

Earth-based Simultaneous Localization and Mapping for Drones in Dynamic Environments

Manuel Simas · Bruno J. Guerreiro · Pedro Batista

Received: date / Accepted: date

Abstract This paper addresses the problem of simultaneous localization, mapping, and moving object tracking (SLAMMOT) with application to unmanned aerial vehicles inside uncertain environments with both static and moving objects. Although this problem has been addressed by the community, the usual approach is to identify the moving objects and remove them from contributing to the localization and mapping algorithm. Conversely, the proposed strategy integrates the moving objects in the vehicle motions estimation, without compromising its accuracy. The proposed solution is based on the design of: i) an extended Kalman filter (EKF); ii) a modified version of the multiple hypothesis tracking method to perform data association and track moving objects; and iii) and the interacting multiple model algorithm to identify the motion models described by the environment's objects. The performance of the devised SLAMMOT filter is studied in simulation and validated with a new public experimental dataset using an RGB-D camera on-board an instrumented quadrotor with ground-truth.

Keywords Dynamic environments · Simultaneous localization, mapping, and moving objects tracking · Unmanned aerial vehicles · Kalman filter

1 INTRODUCTION

With the steady increase in the use of unmanned aerial vehicles (UAVs) for demanding applications, such as the inspection of pipelines or wind turbines, relying on remote pilots can

*This work was partially funded by the FCT projects REPLACE (PTDC/EEI-AUT/32107/2017), CAPTURE (PTDC/EEI-AUT/1732/2020) and DECENTER (PTDC/EEI-AUT/29605/2017), which include Lisboa 2020 and PIDDAC funds, and also by projects LARSYS (UIDB/50009/2020) and CTS (UIDB/00066/2020).

M. Simas and P. Batista
Institute for Systems and Robotics (ISR-Lisbon), Laboratory for Robotics and Engineering Systems (LARSYS), Portugal, and with Instituto Superior Técnico, Universidade de Lisboa, Portugal
E-mail: manuel.simas@tecnico.ulisboa.pt , pbatista@isr.tecnico.ulisboa.pt

B. J. Guerreiro (corresponding author)
Dep. of Electrical and Computer Eng. and CTS-UNINOVA, NOVA School of Science and Technology, Universidade Nova de Lisboa, and with the Institute for Systems and Robotics (ISR-Lisbon), LARSYS, Portugal.
E-mail: bj.guerreiro@fct.unl.pt

be ineffective or even impossible. Solutions to this challenge require higher degrees of autonomy for the systems on board UAVs (also known as drones), among which the use of simultaneous localization and mapping (SLAM) is by now a standard capability. This type of algorithms allows the navigation, exploration, and execution of tasks in unknown environments, where other partial methods may lack performance or reliability by themselves. For instance, techniques that rely exclusively on odometry or inertial measurement units (IMUs) have cumulative errors [1], not guaranteeing consistent results when used without other sources of information. On the other hand, relying exclusively on global positioning system (GPS) receivers may not be entirely possible for indoor environments and result in poor performance for outdoor locations where buildings or natural topology can degrade the GPS signal. The use of other sensors is common for SLAM and for localization and tracking algorithms [2, 3], where the most usual include RGB and RGB-D cameras [4–7], LiDAR [8], and also SONAR [9, 10].

Although SLAM algorithms provide a good solution for many mapping and localization problems, they often assume that the environment is static, which may be a restrictive assumption in many applications. As shown in [11], this assumption leads to inconsistent results when a SLAM algorithm is applied in a dynamic environment, since this solution does not take into consideration dynamic objects. Simultaneous localization, mapping, and moving objects tracking (SLAMMOT) algorithms are an extension of SLAM methods, in which dynamic objects are considered, providing accurate identification of all objects in the environment, which can be used to obtain more consistent and accurate estimates for the robot localization and surrounding map.

The SLAMMOT literature includes valuable and relevant implementations, such as the algorithm proposed in [12]. This solution is based on the partition of measurements in two groups, the static objects (SOs) and the moving objects (MOs), where only the former are used for the robot pose estimate. The map and objects are represented in local occupancy maps [13] and the detection of MOs is performed using a consistency-based [11] and a moving object map-based detectors. Association of MOs is performed using the multiple hypothesis tracking (MHT) algorithm [14], whereas MOs are initialized using the interacting multiple models (IMM) algorithm [15][16]. For data fusion, an extended Kalman filter (EKF) is applied, using only information regarding the SOs of the environment, whereas the loop closure is achieved by feature matching between local maps. Lin and Wang [17] subsequently purposed a stereo-based solution of the previous work. Wolf and Sukhatme approach [18] is based on the representation of two occupancy grid maps, one only containing SOs and the other for the MOs. Based on the first map, MOs are detected using the occupancy grid formulation similar to [11]. An EKF is used for the robot and landmarks estimations. A solution proposed by Vu, Burlet and Aycard [19] also represents local maps using the occupancy grid framework and the vehicle localization is performed considering the matching problem as a maximum likelihood estimation, using the vehicle's dynamics and all the previous local maps, that only account for SOs. The detection and tracking of MOs are done using an adaptive IMM algorithm with 16 motion models, coupled with the MHT method. A common trait to all these SLAMMOT solutions is that they detect and exclude all MOs from the estimation filters, thus rendering a dynamic environment into a static environment.

