IAC–16–C1.3

Performance Analysis of Three Cost Policies for The Control of a Camera in Relative Circumnavigation Scenarios

Andrea Antonello*, Andrea Carron§, Ruggero Carli§, Panagiotis Tsiotras†
*Centre of Studies and Activities for Space - G. Colombo, University of Padova, Italy
§ Department of Information Engineering, University of Padova, Italy
†School of Aerospace Engineering, Georgia Institute of Technology, Atlanta GA - USA

Abstract - In this paper, we address the relative navigation problem of a chaser circumnavigating a target. The chaser has an on-board camera and observes a set of features on the target; the goal is to obtain a detailed map of the landmarks. By controlling the yaw-rotation of the sensor it is possible to maximize the time allocated to landmark observation. An Extended Kalman Filter (EKF) provides state uncertainty information, which can then be used to design a cost function to be minimized by the optimal yaw controller. Three different cost functions are designed and simulated, and their performances are compared with the case of a fixed, nadir-pointing camera. The analysis of localization uncertainties for different sets of initial conditions confirmed the superior performance of the proposed novel control methodology.

I. Introduction

Recent advantages in the field of computing hardware, coupled with the enhancement of sensor performance, have paved the way for autonomous navigation to become a reality. In this framework, map generation and localization are the key for successful autonomous operation and navigation of robots. This is particularly true in the case of orbiting vehicles, in which autonomous formation flying and docking could enable new designs of space systems and enable operations such as inspection, refurbishment, repair, refueling, etc.[1]

Autonomy is heavily dependent on the capability for a satellite to accurately estimate its position with respect to another target. State-of-the-art proximity-navigation policies solve the problem of control and estimation separately. That is, the mutual effects the controller induces on the estimator (and vice versa) are not considered.[2–6]

In this work, we depart from the separation principle of stochastic control, and integrate planning and stochastic optimization with localization in order to perform control of autonomous spacecraft under uncertainty conditions. We approach the problem of a chaser satellite circumnavigating a target satellite in a simplified two-dimensional orbit. The chaser has a vision sensor and observes a set of landmarks on the target: its goal is to obtain a detailed map of these features. The control acts on the yaw-rotation of the sensor in order to maximize the time allocated to landmark observation.

A certain cost function (e.g., the estimation accuracy of the detected landmarks) drives which feature to be selected next, and hence also drives the next control action. An Extended Kalman Filter (EKF) provides the state uncertainty, which can then be used to design the cost function. Since the optimization problem is difficult to solve, we resort to cross-entropy (CE) minimization, which iteratively searches for the near-optimal control action. The final result is a trajectory that achieves the predefined goal in the state space and reduces total localization uncertainty, while limiting actuation cost.

Three different cost functions are proposed and simulated, and their performances compared with the case of a fixed, nadir-pointing camera.

II. Problem Formulation

II.i Relative Navigation in Orbit

We consider an observer and a chaser satellite circumnavigating along a circular trajectory of radius R_{orb} having linear velocity V and orbital velocity $\omega_\phi = V/R_{\text{orb}}$. Typical relative orbits of two satellites flying in formation would result in a 2×1 elliptical orbit.[7] The incorporation of an elliptical reference orbit for the chaser satellite is straightforward.

The objective of the chaser satellite is to obtain a map of a certain set of landmarks that are present on the target satellite. These are features such as edges, patches, arrays of LEDs, etc. For the sake

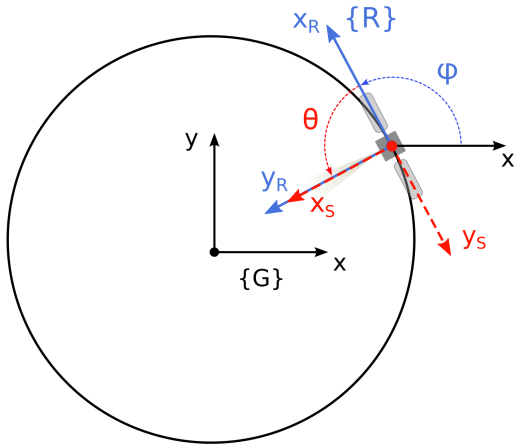


Fig. 1: Problem set up and frame of reference definition

of simulations, we are considering the landmarks as single points distributed in the xy plane. The process of gathering information on the landmark positions is achieved through the application of a Simultaneous Localization and Mapping Algorithm (SLAM), which also allows for the simultaneous improvement of the chaser localization.

The satellite has an onboard sensor which is free to rotate around the axis that is normal to the xy plane going through its center of mass. According to the notation in Fig. 1, 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 ϕ and θ, which respectively represent the heading direction of the satellite and the sensor bearing. Note that in this notation, frame {S} has a positive $\pi/2$ angular offset with respect to {R}: that is, when the sensor points to Nadir, the bearing is set to zero.

Detection of the landmarks — whose number and locations are to be determined — can be obtained with the aid of various sensors, i.e., sonars, lasers, LIDARs, cameras, ecc. In this paper, we use a range and bearing sensor, which outputs the distance and angular displacement of the detected feature in the {S} reference frame.

The sensor is a faithful representation of a real device, having a limited field of view, a fixed angular span and maximum angular acceleration.

