  $\Delta y \neq 0$  is trivial and does not add any significant novelty. One finally obtains:

$$\begin{cases} x(t) = \alpha \sin(nt + \phi_1) \\ y(t) = 2\alpha \cos(nt + \phi_1) \\ z(t) = \beta \sin(nt + \phi_2) \end{cases}$$

$$\begin{cases} \dot{x}(t) = \alpha \cos(nt + \phi_1)n \\ \dot{y}(t) = -2\alpha \sin(nt + \phi_1)n \\ \dot{z}(t) = \beta \cos(nt + \phi_2)n \end{cases}$$

The unit vector normal to the osculating plane can be derived as the unity momentum:

$$\hat{\mathbf{n}} = \frac{\mathbf{x} \times \dot{\mathbf{x}}}{\|\mathbf{x} \times \dot{\mathbf{x}}\|} \quad [12]$$

A new frame of reference attached to the Hill's orbit is defined,  $\{\mathbf{H}\}$ , with the  $x$ -axis and  $y$ -axis lying on the osculating plane and directed towards the apsis and periapsis respectively. The direction is chosen to form a right-handed frame  $\{\hat{\mathbf{i}}_H, \hat{\mathbf{j}}_H, \hat{\mathbf{k}}_H\}$  with the  $z$ -axis, represented by  $\hat{\mathbf{n}}$ . Frames  $\{\mathbf{G}\}$  and  $\{\mathbf{H}\}$  have the same null origin by definition since  $y_0 = 2(\dot{x}_0/n)$ .

### III.iii State Model

The state model of the orbiting satellite, augmented with the position of the landmarks, can then be written in differential form as:

$$\begin{bmatrix} dx(t) \\ dy(t) \\ dz(t) \\ d\phi(t) \\ d\theta(t) \\ d\mathbf{p}_1(t) \\ \vdots \\ d\mathbf{p}_N(t) \end{bmatrix} = \begin{bmatrix} \alpha \cos(\phi + \phi_1)n \\ -2\alpha \sin(\phi + \phi_1)n \\ \beta \cos(\phi + \phi_2)n \\ \omega_\phi(t)dt \\ \omega_\theta(t)dt \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_5 \\ \mathbf{0}_{3N \times 5} \end{bmatrix} d\mathbf{w}(t) \quad [13]$$

where  $x, y, z$  indicate the position of the chaser satellite in the  $\{\mathbf{G}\}$  frame, and the angles  $\phi$  and  $\theta$  are the rotation of the chaser and the sensor expressed in frames  $\{\mathbf{H}\}$  and  $\{\mathbf{R}\}$  respectively; note that  $\phi(t) = nt$  is the cumulative angle: that is, we suppose that in the baseline case (with no control applied), the chaser rotates with an angular velocity vector perpendicular to the relative orbital plane and with magnitude equal to the mean motion ( $\omega_{\phi_k} = n$ ).

The landmark positions  $\mathbf{p}_1, \dots, \mathbf{p}_N$  are expressed in the global frame, and yield an augmented state  $\mathbf{x} \in \mathbb{R}^{3N+5}$ .

In the model,  $d\mathbf{w} \in \mathbb{R}^5$  represents a Wiener process with covariance matrix  $\Sigma_w = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2)$ .

In discrete form, the model becomes:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \phi_{k+1} \\ \theta_{k+1} \\ \mathbf{p}^1_{k+1} \\ \vdots \\ \mathbf{p}^N_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \alpha \cos(\phi_k + \phi_1)n\delta t \\ y_k - 2\alpha \sin(\phi_k + \phi_1)n\delta t \\ z_k + \beta \cos(\phi_k + \phi_2)n\delta t \\ \phi_k + \omega_{\phi_k}\delta t \\ \theta_k + \omega_{\theta_k}\delta t \\ \mathbf{p}^1_k \\ \vdots \\ \mathbf{p}^N_k \end{bmatrix} + \begin{bmatrix} \mathbf{I}_5 \\ \mathbf{0}_{3N \times 5} \end{bmatrix} \mathbf{w}(t_k) \quad [14]$$

### III.iv Measurement Model

Detection of the landmarks occurs only if they are within the field of view and range of the sensor, depicted as the yellow truncated pyramid in Fig. 1. When a feature is detected, the sensor outputs the

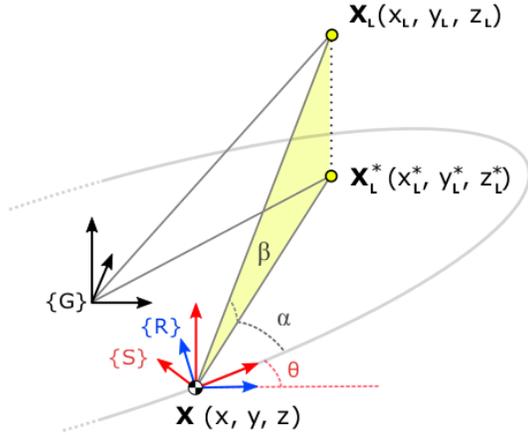


Fig. 2: Measurement model definition.

vector  $\mathbf{z} = (r, \alpha, \beta)$ , where  $r$  is the range and the tuple  $(\alpha, \beta)$  are the azimuth and elevation angles (see Fig. 2).