Scenarios where typical SLAMMOT strategies might show their handicap include environments where, excluding the moving objects, the remaining static landmarks might not be enough to provide an accurate SLAM solution. In these cases, including the moving objects in an integrated fashion can help improve the localization and mapping accuracy. In contrast with the usual strategies, the proposed solution provides a SLAMMOT filter that uses the

information of all elements available in the environment, both static and moving objects, fused with the vehicle's dynamic information. The main contribution of this paper relies on this novel filter, which is thoroughly analyzed in simulation and validated with a new public experimental dataset. The proposed algorithm relies on concepts of linear and non-linear system theory, Kalman filtering, target-measurement assignment, and target tracking, geared towards establishing conditions for convergence. The identification of SOs and MOs is achieved using the IMM algorithm, the data association of all objects in the environment and tracking of MOs is performed using a modified version of the MHT method, whereas an EKF is used for data fusion purposes. This paper builds on a preliminary version of this work presented in [20], providing an in-depth presentation of the method and simulation results, as well as the validation of the proposed method with an experimental dataset.

The remainder of this paper is organized as follows. The system dynamics and the problem statement are provided in Section 2, while Section 3 introduces the filter design, including data association, object detection and tracking, as well as loop closure methods. Simulation results are presented in Section 4 and experimental results that confirm the effectiveness of the filter are shown in Section 5. Conclusions and future work are discussed in Section 6.

2 Problem Definition

This section presents in detail the formulation and design of the filter as solution for the SLAMMOT problem, with application to an aerial vehicle in unknown dynamic environments. The full system is described and includes the dynamics of the vehicle and objects, as well as the measurement model.

2.1 System Dynamics

Consider W as the world-fixed reference frame, B the vehicle's body-fixed reference frame, ${}^W p_B(t) \in \mathbb{R}^2$ the position of B described in W , and $v_B(t) \in \mathbb{R}^2$ the linear velocity of the drone relative to W , expressed in B . Accordingly, the equation that describe the motion kinematics of the drone is given by

$${}^W \dot{p}_B(t) = v_B(t), \quad (1)$$

with

$$v_B(t) = v(t) \begin{bmatrix} \cos(\psi(t)) \\ \sin(\psi(t)) \end{bmatrix}, \quad (2)$$

where $v(t) \in \mathbb{R}$ and $\psi(t) \in \mathbb{R}$ represent, respectively, the linear velocity norm and yaw of the drone relative to W , expressed in B .

The essential condition of a dynamic environment is that includes both static and moving objects. Although the position of a SO in W is constant over time, a MO may require multiple motion models. Our formulation, defines that the motion models of both types of objects is linear, i.e., SOs describe a constant position (CP) model and MOs only describe a constant velocity (CV) model. As a result, with ${}^W p_i(t) \in \mathbb{R}^2$ representing the position of the i -th object in W , the kinematics of a static object is given by the CP model

$${}^W \dot{p}_i(t) = 0 \quad (3)$$

while a moving object considers the CV model given by

$${}^W\ddot{\mathbf{p}}_i(t) = 0. \quad (4)$$

Although the considered moving object model is the simplest possible, considering a constant linear velocity, the use of a Kalman filter will enable the proposed strategy to tackle moving objects with a slowly time-varying linear velocity vector. By considering in its definition process and measurement noises, Kalman filters allow us to work with corrupted measurements and erroneous models, which in this case enables us to use simpler dynamics and adjust the process covariance matrices to define to what extent we can accept deviations from this model. Furthermore, other models can be considered using the proposed strategy, for instance, accounting for constant linear acceleration or constant angular velocity.