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

II.ii State Model

The state model of the orbiting satellite, augmented with the position of the landmarks and expressed in differential form, is the following:

$$\begin{bmatrix} \mathrm{d}x(t) \\ \mathrm{d}y(t) \\ \mathrm{d}\phi(t) \\ \mathrm{d}\theta(t) \\ \mathrm{d}\mathbf{p}_1(t) \\ \vdots \\ \mathrm{d}\mathbf{p}_N(t) \end{bmatrix} = \begin{bmatrix} V\cos\phi(t)\mathrm{d}t \\ V\sin\phi(t)\mathrm{d}t \\ \omega_\phi(t)\mathrm{d}t \\ \omega_\theta(t)\mathrm{d}t \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_4 \\ \mathbf{0}_{2N\times 4} \end{bmatrix} \mathrm{d}\mathbf{w}(t) \quad [1]$$

where x and y indicate the position of the chaser satellite in the {G} frame, whereas angles ϕ and θ are the rotation of the chaser and the sensor expressed in frames {G} and {R} respectively. The landmark positions are expressed in the global frame, and yield an augmented state $\mathbf{x} \in \mathbb{R}^{2N+4}$.

In the model, $\mathrm{d}\mathbf{w} \in \mathbb{R}^4$ represents Wiener process noise, with covariance matrix $\mathbf{\Sigma}_w = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2)$.

In the simulations, a discrete-time state model derived from [1] using Euler discretization will be used:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \\ \theta_{k+1} \\ \mathbf{p}_{1k+1} \\ \vdots \\ \mathbf{p}_{Nk+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \phi_k \\ \theta_k \\ \mathbf{p}_{1k} \\ \vdots \\ \mathbf{p}_{Nk} \end{bmatrix} + \begin{bmatrix} V\cos(\phi_k)\,\delta t \\ V\sin(\phi_k)\,\delta t \\ \omega_{\phi_k}\,\delta t \\ \omega_{\theta_k}\,\delta t \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{I}_4 \\ \mathbf{0}_{2N\times 4} \end{bmatrix} \mathbf{w}(t_k) \quad [2]$$

where δt is the discretization step and $\mathbf{w}(t_k)$ denotes white Gaussian noise. In this model, the only control parameter is the angular velocity of the sensor ω_{θ_k} at time t_k. Since the sensor has a limited field of view, the capability of controlling ω_{θ_k} may have a significant influence on the uncertainty reduction of the state estimate.

II.iii Measurement Model

Detection of the landmarks occurs only if they are within the field of view and range of the sensor. When a feature is detected, the sensor outputs $\mathbf{z} = (r, \alpha)$, where r is the range and α is the bearing of the observed landmark . The measurement model, expressed in continuous form, is given by:

$$\mathbf{z}(t) = {}^S_R\mathbf{R}(\theta(t)){}^R_G\mathbf{R}(\phi(t))(\mathbf{p}_i(t) - \mathbf{p}_R(t)) + \mathbf{v}(t) \quad [3]$$

where $\mathbf{p}_i = (p_{x_i}, p_{y_i})$ and $\mathbf{p}_R = (x, y)$ are the position of the landmarks and the observer satellite, re-

spectively. The term $\mathbf{v}(t)$ corresponds to the observation noise of the sensor which is considered zero-mean Gaussian with covariance matrix $\mathbf{\Sigma}_v = \text{diag}(\sigma_{\text{I}}^2, \sigma_{\text{II}}^2)$. The matrices ${}^S_R\mathbf{R}(\theta(t))$ and ${}^R_G\mathbf{R}(\phi(t))$ express rotational transformations from the global {G} to the observer frame of reference {R} and from {R} to the sensor reference frame {S}, respectively. In compact form the observation model is written as:

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}(t). \quad [4]$$

However, in a real scenario, measurements will be taken discretely, according to the sampling strategy adopted. The measurement model, in discrete time form, can therefore be expressed as:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \quad [5]$$

where $\mathbf{h}(\mathbf{x}_k)$ is given by:

$$\mathbf{h}(\mathbf{x}_k) = \begin{bmatrix} \sqrt{(x_L - x_k)^2 + (y_L - y_k)^2} \\ \tan^{-1}\left(\dfrac{y_L - y_k}{x_L - x_k}\right) - \phi_k - \theta_k \end{bmatrix} \quad [6]$$

In order to proceed, we consider the discretized version of the dynamics in [2]. The dimension of the state is initially $\mathbf{x} \in \mathbb{R}^4$ and is augmented every time a new landmark is detected. We divide the design of the Extended Kalman Filter into prediction and update (see Section III.v).

III. Main problem

We want to estimate the position of the landmarks by evaluating the measurements taken by the sensor. To do this, we control the rotation of the sensor in the x-y plane to maximize the performance over a finite time horizon. The objective is to minimize a cost function that encloses both the final uncertainty of the estimate and the actuation cost. The cost function can be written as:

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \|e^2(t_N)\| + \int_0^{t_N} \left(Q(\mathbf{x}) + \frac{1}{2}\mathbf{u}^T R\mathbf{u}\right) \mathrm{d}t \quad [7]$$

where $\|e^2(t_N)\|$ is the terminal cost at a certain time horizon setpoint $T = t_N$. Since we do not know this error, a strategy for its approximation needs to be designed.

This strategy is obtained by approximating the error with a measure of the estimation uncertainty. We introduce a strategy based on the covariance matrix trace, along with alternative strategies based deriving from the observation of the landmarks. All these strategies evaluate both the terminal performances of the piecewise control trajectory and the actuation cost needed to achieve it.