The measurement model, expressed in continuous form, is the following:

$$\mathbf{z}(t) = {}^S_R \mathbf{R}(\theta(t)) {}^R_H \mathbf{R}(\phi(t)) {}^H_G \mathbf{R}(\mathbf{p}_i(t) - \mathbf{p}_R(t)) + \mathbf{v}(t) \quad [15]$$

where  $\mathbf{p}_i = (p_{x_i}, p_{y_i}, p_{z_i})$  and  $\mathbf{p}_R = (x, y, z)$  are the position of the landmarks and the observer satellite, respectively, expressed in the base frame. The term  $\mathbf{v}(t)$  corresponds to the observation noise of the sensor which is considered zero-mean Gaussian with covariance matrix  $\Sigma_v = \text{diag}(\sigma_I^2, \sigma_{II}^2, \sigma_{III}^2)$ . The matrices  ${}^H_G \mathbf{R}$ ,  ${}^S_R \mathbf{R}(\theta(t))$  and  ${}^R_H \mathbf{R}(\phi(t))$  express rotational transformations from the base frame  $\{G\}$  to the orbit  $\{H\}$ , from  $\{H\}$  to the observer frame  $\{R\}$  and from  $\{R\}$  to the sensor frame  $\{S\}$ , respectively. In compact form, the observation model is then written as:

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}(t) \quad [16]$$

In a real scenario, however, measurements will be taken discretely, according to the sampling strategy adopted. The measurement model, in discrete time form, can then be written as:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad [17]$$

Referring to Fig. 2, the measurement model mapping

function can be described as:

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \sqrt{(x_L - x)^2 + (y_L - y)^2 + (z_L - z)^2} \\ \tan^{-1} \left( \frac{y_L^* - y_G}{x_L^* - x_G} \right) - \phi - \theta \\ \tan^{-1} \left( \frac{\text{sgn}((\mathbf{x}_L - \mathbf{x}_L^*) \cdot \mathbf{n}) \|\mathbf{x}_L - \mathbf{x}_L^*\|}{\|\mathbf{x}_L^* - \mathbf{x}\|} \right) \end{bmatrix} \quad [18]$$

The projected vector  $\mathbf{x}_L^*$  can be further expressed as a function of the state by knowing the transformation map between frames  $\{G\}$  and  $\{H\}$ , which remains constant throughout the simulation:

$$\mathbf{x}_L^* = \mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n}) \mathbf{n} \quad [19]$$

The measurement model can then be written as:

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \sqrt{(x_L - x)^2 + (y_L - y)^2 + (z_L - z)^2} \\ \tan^{-1} \left( \frac{(\mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n}) \mathbf{n} - \mathbf{x}) \cdot \mathbf{j}_H}{(\mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n}) \mathbf{n} - \mathbf{x}) \cdot \mathbf{i}_H} \right) - \phi - \theta \\ \tan^{-1} \left( \frac{\text{sgn}(((\mathbf{x}_L \cdot \mathbf{n}) \mathbf{n}) \cdot \mathbf{n}) \|(\mathbf{x}_L \cdot \mathbf{n}) \mathbf{n}\|}{\|\mathbf{x}_L - (\mathbf{x}_L \cdot \mathbf{n}) \mathbf{n} - \mathbf{x}\|} \right) \end{bmatrix} \quad [20]$$

#### IV. MAIN PROBLEM

The problem is to estimate the position of the landmarks by properly evaluating the measurements taken by the sensor. To perform this task, we control the rotation of the sensor along the orbital plane's normal to maximize the performances in a finite time horizon. That is, we want to minimize a cost function containing both the final uncertainty of the estimate and the actuation cost. Such a function is of the form:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \|e^2(t_N)\| + \int_0^{t_N} \left( Q(\mathbf{x}) + \frac{1}{2} \mathbf{u}^T R \mathbf{u} \right) dt \quad [21]$$

where  $\|e^2(t_N)\|$  is the terminal cost at a certain time horizon setpoint  $T = t_N$ . Unfortunately, this error is not known and a strategy for its approximation needs to be introduced.

To do this, we approximate the error with a measure of the estimation uncertainty. We introduce two main cost strategies based on a) the covariance matrix trace and on b) the time under observation of the landmarks. Both strategies evaluate the terminal performances of the piecewise control trajectory and the actuation cost needed to achieve it.

CE minimization is utilized to iteratively select attitude trajectories minimizing the afore-mentioned cost. The result is a near-optimal path, in terms of

achieving a predefined goal in the state space, while reducing the localization error and the total uncertainty. With this method, we are able to obtain near-optimal solutions to the orbital navigation problem by jointly considering control, planning and estimation.

After the first reconnaissance turn has been completed, and the state vector  $\mathbf{x}$  has been augmented to dimension  $\mathbb{R}^{5+3 \times \mathcal{N}}$  through landmark observations<sup>†</sup>, the CE routine is implemented.

Since the trajectory simulated in the EFK prediction routine is dependent on the intrinsic uncertainty of the sensor, a long time horizon will induce a build up of errors and uncertainties. The CE routine is then applied to a finite time horizon, equal to a fraction of the orbital period.

In this simulation, the orbit has been divided in  $s$  sectors: each sector will then be further divided in  $m$  sampling boxes, where  $m$  is the size of the control action vector  $\lambda$ . Depending on the implemented discretization, each box will consist of  $k$  iterations. That is, for each box  $m_i$ , constant control parameters  $\lambda_i$  are applied for  $k$  number of iterations inside the box.

Once the aforementioned parameters have been selected,  $N_{\text{traj}}$  random control laws are drawn by using the starting distribution parameters  $\mathbf{v}_0$ . An EKF simulation is then run for each of the  $N_{\text{traj}}$  control laws, leading to different trajectories; these are ordered according to their respective cost and a quantile  $q$ -th is selected. The best  $q$ -th quantile provides the new parameters  $\mathbf{v}_i$  from which the next  $N_{\text{traj}}$  control laws are drawn. The process iterates for the  $N_{CE}$  cross-entropy optimization steps. The output of the algorithm is the near-optimal control law  $\lambda \in \mathbb{R}^m$  with  $m$  being the number of boxes in which the  $s_i$  sector is divided.

We introduce two cost functions that are feasible in for the CE minimization strategy.