The proposed filter requires the acquisition of a series of measurements over time, with respect to the surroundings and the drone's motion. In this formulation, the velocity norm and the yaw angle are considered as the inputs to the system and available as the control vector $\mathbf{u}(t) = [\mathbf{v}(t) \ \psi(t)]^T \in \mathbb{R}^2$. In practice, the heading of the vehicle can be computed using an IMU equipped with a magnetometers, usually resorting to filters dedicated to attitude estimation. On the other hand, the velocity norm can be obtained from odometry sensors in the wheels of ground vehicles, or using optical flow sensors in the case of aerial vehicles. With a slight abuse of the term, we refer to these quantities as the vehicle odometry, as these variables are considered as a known input corrupted by noise, defined as

$$\mathbf{v}_m(t) = \mathbf{v}(t) + \mathbf{n}_{\mathbf{v}_m}(t) \quad (5)$$

$$\psi_m(t) = \psi(t) + \mathbf{n}_{\psi_m}(t) \quad (6)$$

with $\mathbf{v}_m(t) \in \mathbb{R}$ and $\psi_m(t) \in \mathbb{R}$ representing, respectively, the sensors measurements of the velocity norm and yaw of the drone. The associated disturbances, $\mathbf{n}_{\mathbf{v}_m}(t) \in \mathbb{R}$ and $\mathbf{n}_{\psi_m}(t) \in \mathbb{R}$, are assumed to be zero-mean white Gaussian noise with standard deviation $\sigma_{\mathbf{v}_m}$ and σ_{ψ_m} , respectively.

The information with respect to the surroundings is acquired by measurements that describe the relative position between the position of the drone and environment's objects position, both represented in the world-fixed reference frame. Let $\mathbf{y}_i \in \mathbb{R}^2$ denote the measure with regard of the i -th object, thus, the measurement model is given by

$$\mathbf{y}_i(t) = ({}^W\mathbf{p}_i(t) - {}^W\mathbf{p}_B(t)) + \mathbf{n}_y(t), \quad (7)$$

that it is also corrupted by zero-mean white Gaussian noise, $\mathbf{n}_y(t) \in \mathbb{R}^2$, with standard deviation σ_y in each component.

In this formulation, the complete state vector can be decomposed into the vehicle and all objects information. The vehicle information only comprises the drone's position, $\mathbf{x}_V(t) := {}^W\mathbf{p}_B(t)$, while the state of the i -th object is composed by its position and velocity vectors, $\mathbf{x}_{O_i}(t) := [{}^W\mathbf{p}_i^T(t) \ {}^W\mathbf{v}_i^T(t)]^T$, where ${}^W\mathbf{v}_i(t) \in \mathbb{R}^2$ denotes the velocity of the object, described in W . Thus, with n observed objects, the complete state space vector is represented by $\mathbf{x}(t) := [\mathbf{x}_V^T(t) \ \mathbf{x}_{O_1}^T(t) \ \dots \ \mathbf{x}_{O_n}^T(t)]^T \in \mathbb{R}^{2+4n}$.

2.2 Problem Statement

This research presents the design of a SLAMMOT filter, for aerial vehicles, that acknowledges the existence of static and dynamic elements in the environment, and, in particular, incorporates the information provided by all these objects into the filter.

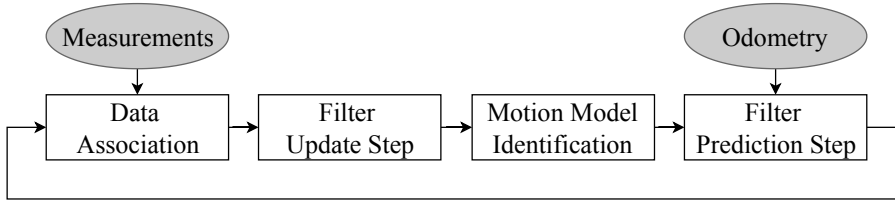


Fig. 1 Overall SLAMMOT algorithm, where the gray ovals denote data input and rectangles represent the main processes.

3 Filter Design

The design of the filter is presented in this section. The main components of the overall algorithm are presented in Fig. 1, where at each instant for which there are object observations, the first step of the algorithm is the data association. In this step, the MHT keeps track of several possibilities of objects of the environment that can be associated with objects in the state, keeping a collection of hypothesis in the form of measurement matrices, that can be further justified or falsified with future observations. Given the most probable measurement matrix computed by the MHT, the update step of the EKF can be performed with the information of both static and moving objects. To identify which dynamic model best fits each of the objects in the state of the filter, the IMM algorithm computes the EKF update and prediction steps for each motion model and for each state object, with their respective probabilities. Given the most probable model for each object, the last step is to compute the prediction step of the main EKF filter, to be used in the next iteration of the filter. In the remaining of this section, each of these steps will be further detailed.