III.i Trace of the covariance matrix (TCM)

The first strategy uses the trace of the covariance matrix as a measure of the uncertainty for the state estimate given by an Extended Kalman Filter (EKF):

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \psi^1(\mathbf{x}_{t_N}) + \sum_{k=0}^{N} \left(Q(\mathbf{x}_{t_N}) + \frac{1}{2}\mathbf{u}(\mathbf{t_k})^T R\mathbf{u}(\mathbf{t_k})\right) \quad [8]$$

in which the terminal cost is:

$$\psi^1(\mathbf{x}_{t_N}) = \text{trace}(\Sigma(t_N)). \quad [9]$$

For simplicity, let in the following $Q(x) = 0$ to obtain:

$$\mathcal{L}_{\text{tcm}} = \text{trace}(\Sigma(t_N)) + \sum_{k=0}^{N} \left(\frac{1}{2}\mathbf{u}(\mathbf{t_k})^T R\mathbf{u}(\mathbf{t_k})\right) \quad [10]$$

III.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 sampled trajectory, we count the number of landmarks seen by the sensor at each iteration (T_i). The total number of observed landmarks is then summed:

$$\psi^2(\mathbf{x}_{t_N}) = \sum_{i=1}^{N} T_i \quad [11]$$

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

$$\mathcal{L}_{\text{tuo}} = \sum_i T_i + \sum_{k=0}^{N} \left(\frac{1}{2}\mathbf{u}(\mathbf{t_k})^T R\mathbf{u}(\mathbf{t_k})\right) \quad [12]$$

III.iii Modified Time under observation (MTUO)

The previous strategy maximizes the time under observation of the target, but may lead to overlooking features in relatively remote areas of the working space. In order to avoid this situation, we modify Eq. [14] by defining the trajectory cost based not only on the TUO, but also on the number of different landmarks observed. That is, we count the TUO and the number of different landmarks observed for each iteration (N_{lnd_i}) and we define MTUO as the normalized sum:

$$\psi^3(\mathbf{x}_{t_N}) = \sum_i (\hat{T}_i + \hat{N}_{\text{lnd}_i}) \quad [13]$$

Normalization is mandatory in order to correctly compare and sum the two partial costs. We normalize the two terms as follows:

$$\hat{T}_i = \frac{T_i - T_{i,\min}}{T_{i,\max} - T_{i,\min}}$$

$$\hat{N}_{\text{lnd}_i} = \frac{N_{\text{lnd}_i} - N_{\text{lnd}_{i,\min}}}{N_{\text{lnd}_{i,\max}} - N_{\text{lnd}_{i,\min}}}$$

So that $\{\hat{T}_i, \hat{N}_{\text{lnd}_i}\} \in [0,1]$ and the new cost $\mathcal{L}_{\text{mtuo}} \in [0,2]$.

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

$$\mathcal{L}_{\text{mtuo}} = \sum_i [\hat{T}_i + \hat{N}_{\text{lnd}_i}] + \sum_{k=0}^{N} \left(\frac{1}{2} \mathbf{u}(\mathbf{t_k})^T R \mathbf{u}(\mathbf{t_k}) \right) \quad [14]$$

III.iv Cross Entropy Minimization

We now present the Cross Entropy minimization algorithm, and show how it can be used to solve a certain class of stochastic optimal control problems. Assume that the following stochastic dynamics system is given:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u})\delta t + g(x)\mathrm{d}\mathbf{w} \quad [15]$$

where $\mathbf{x} \in \mathbb{R}^n$ 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 Wiener process with covariance $\mathbf{\Sigma}_w$. Our objective is to minimize a cost function of the form:

$$\min \mathbb{E}_p[\mathcal{L}(\mathbf{x}, \mathbf{u})], \quad [16]$$

where the expectation in [16] is with respect to the trajectories of [15]. Assuming that $\mathbf{u}(t)$ 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 λ. According to the CE minimization method,[8] we rewrite the cost function as follows:

$$J(\lambda) = \mathbb{E}_p\ [\mathcal{L}(\lambda)] = \int p(\lambda)\ \mathcal{L}(\lambda)\mathrm{d}\lambda \quad [17]$$

where $p(\lambda)$ is the probability density function corresponding to sampling trajectories based on [15]. This cost function can be approximated as follows, performing importance sampling from a proposal probability density $q(\lambda)$,

$$\hat{J}(\lambda) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \left[\frac{p(\lambda_i;\mu)}{q(\lambda_i)} \mathcal{L}(\lambda_i) \right] \quad [18]$$

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

$$q^*(\lambda) = \underset{q}{\text{argmin}}\ \text{Var}\left[\frac{p(\lambda;\mu)}{q(\lambda)} \mathcal{L}(\lambda) \right] = \left[\frac{p(\lambda;\mu)\mathcal{L}(\lambda)}{J(\lambda)} \right] \quad [19]$$

and it is the optimal (with respect to variance) importance sampling density. The goal of CE is to find the parameters $\psi \in \Psi$ in the parametric class of pdfs $p(\lambda;\psi)$, such that the probability distribution $p(\lambda;\psi)$ approaches the optimal distribution $q^*(\lambda)$ given in [19]. The optimal parameters can be approximated numerically as:

$$\psi^* \approx \underset{\psi}{\text{argmax}}\ \frac{1}{N_s} \sum \mathcal{L}(\lambda)\ \ln[p(\lambda;\psi)], \quad [20]$$

in which the Kullback-Leibler divergence was used as the distance metric between $q^*(\lambda)$ and $p(\lambda;\psi)$. We want to compute the value of λ that satisfies the following equation:

$$\mathbb{P}(\mathcal{L} \leq \epsilon) = \mathbb{E}_{p(\lambda;\mu)}[I_{\{\mathcal{L}\leq\epsilon\}}] \quad [21]$$

where ϵ is a small constant and I is the indicator function. Using [18], 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 λ_i are i.i.d samples drawn from $p(\lambda;\psi)$. Based on [20], the goal is to find the optimal ψ^*, which is defined as:

$$\psi^* \approx \underset{\psi}{\text{argmax}}\ \frac{1}{N_s} \sum I_{\{\mathcal{L}(\lambda_i)\leq\epsilon\}}\ \ln[p(\lambda_i;\psi)], \quad [22]$$

where now the samples λ_i are generated according to probability density $p(\lambda;\mu)$. In order to estimate the above probability, it is infeasible to use a brute force method, e.g. Monte-Carlo:[9] this is because the event $\{\mathcal{L} \leq \epsilon\}$ is rare. An alternative is to start with $\epsilon_1 > \epsilon$ for which the probability of the event $\{\mathcal{L} \leq \epsilon_1\}$ is equal to some $\rho > 0$. Thus, the value ϵ_1 is set to the ρ-th quantile of $\mathcal{L}(\lambda)$ which means that ϵ_1 is the largest number for which:

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

The parameter ϵ_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 ψ_1 for the level ϵ_1 is calculated according to [22] using the value ϵ_1. This iterative procedure terminates when $\epsilon_k \leq \epsilon$, in which case the corresponding parameter ψ_k is the optimal one and thus $\psi^* = \psi_k$.

To summarize, in order to find the optimal trajectory λ^* and the corresponding optimal parameters ψ_k, the process of estimating rare event probabilities is iterated until $\epsilon \to \epsilon^*$, where $\epsilon^* = \min \mathcal{L}(\epsilon)$. Since

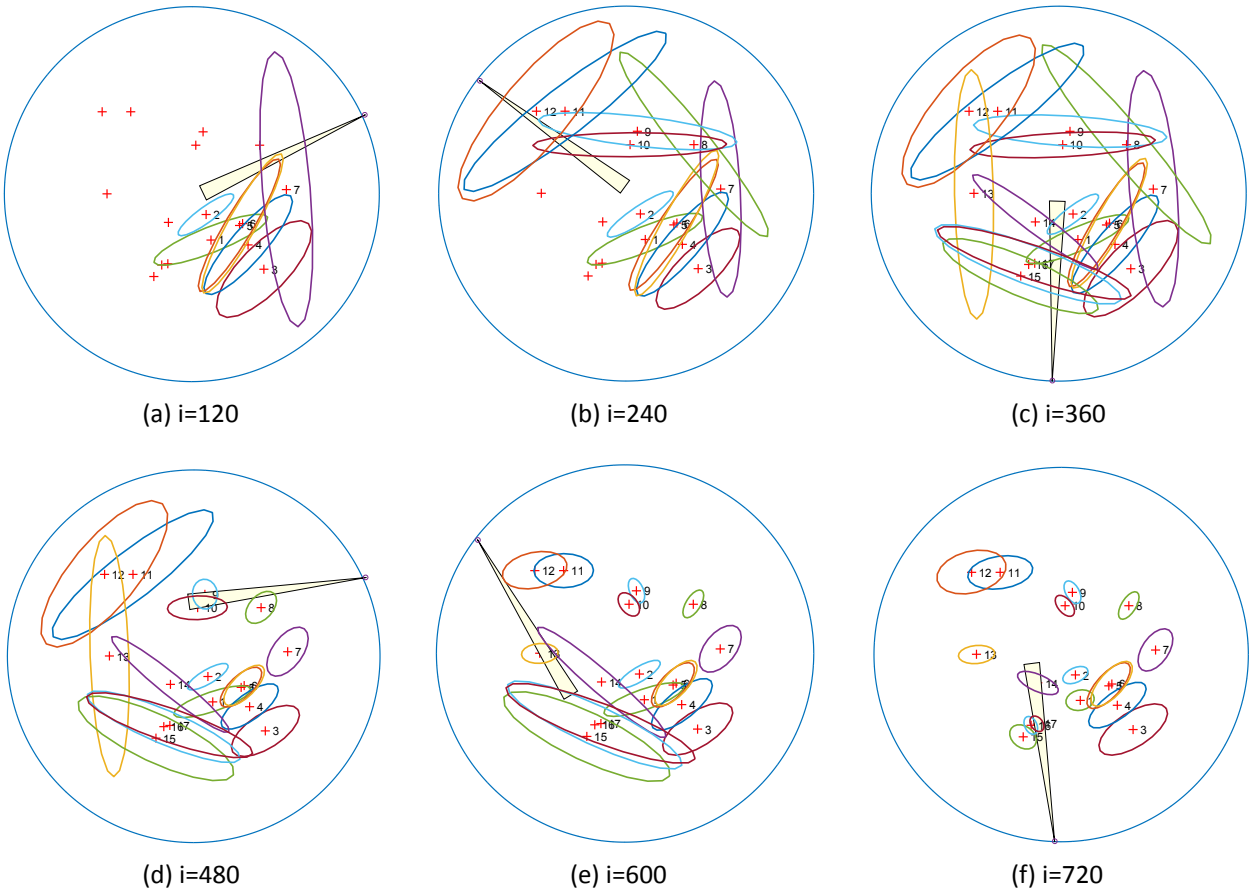


Fig. 2: Recognition phase (a-c) and Cross Entropy minimization (d-f) at different timesteps: note the progressive numbering assigned to the landmarks by the algorithm. The uncertainty of each feature is displayed as a covariance ellipse.

ϵ^* is not known a priori, we choose as ϵ^* the value of ϵ for which no further improvement within a predefined tolerance in the iterative process is observed. The overall problem is summarized in the table below.