#### IV.i Trace of the covariance matrix (TCM)

Using a cost function that includes a measure of estimation uncertainty can lead to trajectories that reduce the overall uncertainty of a map:<sup>11</sup> hence, a first cost strategy implements the trace of the covariance matrix as a measure of uncertainty for the state estimate provided by an Extended Kalman Filter (EKF). Such a function will be of the form:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \psi(\mathbf{x}_{t_N}) + \sum_{k=0}^N \left( Q(\mathbf{x}_{t_N}) + \frac{1}{2} \mathbf{u}(\mathbf{t}_k)^T R \mathbf{u}(\mathbf{t}_k) \right) \quad [22]$$

<sup>†</sup> $N \leq \mathcal{N}$

with the terminal cost being:

$$\psi_1(\mathbf{x}_{t_N}) = \text{trace}(\Sigma(t_N)) \quad [23]$$

Letting for simplicity  $Q(x) = 0$ , we have:

$$\mathcal{L}_{\text{tcm}} = \text{trace}(\Sigma(t_N)) + \sum_{k=0}^N \left( \frac{1}{2} \mathbf{u}(\mathbf{t}_k)^T R \mathbf{u}(\mathbf{t}_k) \right) \quad [24]$$

#### IV.ii Time under observation (TUO)

In this second strategy, the cost is defined as the time under observation of the landmarks by the sensor. For each trajectory obtained from the sampling, the number of landmarks seen by the sensor at each iteration is stored ( $T_i$ ).

For each trajectory the total number of observed landmarks is then summed:

$$\psi_2(\mathbf{x}_{t_N}) = \sum_i T_i \quad [25]$$

The complete function, taking into account the actuation cost, is then:

$$\mathcal{L}_{\text{tuo}} = \sum_i T_i + \sum_{k=0}^N \left( \frac{1}{2} \mathbf{u}(\mathbf{t}_k)^T R \mathbf{u}(\mathbf{t}_k) \right) \quad [26]$$

## V. DESIGN OF THE ALGORITHM

In order to treat the minimization problem with the cross entropy approach, we divide the problem in two parts. First, the chaser satellite performs a so called *recognition* orbit, which is needed to obtain a rough estimate of the landmarks' location. In this part, no control input is applied,  $\omega_\theta = 0$ .

In order to proceed, we consider the discretized version of the dynamics in Eq. [14]. The dimension of the state is initially  $\mathbf{x} \in \mathbb{R}^5$  and changes according to the number of landmarks detected. The proposed algorithm consists on the two phases:

- the *recognition* phase, during which the measurements taken by the chaser provide a first estimation of the visible landmarks
- the *incremental* estimation phase, during which the chaser keeps taking measurements in order to improve the overall state estimation.

The recognition phase is necessary since the chaser does not know the number and the position of the landmarks and, in turn, the dimension of the overall system state. During the recognition phase the chaser runs an EKF algorithm whose state is augmented whenever a measurement related to a new landmark is collected.<sup>12</sup>

### V.i Recognition Phase

Let  $\tilde{N}$  be the number of landmarks recognized up to the time instant  $k$  so the current state of the EKF

$$\mathbf{x}_k^{(\tilde{N})} = \left[ x_k \ y_k \ z_k \ \phi_k \ \theta_k \ \mathbf{p}_k^{(1)} \ \mathbf{p}_k^{(2)} \ \dots \ \mathbf{p}_k^{(\tilde{N})} \right]^T.$$

We divide the design of the EKF into prediction and update steps.

#### Prediction step

The update equation is

$$\begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \\ \hat{z}_{k+1|k} \\ \hat{\phi}_{k+1|k} \\ \hat{\theta}_{k+1|k} \\ \hat{p}_{k+1|k}^{(1)} \\ \hat{p}_{k+1|k}^{(2)} \\ \vdots \\ \hat{p}_{k+1|k}^{(\tilde{N})} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k} \\ \hat{y}_{k|k} \\ \hat{z}_{k|k} \\ \hat{\phi}_{k|k} \\ \hat{\theta}_{k|k} \\ \hat{p}_{k|k}^{(1)} \\ \hat{p}_{k|k}^{(2)} \\ \vdots \\ \hat{p}_{k|k}^{(\tilde{N})} \end{bmatrix} + \begin{bmatrix} \alpha \cos(\phi_{k|k} + \phi_1)n\delta t \\ -2\alpha \sin(\phi_{k|k} + \phi_1)n\delta t \\ \beta \cos(\phi_{k|k} + \phi_2)n\delta t \\ \omega_{\phi_k} \delta t \\ \omega_{\theta_k} \delta t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

or in a more compact form

$$\hat{\mathbf{x}}_{k+1|k}^{(\tilde{N})} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}^{(\tilde{N})}, \omega_{\phi_k}, \omega_{\theta_k}).$$

The update of the covariance matrix is given by

$$P_{k+1|k}^{(\tilde{N})} = F_k P_{k|k}^{(\tilde{N})} F_k^T + Q_k, \quad [27]$$

where

$$F_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}. \quad [28]$$

The matrix  $F_k$  has the following structure

$$F = \begin{bmatrix} F_k^{\text{mot}} & 0 \\ 0 & \mathbb{I}_{\tilde{N}} \end{bmatrix}, \quad [29]$$

where  $F_k^{\text{mot}}$  is given by the following

$$\begin{aligned} \mathbf{F}_k^{\text{mot}} &= \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial z} & \frac{\partial f_5}{\partial \phi} & \frac{\partial f_5}{\partial \theta} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & -\alpha \sin(\phi_{k|k} + \phi_1)n\delta t & 0 \\ 0 & 1 & 0 & -2\alpha \cos(\phi_{k|k} + \phi_1)n\delta t & 0 \\ 0 & 0 & 1 & -\beta \sin(\phi_{k|k} + \phi_2)n\delta t & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad [30]$$

where  $0_N$  is a null matrix of dimension  $N$ .

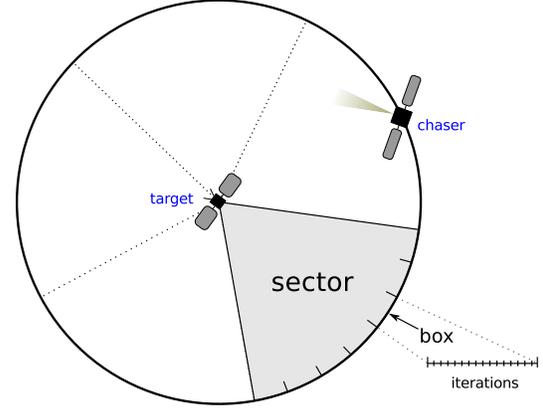


Fig. 3: Discretization strategy for cross-entropy algorithm

#### Update step

We assume to have the information provided by the range and bearing sensor  $\mathbf{z} = [r, \alpha, \beta]$ . Furthermore, we assume that we collect multiple measurements at the same time instant  $k+1$ , e.g.  $\bar{\mathbf{z}}_{k+1}$ . This vector can be divided in two components, the first component  $\mathbf{z}_{k+1}^{(1)}$  which is given by all the measurements associated to already seen landmarks and the second component  $\mathbf{z}_{k+1}^{(2)}$  which represents measurements associated to new landmarks. The measurement model can be written as

$$\bar{\mathbf{z}}_{k+1} = \begin{bmatrix} \mathbf{z}_{k+1}^{(1)} \\ \mathbf{z}_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{h}^{(1)}(\hat{\mathbf{x}}_{k+1}) + v_{k+1}^{(1)} \\ \mathbf{h}^{(2)}(\hat{\mathbf{x}}_{k+1}) + v_{k+1}^{(2)} \end{bmatrix}.$$

We proceed with the computation of the Jacobian of the observation model with respect to the robot pose and the observed landmark coordinates. At iteration  $k+1$  we get

$$\mathbf{H}_{k+1} = \left. \frac{\partial \mathbf{h}_{k+1}^{(1)}}{\partial \mathbf{x}^{(\tilde{N})}} \right|_{\hat{\mathbf{x}}_{k+1|k}} \quad [31]$$

By having the output matrix  $\mathbf{H}_{k+1}$  we can update the state related to all chaser attitude and all the already seen landmarks

$$\begin{aligned} K_{k+1} &= P_{k+1|k}^{(\tilde{N})} H_{k+1}^T \left( H_{k+1} P_{k+1|k}^{(\tilde{N})} H_{k+1}^T + R_{k+1} \right)^{-1} \\ x_{k+1|k+1}^{(\tilde{N})} &= x_{k+1|k}^{(\tilde{N})} + K_{k+1} \mathbf{z}_{k+1}^{(1)} \\ P_{k+1|k+1}^{(\tilde{N})} &= (\mathbb{I} - K_{k+1} H_{k+1}) P_{k+1|k}^{(\tilde{N})} \end{aligned}$$

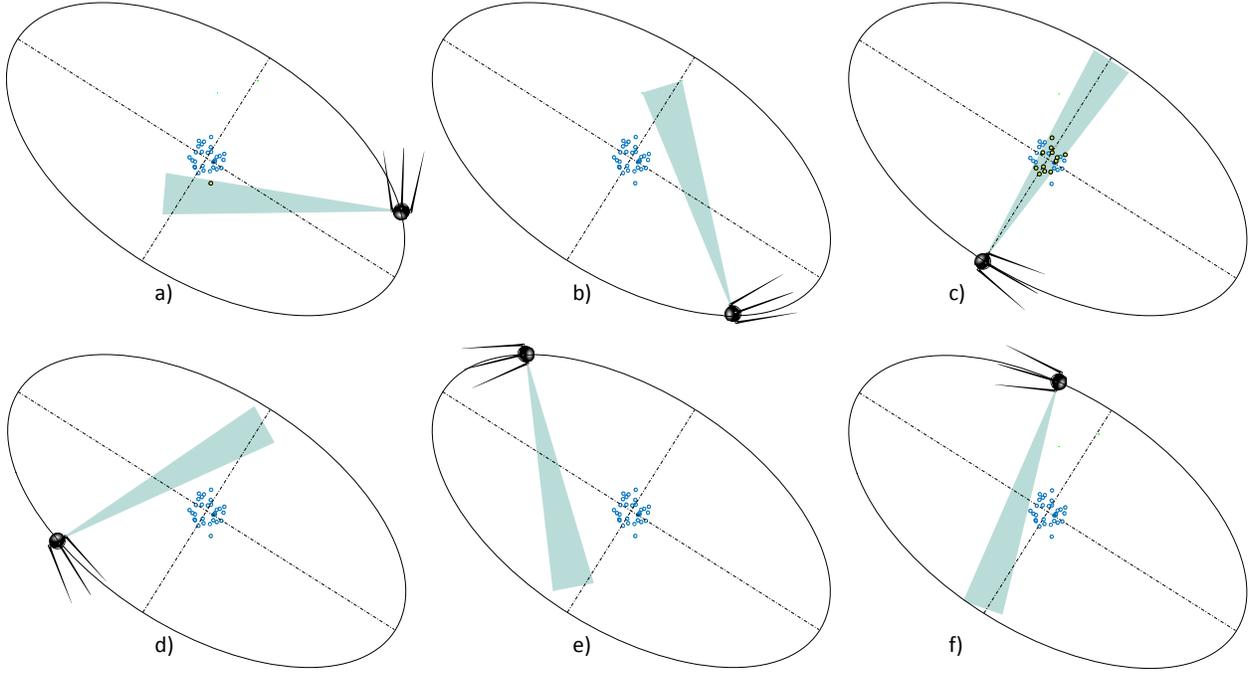


Fig. 4: Reconnaissance orbit: the sensor is fixed with respect to the satellite ( $\omega_\theta = 0$ ).