The proposed algorithm consists of two phases:

- the *recognition* phase, during which the first orbit the measurements taken by the chaser provide a first estimation of all the 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

Cross Entropy Algorithm

1: Draw N samples for λ 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 λ and sort them in ascending order.
3: Select the best performing ρ-th percentile and find the optimal parameters $(\mu_\xi^*, \Sigma_\xi^*)$ which maximize

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

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

chaser runs an Extended Kalman filter algorithm whose state is augmented whenever a measurement related to a new landmark is collected.

III.v Recognition Phase

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

$$\mathbf{x}_k^{(\tilde{N})} = \begin{bmatrix} x_k & y_k & \phi_k & \theta_k & \mathbf{p}_k^{(1)} & \mathbf{p}_k^{(2)} & \dots & \mathbf{p}_k^{(\tilde{N})} \end{bmatrix}^T.$$

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

Prediction step

The update equation is

$$\begin{bmatrix} \hat{x}_{k+1|k} \\ \hat{y}_{k+1|k} \\ \hat{\phi}_{k+1|k} \\ \hat{\theta}_{k+1|k} \\ \hat{p}^{(1)}_{k+1|k} \\ \hat{p}^{(2)}_{k+1|k} \\ \vdots \\ \hat{p}^{(\tilde{N})}_{k+1|k} \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k} \\ \hat{y}_{k|k} \\ \hat{\phi}_{k|k} \\ \hat{\theta}_{k|k} \\ \hat{p}^{(1)}_{k|k} \\ \hat{p}^{(2)}_{k|k} \\ \vdots \\ \hat{p}^{(\tilde{N})}_{k|k} \end{bmatrix} + \begin{bmatrix} V\cos(\hat{\theta}_{k|k})\delta t \\ V\sin(\hat{\theta}_{k|k})\delta t \\ \omega_{\phi_k}\delta t \\ \omega_{\theta_k}\delta t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

or, in a more compact form,

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

The update of the covariance matrix is given by

$$P^{(\tilde{N})}_{k+1|k} = F_k P^{(\tilde{N})}_{k|k} F_k^T + Q_k, \qquad [23]$$

where

$$F_k = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}. \qquad [24]$$

The matrix F_k has the following structure

$$F = \begin{bmatrix} F_k^{\text{mot}} & 0 \\ 0 & \mathbb{I}_{\tilde{N}} \end{bmatrix}, \qquad [25]$$

where F_k^{mot} is given by the following expression

$$\begin{aligned} \mathbf{F}_k^{\text{mot}} &= \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \phi} & \frac{\partial f_1}{\partial \theta} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial \phi} & \frac{\partial f_4}{\partial \theta} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -V\sin(\hat{\phi}_{k|k})\delta t & 0 \\ 0 & 1 & V\cos(\hat{\phi}_{k|k})\delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \qquad [26]$$

Finally the matrix Q_k is of the form

$$Q_k = \begin{bmatrix} \Sigma_w & 0 \\ 0 & 0_N \end{bmatrix},$$

where 0_N is a null matrix of dimension N.

Update step

Assuming we have the information provided by the range and bearing sensor $\mathbf{z} = [r, \alpha]$, 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 $\mathbf{z}^{(1)}_{k+1}$ which is given by all the measurements collected from already seen landmarks, and the second $\mathbf{z}^{(2)}_{k+1}$ which represents measurements collected by observing new landmarks. The measurement model can be written as

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

Let us compute the Jacobian of the observation model with respect to the robot pose and the observed landmark coordinates. At iteration $k+1$ we obtain

$$\mathbf{H}_{k+1} = \left.\frac{\partial \mathbf{h}^{(1)}_{k+1}}{\partial \mathbf{x}^{(\tilde{N})}}\right|_{\hat{x}_{k+1|k}} \qquad [27]$$

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

$$\begin{aligned} K_{k+1} &= P^{(\tilde{N})}_{k+1|k} H^T_{k+1} \left(H_{k+1} P^{(\tilde{N})}_{k+1|k} H^T_{k+1} + R_{k+1} \right)^{-1} \\ x^{(\tilde{N})}_{k+1|k+1} &= x^{(\tilde{N})}_{k+1|k} + K_{k+1}\mathbf{z}^{(1)}_{k+1}. \\ P^{(\tilde{N})}_{k+1|k+1} &= (\mathbb{I} - K_{k+1}H_{k+1}) P^{(\tilde{N})}_{k+1|k} \end{aligned}$$

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

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

Then we can extend the state

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

and the covariance matrix

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

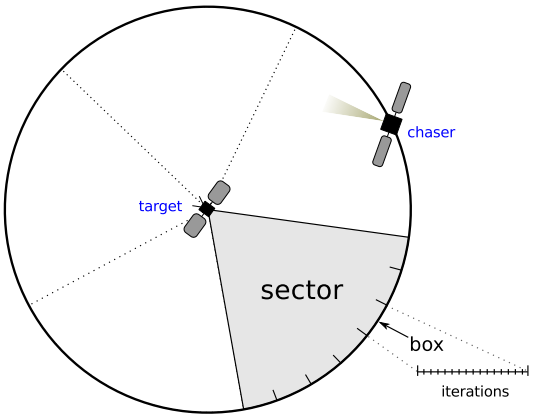


Fig. 3: Discretization strategy for cross-entropy algorithm

where

$$P^{(\tilde{N}+1,\tilde{N})} = \left(P^{(\tilde{N},\tilde{N}+1)}\right)^T = \left[\begin{array}{c} \frac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial x_k} \\ \frac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial y_k} \\ \frac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \phi_k} \\ \frac{\partial \hat{p}^{(\tilde{N}+1)}_{k+1|k+1}}{\partial \theta_k} \end{array}\right]\Bigg|_{(\hat{x}_{k+1|k},\bar{\mathbf{z}}_{k+1})}$$

and

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

III.vi Incremental Estimation Phase

After the recognition phase an initial guess of the landmark's 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 the following steps are repated:

1. We draw N_{traj} random possible acceleration trajectories for the sensor, $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{N_{traj}}\}$, from a Gaussian distribution with parameter v_i (the particular controller used in this paper will be explained in Section III.vii).