Without loss of generality, suppose that  $\mathbf{z}_{k+1}^{(2)}$  refers to just one new landmark  $p^{(\tilde{N}+1)}$ , then we have that

$$\hat{\mathbf{p}}_{k+1|k+1}^{(\tilde{N}+1)} = \begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \\ \hat{z}_{k+1|k} \end{bmatrix} + {}^R_H \mathbf{R} \begin{bmatrix} r \cos(\beta) \cos(\alpha + \hat{\phi}_{k+1|k} + \hat{\theta}_{k+1|k}) \\ r \cos(\beta) \sin(\alpha + \hat{\phi}_{k+1|k} + \hat{\theta}_{k+1|k}) \\ r \sin(\beta) \end{bmatrix}.$$

Then we can extend the state

$$\mathbf{x}_{k+1|k+1}^{(\tilde{N}+1)} = \begin{bmatrix} \mathbf{x}_{k+1|k+1}^{(\tilde{N})} \\ \hat{\mathbf{p}}_{k+1|k+1}^{(\tilde{N}+1)} \end{bmatrix},$$

and the covariance matrix

$$\mathbf{P}_{k+1|k+1}^{(\tilde{N}+1)} = \begin{bmatrix} \mathbf{P}_{k+1|k+1}^{(\tilde{N})} & \mathbf{P}(\tilde{N}, \tilde{N}+1) \\ \mathbf{P}(\tilde{N}+1, \tilde{N}) & \mathbf{P}(\tilde{N}+1) \end{bmatrix},$$

where

$$\mathbf{P}(\tilde{N}+1, \tilde{N}) = \left( \mathbf{P}(\tilde{N}, \tilde{N}+1) \right)^T = \begin{bmatrix} \frac{\partial \hat{\mathbf{p}}_{k+1|k+1}^{(\tilde{N}+1)}}{\partial \mathbf{x}_k} \\ \frac{\partial \hat{\mathbf{p}}_{k+1|k+1}^{(\tilde{N}+1)}}{\partial \mathbf{y}_k} \\ \frac{\partial \hat{\mathbf{p}}_{k+1|k+1}^{(\tilde{N}+1)}}{\partial \phi_k} \\ \frac{\partial \hat{\mathbf{p}}_{k+1|k+1}^{(\tilde{N}+1)}}{\partial \theta_k} \end{bmatrix} \bigg|_{(\hat{\mathbf{x}}_{k+1|k}, \bar{\mathbf{z}}_{k+1})}$$

$$\mathbf{P}(\tilde{N}+1) = \frac{\partial \hat{\mathbf{p}}_{k+1|k+1}^{(\tilde{N}+1)}}{\partial \mathbf{z}_{k+1}} \bigg|_{(\hat{\mathbf{x}}_{k+1|k}, \bar{\mathbf{z}}_{k+1})}.$$

#### V.ii Incremental Estimation Phase

After the recognition phase an initial guess of the landmark' position is stored in the state of the system. At this point the core of the algorithm runs to improve the estimate of the state, and this information is exploited to control the vision sensor.

Specifically, for any orbit all the following steps are repeated:

1. We draw  $N_{\text{traj}}$  random possible acceleration trajectories for the sensor,  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{N_{\text{traj}}}\}$ , from a Gaussian distribution with parameter  $v_i$  (the particular controller used in this paper will be explained in Section V.iii).
2. For all  $\lambda$  we simulate the behavior of the camera running an Extended Kalman filter as explained in Section V.i.
3. Once the state has been estimated at any time instant we can evaluate one of the cost function presented in Section IV and perform the CE algorithm. Basically we have to select the  $\rho - th$

best performing percentile, i.e. the trajectories with an associated lower cost.

4. From these reduced subset of samples the new parameters for the distribution are inferred. The aforementioned procedure is repeated up to the convergence of the cross entropy method and then the optimal solution is applied.

#### V.iii Cross Entropy optimization in the Case of Orbital Self-Localization

The controller acts on the angular velocity of the sensor,  $\omega_\theta$ . Recalling Eq. [21], we can rewrite the discrete cost as:

$$\hat{\mathcal{L}}(\mathbf{x}, \mathbf{u}) \approx \psi(\mathbf{x}_{t_N}) + \sum_{k=0}^N \left( \frac{1}{2} \mathbf{u}(\mathbf{t}_k)^T R \mathbf{u}(\mathbf{t}_k) \right), \quad [32]$$

where in Eq. [22] we let  $Q(\mathbf{x}) = 0$  and  $\psi(\mathbf{x}_{t_N}) = \|e^2(t_N)\|$ . The control law is parametrized as follows:

$$\omega_\theta(t_k) = \mathbf{u}(\omega_\theta(t_{k-1}), \eta(t_{k-1}; \lambda)) \quad [33]$$

$$\omega_\theta(t_k) = \omega_\theta(t_{k-1}) + \eta(t_{k-1}; \lambda) \delta t, \quad [34]$$

where  $\eta(t_{k-1}; \lambda)$  is the rotational acceleration, which is parameterized as a piecewise trajectory composed by  $m$  constant pieces. The choice of parameterizing the acceleration allows to have smooth (at least of class  $C^1$ ) angular trajectories.

Each constant acceleration  $\eta_m$  is being applied for a constant  $\delta t_i$ , where  $t_{\text{sect}} = \sum_{i=1}^m \delta t_i$ . The sum of all time intervals is fixed and is equal to the time horizon corresponding to the duration of each sector  $s$  (refer to the table in Section VI). The parameter vector  $\lambda$  is defined as  $\lambda^T = (t_1, \eta_1, \dots, t_m, \eta_m) \in \mathbb{R}^{2m}$ .