2. For all λ we simulate the behavior of the camera running an Extended Kalman filter as explained in Section III.v.

3. Once the state has been estimated at any time instant we can evaluate one of the cost function presented in Section III 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.

III.vii Controller

The controller acts on the angular velocity of the sensor, ω_θ. Recalling Eq. [12], we can rewrite the discretized cost as:

$$\hat{\mathcal{L}}(\mathbf{x},\mathbf{u}) \approx \psi(\mathbf{x}_N) + \frac{1}{2}R\sum_{k=0}^{N}\omega_{\theta_k}^2, \qquad [28]$$

where in Eq. [8] we let $Q(\mathbf{x}) = 0$ and $\psi(\mathbf{x}_N) = \|e^2(N)\|$. The control law is parametrized as follows:

$$\begin{aligned} \omega_{\theta_k} &= \mathbf{u}(\omega_{\theta_{k-1}}, \eta(k-1;\lambda)) \\ &= \omega_{\theta_{k-1}} + \eta(k-1;\lambda)\delta t \end{aligned} \qquad [29]$$

where $\eta(k-1;\lambda)$ 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.

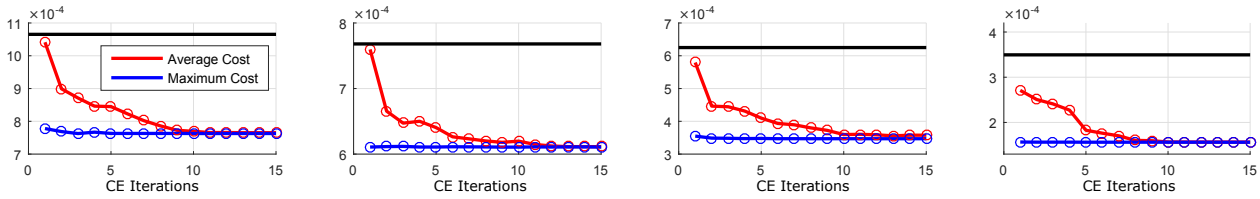
Each constant acceleration η_m is being applied for a constant duration δt_i, where $t_{\text{sect}} = \sum_{i=1}^{m}\delta t_i$. The sum of all time intervals is fixed and equal to the time horizon corresponding to the duration of each sector s (refer to the table in Section IV). The parameter vector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

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

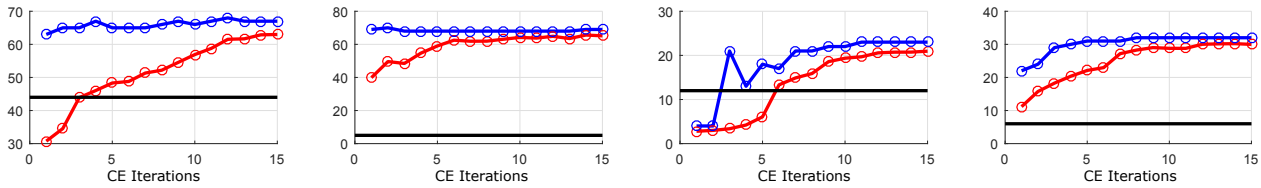
In the numerical simulations, and without loss of generality, we maintain the controller timestep constant $\delta t_i = \delta t_m = t_{\text{sect}}/m$. The accelerations η_i are initially obtained from a uniform distribution $\mathcal{U}([\eta_{min}, \eta_{max}])$, where the bounds are dictated by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

III.viii Algorithm Set Up

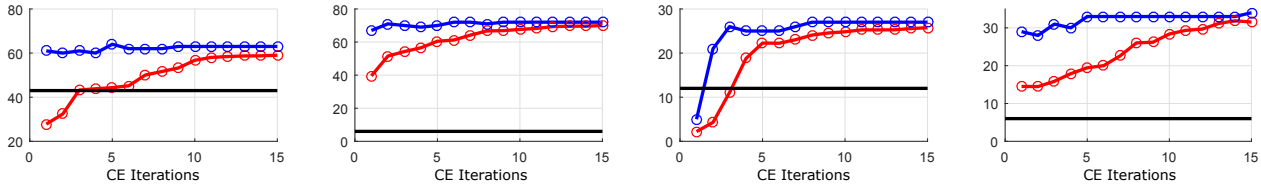
To evaluate the proposed control policies, we consider the scenario of a satellite circumnavigating another satellite in orbit, while observing a set of feature points (landmarks) on the target satellite. The objective of observing the satellite is to accurately localize the landmarks.



(a) Cost policy based on the $\mathcal{L}_{trace}$ cost. CE optimization for different orbit sectors.



(b) Cost policy based on the $\mathcal{L}_{\text{tuo}}$ cost. CE optimization for different orbit sectors.



(c) Cost policy based on the $\mathcal{L}_{mtuo}$ cost. CE optimization for different orbit sectors.

Fig. 4: Cross Entropy optimization for the proposed cost policies. The solid black line represents the null cost. In this case, $N_{CE} = 15$.