Each parameter vector  $\lambda$  corresponds to a unique control vector  $\mathbf{u}$ , which generates a trajectory  $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t_N}]$

In the simulation, and without loss of generality, we maintain the controller timestep constant  $\delta t_i = \delta t_m = t_{\text{sect}}/m$ . The accelerations  $\eta_i$  are initially obtained from a uniform distribution  $\mathcal{U}([\eta_{\min}, \eta_{\max}])$ , where the bounds are dictated by the specifics of the sensor.

#### V.iv Algorithm Set Up

Control in this scenario means that the active rotation of the spacecraft about one of its axes is such that the sensor points to the landmarks detected in a previous time step. A certain cost function (e.g. the estimation accuracy of the detected features or the cumulative number of features seen) drives which feature(s) to be observed next, and hence also drives

---

#### Problem Algorithm

---

- 1: **for**  $s = 1$  **to** total sectors per turn **do**
  - 2:   Select initial distribution parameters  $\mathbf{v}_0$
  - 3:   **for**  $i = 1$  **to** total CE optimization steps **do**
  - 4:     Draw  $N_{\text{traj}}$  random acceleration vectors  $\lambda \in \mathbb{R}^m$  from distribution with parameters  $v_i$
  - 5:     **for**  $j = 1$  **to**  $N_{\text{traj}}$  **do**
  - 6:       Run a simulated EKF with the input  $N_{\text{traj}}^j$
  - 7:       Evaluate the cost function [22] and store the value
  - 8:     **end for**
  - 9:     Sort all the cost function values in ascending order
  - 10:     Extract the  $\rho$ -th quantile
  - 11:     Run the cross entropy optimization [8] and extract the new distribution parameters  $\mathbf{v}_{i+1}$
  - 12:   **end for**
  - 13:   Apply the obtained near optimal control law  $\lambda \in \mathbb{R}^m$  to sector  $s$ .
  - 14: **end for**
- 

the control action. The control and estimation steps are therefore coupled. On the contrary, existing work in proximity operations solve the problem of control and estimation independently.<sup>13,14</sup>

## VI. SIMULATION RESULTS

The proposed algorithm has been used to simulate the acquisition and tracking of a set of landmarks on a virtual object located in the centroid of the closed Hill orbit.

In both cases, in order not to lose generality, landmarks are randomly placed according to a uniform distribution, so that the presence of particular geometrical properties/simmetries will not affect performance. The landmarks are thus generated according to the following distribution:

$$\mathbf{p} = \mathcal{U}(0, [\alpha, \beta, \gamma])$$

The distribution coefficient can be chosen to mimic the primitive shape a particular object. In this case, we chose  $\alpha = \beta = \gamma = 0.2 d_{\min}$ , with  $d_{\min}$  being the semi-minor axis of the relative CW orbit. In Table 1 we report the characteristics of the simulated scenario. We treat the case of a single reconnaissance orbit followed by a single optimization orbit.

The first orbit allows for the recognition of all the landmarks that fall in the field of view of the sensor: they are stored in the state and they are assigned a progressive number. The first orbit, during

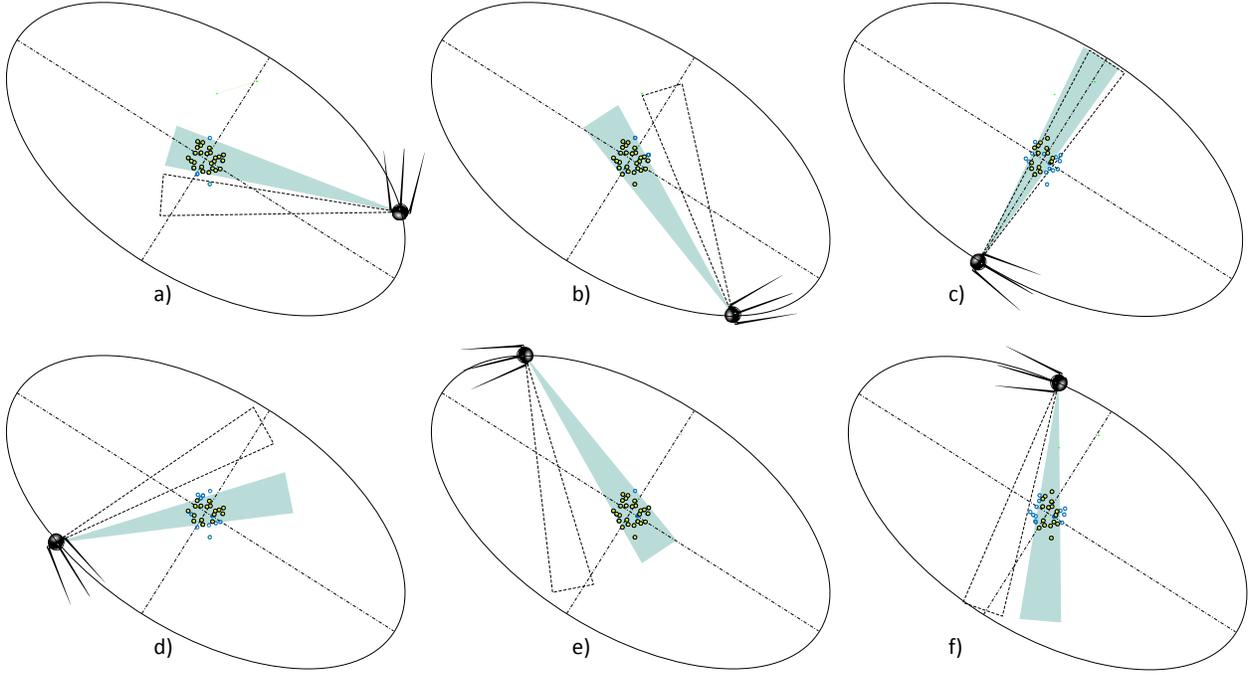


Fig. 5: Controlled orbit through CE optimization: the uncontrolled sensor is represented as the ghost dashed shape.