Problem Algorithm

for $s = 1$ **to** total sectors per turn **do**
  Selected initial distribution parameters $\mathbf{v}_0$
  for $i = 1$ **to** total CE optimization steps **do**
    Draw N_{traj} random acceleration vectors $\lambda \in \mathbb{R}^m$ from distribution with parameters v_i
    for $j = 1$ **to** N_{traj} **do**
      Run a simulated EKF with the input N^j_{traj}
      Evaluate the cost function [8] and store the value
    end for
    Sort all the cost function values in ascending order
    Take the ρ-th quantile Run the cross entropy optimization [22] and extract the new distribution parameters $\mathbf{v}_{i+1}$
  end for
  Apply the obtained near optimal control law $\lambda \in \mathbb{R}^m$ to sector s.
end for

The optimization step follows a first full orbit around the object in which the chaser satellite observes the landmarks in a *recognition* mode: in this first part, no control is applied to the sensor, which, for example, points towards the center of the orbit at all times.

After a first turn has been completed, and the state vector $\mathbf{x}$ has been augmented to dimension $\mathbb{R}^{4+2\times\mathcal{N}}$ through landmark observations[†], 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 result in 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 is divided in m sampling boxes, where m is the size of the control action vector λ. Depending on the implemented discretization, each box consists of l iterations. That is, for each box m_i, constant control parameters λ_i are applied for l

[†] $\mathcal{N} \leq N$

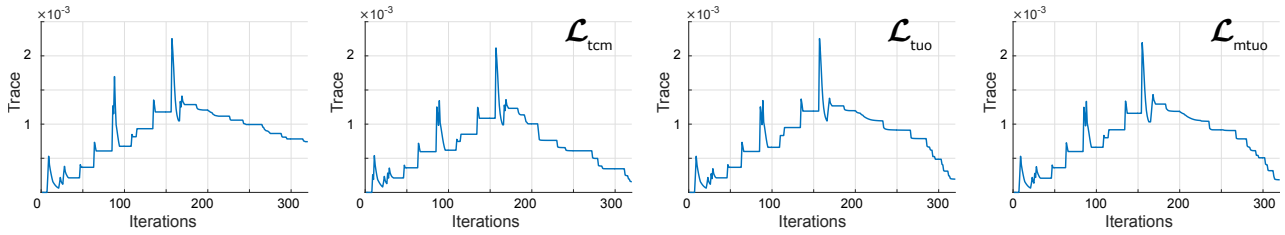


Fig. 5: Trace of the covariance matrix in the uncontrolled and controlled cases.

Fig. 6: Landmark observations' potential map in the uncontrolled and controlled cases.

number of iterations of the box. Fig. 3 illustrates the discretization strategy.

The cross entropy control strategy for the simulation is listed in the Problem Algorithm routine: N_{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_{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_{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.

IV. Simulation results

We present the results for the three proposed cost policies. All the simulations, in order to maintain consistency, comprise the same number and location of landmarks, with the same orbit and sensor characteristics. The simulation parameters are listed in Table 1.

The position of the landmarks is obtained by randomly extracting $N_{\text{lnd}} \times 2$ values in $\mathcal{N}(0, \frac{R}{2})$.

The simulation starts with a first orbit in *recognition mode*, during which the sensor is kept Nadir-pointing (that is, $\mathbf{x}(4) = \theta = 0$). Data regarding the landmarks is collected, along with the uncertainty of the EKF extimate. Graphically, the uncertainty can

Table 1: Simulation parameters

Orbit	Orbit radius	R	5
	Angular velocity	ω	0.2
Set up	Sectors per turn	s	6
	Boxes per sector	m	10
	Iterations per box	l	20
	Number of landmarks	N_{lnd}	10
Sensor	Range	r	4.7
	Bearing	α	5
	Max acceleration	a_{max}	2
Noise	Model	σ_ω	0.002
	Measurement	σ_v	0.002

be represented with covariance ellipses.

In Fig. 2 (a-c), the *recognition mode* simulation is illustrated at different steps during the first turn. Initial uncertainty is dictated by the simulation and sensor noise and is influenced by the number of measurements taken.

After this first initial turn, the cross entropy control is applied. The algorithm's ability to drastically reduce the uncertainty and to improve the mutual localization has been demonstrated for all three different control strategies.

In Fig. 2 (d–f), the cross entropy minimization orbit is depicted at three different steps, in which the shrinking in the size of the covariance matrix ellipses can be clearly seen (in this particular case, the cost strategy based on $\mathcal{L}_{\text{tcm}}$ has been implemented). Note that the beam points always towards a group of landmarks, whereas in the recognition mode the sensor is

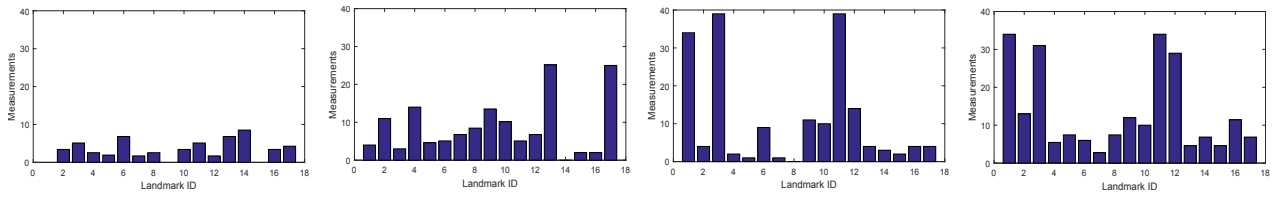


Fig. 7: Cumulative landmark detection in he uncontrolled and controlled cases.

kept pointed at the center of the orbit.

Analysis of the proposed optimization strategies is presented in Fig. 4. These plots represent the average and maximum cost associated with the parameter vector λ_i for $i = 1 \dots N_{CE}$, for different sectors s along the circular orbit. Due to the design of the cost policies, the objective is to find the minimum for $\mathcal{L}_{\text{trace}}$ and the maxima for $\mathcal{L}_{\text{tuo}}, \mathcal{L}_{\text{mtuo}}$.