Table 1: Simulation parameters<sup>‡</sup>

ORBIT	SEMI-M AXIS	$d_{\max}$	29.2
	SEMI-M AXIS	$d_{\min}$	15.7
	MEAN MOTION	$n$	0.38
SET UP	SECTORS PER TURN	$s$	6
	BOXES PER SECTOR	$m$	5
	ITERATIONS PER BOX	$k$	10
	NUMBER OF LANDMARKS	$N_{\text{Ind}}$	30
SENSOR	RANGE	$r$	31.4
	BEARING	$\alpha$	10°
	MAX ACCELERATION	$a_{\max}$	0.8
	MAX ANGULAR SPAN	$\theta_{\max}$	60°
NOISE	MODEL	$\sigma_{\omega}$	0.005
	MEASUREMENT	$\sigma_v$	0.005

which no control is applied ( $\omega_{\theta} = 0$ ), is represented in Fig. 4. The sensor rotates together with the spacecraft, whose angular velocity is equal to the mean motion of the relative orbit. That is, the satellite completes a full revolution around its axis for each orbit.

Due to the elliptical shape of the orbit, the uncontrolled sensor spends a substantial amount of the orbit without acquiring any landmarks, even though its range would allow for potential observations (Fig. 4). In the figures, the observed landmarks are represented as green circles: note that, since the landmarks are positioned in 3D space, some of them are not de-

tected even if they appear in the 2D projection of the sensor’s field of view in Fig. 4. After the first orbit is completed, the Cross-Entropy routine is applied, starting from the first sector. In reference,<sup>6</sup> an optimization strategy based on the trace of the covariance matrix was presented and successfully simulated; in this paper, we present the results obtained by applying the cost function based on the time under observation, introduced in Section IV.

Although the CE routine can be computationally expensive if a fine trajectory quantization is sought, the nature of this method allows for parallelization. The optimization algorithm, in fact, can be run as soon as the satellite has completed the acquisition of first sector’s landmarks (beginning of the first orbit), in parallel with the acquisition of the upcoming sectors.

Since the time required for the CE algorithm to minimize the objective cost can be tailored by tuning the discretization step, the number of trial trajectories, CE iteration, etc., the optimized orbit can run in real time, keeping the sensor under control at all times. That is, by the time the satellite enters the second orbit, the optimized parameter vector pertaining the first sector,  $\lambda_1$ , is readily available.

The behavior of the satellite in the CE controlled orbit is shown in Fig. 5. The controlled sensor’s FOV

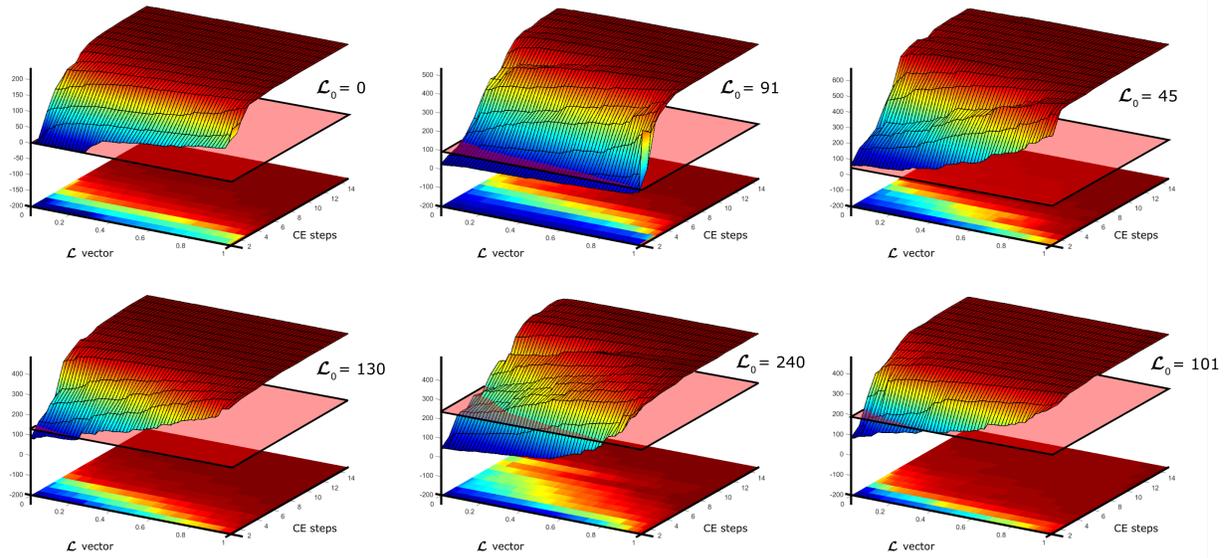


Fig. 6: Controlled orbit through CE optimization: ordered cost vectors  $\mathcal{L}_i$  plotted against the CE progressive optimization steps.

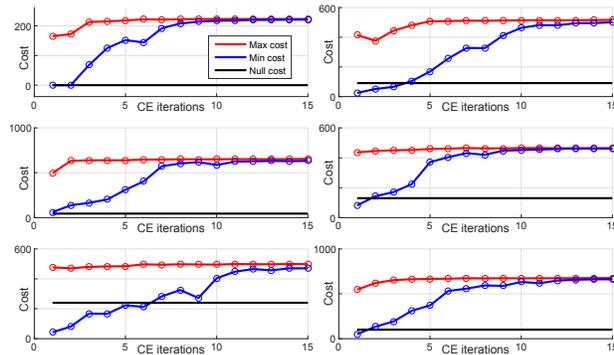


Fig. 7: Cost performances for the CE progressive optimization.

is the green triangular projection, whereas the dashed triangle represents the sensor's behavior when no controlled is applied. It is clearly noted that the controlled sensor is kept pointing at the landmarks at all times, maximizing the time under observation.

Apart from being limited in the acceleration profile, the sensor has to be also limited in terms of angular displacement. A switching control has been designed to avoid unrealistic trajectories: the parameters vectors  $\lambda_i$  drawn from the normal distribution are screened and discarded if the control vector generates a trajectory such that  $|\max(\theta_i)| \leq \theta_{\max}$ .

A policy to limit the actuation cost has been used in order to prevent unnecessary control. At the beginning of each CE optimization, the null cost  $\mathcal{L}_{0,s}$

is computed for sector  $s$ . This is the cost in the case no control is being applied, i.e., the sensor kinematics is governed merely by the angular velocity at the previous timestep  $\omega_{s-1,m}$ .

The null cost is used as a reference to compare with the performance of the optimization routine. The plots in Fig. 6 represent the cost vector optimization process.

For each CE iteration (in this case  $N_{CE} = 15$ ), the cost vector  $\mathcal{L}_{N_{CE}}$  is computed and ordered as described in the algorithm in Section II. The stacking of the subsequent cost vectors creates the three-dimensional surfaces in the figures. The horizontal red plane depicts the null cost performance: when the surface is above the plane, the CE performances are superior than the non controlled case. As can be seen from these results, even the lowest  $\mathcal{L}$ -vector percentiles are risen above the null-cost plane in the first two-to-three CE iterations. In all cases, the top percentiles are always above the null cost starting from the initial CE iteration.<sup>6</sup>

In general, the CE optimization allows convergence to the optimal solution by progressively reducing the difference between the lower and upper percentiles of the cost vectors  $\mathcal{L}$ .

In Fig. 7, the maximum and minimum cost for each CE optimization step is represented: the strategy is capable of finding the maximum cost very early in the process (red lines). Then, the method takes 7-to-10 iterations to even out (in most cases monotonically)

the range between the minimum and maximum cost vectors (blu lines). The solid black lines represent the null cost case, which is substantially outperformed in each of the orbital sectors.

## VII. CONCLUSIONS

This paper presents a novel approach for solving the active self-localization problem during relative navigation in orbit using Cross Entropy (CE) minimization, expanding the previous work in a 2D framework.<sup>6</sup> Using the Clohessy-Wiltshire model, a real case chaser-target orbital scenario was presented.

By jointly considering the planning, control and estimation problems it was possible to balance the control actuation costs and the obtainable localization uncertainty: this has been obtained by incorporating an uncertainty measure in the cost functions, which is then utilized to select near-optimal trajectories in terms of estimation uncertainty. Results for the cost function based on the time under observation confirmed the validity of the method.

It is well known<sup>15</sup> that the main drawback of Cross Entropy implementation in control design is due to the substantial computational efforts required during optimization: to overcome this issue, by discretizing the orbit in a finite time horizon sectors, it was possible to use parallelization and to hence design a real-time controller. In our approach, the optimization is run in parallel with landmark acquisition and no hold-time is needed for computation.

Future work will focus on the validation of the presented method through the aid of experimental data, and high-fidelity simulation using a satellite simulator and a realistic orbital scenario.

## REFERENCES

- [1] R. Ambrose, B. Wilcox, B. Reed, L. Matthies, D. Lavery, and D. Korsmeyer, "Draft robotics, tele-robotics and autonomous systems roadmap," *NASAs Space Technology Roadmaps, National Aeronautics and Space Administration*, 2010.
- [2] R. Sim and N. Roy, "Active exploration planning for slam using extended information filters," in *Proc. 20th Conf. Uncertainty in AI*, vol. 2, 2004.
- [3] C. Leung, S. Huang, and G. Dissanayake, "Active slam in structured environments," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1898–1903, IEEE, 2008.
- [4] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, P. L'Ecuyer, *et al.*, *The Cross-Entropy Method For Optimization*, vol. 31. 2013.
- [5] D. P. Kroese and R. Y. Rubinstein, "Monte carlo methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 1, pp. 48–58, 2012.
- [6] M. Kontitsis, P. Tsiotras, and E. Theodorou, "An information-theoretic active localization approach during relative circumnavigation in orbit," in *AIAA Guidance, Navigation, and Control Conference*, (San Diego, California, USA), p. 872, 4-8 January 2016.
- [7] K. Alfriend, S. R. Vadali, P. Gurfil, J. How, and L. Breger, *Spacecraft Formation Flying: Dynamics, Control And Navigation*, vol. 2. Butterworth-Heinemann, Burlington, MA, 2009.
- [8] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*. AIAA, 2003.
- [9] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*, vol. 12. Springer Science & Business Media, 2001.
- [10] H. Curtis, *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, Burlington, MA, 2013.
- [11] M. Kontitsis, E. A. Theodorou, and E. Todorov, "Multi-robot active slam with relative entropy optimization," in *American Control Conference*, (Washington, DC, USA), pp. 2757–2764, 17-19 June 2013.
- [12] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications Inc., New York, 2012.
- [13] S. Augenstein, S. M. Rock, P. Enge, and C. J. Tomlin, *Monocular Pose And Shape Estimation Of Moving Targets, For Autonomous Rendezvous And Docking*. PhD thesis, Stanford University, 2011.
- [14] B. E. Tweddle, *Computer Vision-based Localization And Mapping Of An Unknown, Uncooperative And Spinning Target For Spacecraft Proximity Operations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2013.
- [15] C. Li and P. K.-S. Tam, "An iterative algorithm for minimum cross entropy thresholding," *Pattern Recognition Letters*, vol. 19, no. 8, pp. 771–776, 1998.