The cost strategy based on the trace of the covariance matrix enables the top λ_i quantile to converge almost immediately to the optimal trajectory (blue line in Fig. 4 (a)). On the contrary, the costs based on the landmark observation inferences ($\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$) show a slower convergence rate towards the optimal value, albeit the latter is always reached without significant oscillations (Fig. 4 (b-c)). In all the performed simulations, the proposed cost strategies outperform the non controlled case (solid horizontal line in Fig. 4).

A further analysis to gain an insight of the performance takes into account the overall uncertainty reduction made possible with the proposed cost strategies. A good estimate of this uncertainty is again the trace of the covariance matrix: in Fig. 5 the trace behavior is plotted for the non controlled and the controlled cases. The trace increase in the first leg of the curve is due to the landmark acquisition and population of the (initially empty) covariance matrix during the first orbit. Since $\omega_\theta = 0$ during the *recognition* phase, this first part is identical (with the obvious differences due to noise) for all cases.

The second part of the curve is influenced by the strategy under analysis. As expected, the decay of $\text{trace}(\Sigma(t_N))$ is faster in the controlled case ($\mathcal{L}_{\text{tcm}}$, $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$) compared to the non-controlled scenario (Fig. 5 (a)). In particular, the cost $\mathcal{L}_{\text{tcm}}$ allows for the best performance in terms of uncertainty reduction, with $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ performing very similarly (Fig. 5 (c-d)).

Finally, we studied how the different costs lead to differences in landmark detection. To do this, we represented the time spent under observation as the potential map in Fig. 4, in which the color intensity represents the number of times each landmark has been measured in the same orbital portion. The scatter plot has then been interpolated and a 3D surface was computed. As expected, in the non controlled case most observations happen in the proximity of the orbit's center, Fig. 4 (a). The cost based on the trace, Fig. 4 (b), presents fairly good performances in terms of duration and distribution of the observations. Cost strategies based on $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ present a very similar shape in terms of observed landmarks: however, the cost $\mathcal{L}_{\text{mtuo}}$ allows for an even distribution of the observations. This is due to the additional term in Eq. [14], which takes also into account the number of different features: in the potential plot, this is confirmed by the more uniform gradient among the landmarks.

Overall, strategies based on $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ allow for a higher number of observed features: we present another performance indicator for the algorithm. In Fig. 7, the cumulative landmark observations are shown with the aid of bar charts: each bar represents a landmark and the height of each bar represents the number of detection during a turn.

It is interesting to note how the non-controlled case, Fig. 7 (a), performs poorly, both in terms of number of detected landmarks and distribution (some landmarks, for example, are never detected). The first control policy, based on $\mathcal{L}_{\text{tcm}}$, allows for a significant performance increase, both in terms of number and frequency of the observations (Fig. 7 (b)). Cases $\mathcal{L}_{\text{tuo}}$ and $\mathcal{L}_{\text{mtuo}}$ show again a similar structure in the observation frequency, but with the substantial difference of a much more even distribution in the case of $\mathcal{L}_{\text{mtuo}}$.

In conclusion, based on the potential and cumulative analyses, the modified $\mathcal{L}_{\text{mtuo}}$ allows for more frequent and more even observations if compared to the performance of $\mathcal{L}_{\text{tcm}}$.

V. Conclusions

We analyzed the performances of three different control strategies for solving the active self-localization problem during relative navigation in or-

bit using Cross Entropy (CE) minimization.[10]

By jointly considering the planning, control and estimation problems it is 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. Costs based on the trace of the covariance matrix and on the time under observation have been designed and tested, showing good performances in terms of convergence and mapping/tracking capabilities.

In particular, the cost $\mathcal{L}_{\text{mtuo}}$ outperformed its simpler version $\mathcal{L}_{\text{tuo}}$ by taking into account a higher landmark cardinality while maintaining a similar absolute number of observations. The cost based on the trace, $\mathcal{L}_{\text{tcm}}$, although outperforming the non-controlled case, provided slightly worse performances compared to $\mathcal{L}_{\text{mtuo}}$, both in terms of quantity of features observed and their respective frequency.

Future work will focus on the corroboration 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] J. L. Junkins, D. C. Hughes, D. C. Hughes, K. P. Wazni, K. P. Wazni, V. Pariyapong, and V. Pariyapong, "Vision-based navigation for rendezvous, docking and proximity operations," in *AAS Guidance and Control Conference*, (Breckenridge, Colorado), pp. 3–7, 7-10 February 1999.

[3] P. Singla, K. Subbarao, and J. L. Junkins, "Adaptive output feedback control for spacecraft rendezvous and docking under measurement uncertainty," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 4, pp. 892–902, 2006.

[4] J. M. Kelsey, J. Byrne, M. Cosgrove, S. Seereeram, and R. K. Mehra, "Vision-based relative pose estimation for autonomous rendezvous and docking," in *IEEE Aerospace Conference*, (Big Sky, MT, USA), p. 20, 4-11 March 2006.

[5] 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.

[6] 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.

[7] H. Curtis, *Orbital Mechanics for Engineering Students*. Butterworth-Heinemann, Burlington, MA, 2013.

[8] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, P. L'Ecuyer, *et al.*, *The Cross-Entropy Method For Optimization*, vol. 31. 2013.

[9] D. P. Kroese and R. Y. Rubinstein, "Monte carlo methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 1, pp. 48–58, 2012.

[10] 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.