3.1 Data Association

The target-measurement assignment problem consists of correctly associating, over time, each available measurement with the corresponding object. This data association process is required to be robust and consistent, to ensure that the uncertainty and error of the estimates decreases. Additionally, in SLAMMOT formulations, it is essential for convergence of the objects' motion model identification and tracking of MOs. Moreover, the effectiveness of data association allows loop closure to occur.

The proposed algorithm uses a modified version of the MHT method to track both SOs and MOs. The MHT algorithm works in a two-step process. The first step is related to the hypotheses generation, by mapping the elements of the objects observations vector at each instant $z_k := [z_1^T(k) \dots z_{n_z}^T(k)]^T$ with the existing objects in the state vector to define the measurement vector as $y_k := H_k x_k$. An object is only associated with a measurement if the latter lies within the object's validation region and, consequently, the hypotheses that are generated do not contain associations between targets and measurements that are not inside targets' validation region. This validation region depends on the objects' state and respective uncertainty, and also on the assumed observation noise. This region is known as the η -sigma validation region, since its size is variable and controlled by the empirical rule [21]. Then, in the second step the probability of each hypothesis is computed and this calculation depends on the type of the sensor that is available. The first type of sensor can provide position measurements and information regarding the number of targets within that

set of measurements, while the second does not provide the number of objects. In the proposed formulation we assume that each measurement is related to one target and, for this reason, the number of objects at each scan is available. In addition, it is assumed that there is no false measurements. Consequently, the probability calculation depends on the following parameters (further details in [14]):

- Probability of detection (P_D);
- Density of new targets (β_{NT});
- Density of false targets (β_{FT});
- Number of measurements associated with prior targets (N_{DT});
- Number of measurements associated with false targets (N_{FT});
- Number of measurements associated with new targets (N_{NT});
- Number of previously known targets (N_{TGT});
- Likelihood of a measurement be associated with a target.

According to the previous assumptions, in the presented formulation we consider $P_D = 1$, $\beta_{FT} = 0$ and $N_{FT} = 0$. The density of new targets is given in function of the quantity of new area that is sensed by the sensor, yielding

$$\beta_{NT} = 0.02 + \frac{A_N}{A_N + A_O} \quad (8)$$

with A_N and A_O denoting, respectively, the quantity of new and old area covered by the sensor. To account for possible new targets when the vehicle is at rest, and a null value for the density of new targets, a constant value of 0.02 is considered. The selection of this value is empirical and tailored for the envisioned application scenarios. The likelihood calculation depends on the validation area, thus, directly depends on the uncertainty of each known object. Since the validation area of each object may comprise high values of uncertainty, to avoid miss association between different types of objects, it was additionally imposed that an existing object is only assigned to a given measurement if the probability of this association is higher than a given threshold. This is particular relevant for MOs, which in result of their movement may have larger validation regions within which is possible to have low probability associations with SOs.

For each step, the MHT algorithm computes all possible hypotheses. For this reason, MHT is considered to be an optimal method [22], at the cost of exponential complexity. Consequently, it is necessary to apply pruning techniques, resulting in suboptimal performance. Nevertheless, the MHT method is more robust than other data association approaches, such as the probabilistic data association filter (PDAF) [23], the joint probabilistic data association filter (JPDAF) [24], or the global nearest neighbor (GNN) [25], as shown in [26], [27] and [28].

3.2 Moving Objects Detection and Tracking

The correct identification of SOs and MOs is essential to achieve robust data association and, consequently, to accurately estimate the motion of both the surrounding objects and the vehicle. In the proposed solution, the IMM algorithm is used for the motion models identification. When compared with other methods, as shown by Blom and Bar-Shalom [16] and Pitre et al. [29], the IMM algorithm outperforms or have a similar performance even when compared with more complex algorithms.

This algorithm uses r filters to estimate the state and covariance of r motion models, for each object. When a measurement is available, the algorithm uses the r predictions and mixes them using a transition matrix $\mathbf{T} \in \mathbb{R}^{r \times r}$ and the current models' probabilities, to obtain r mixed predictions. Since the transition matrix defines the probability of an object changing to another model, these new predictions represent the weighted sum of the prior predictions, considering each model probability. Subsequently, each prediction is updated using the current measurement, yielding r updated estimates. Then, the likelihood of each model is calculated using the innovation covariance matrix, obtained from the update step, and the current measurement. The output is the probability of each model, given that measurement. The current formulation only considers the CP and CV models. As a result, the IMM uses two filters for each object and $\mathbf{T} \in \mathbb{R}^{2 \times 2}$. The original IMM algorithm, after computing the likelihood of each model, performs state mixing, using the model's probabilities and the updated estimates of each model. In this implementation, this step is not performed and only the probability of each model is used to select the most probable model to be used in the main filter.

The tracking of MOs is accomplished by both IMM and MHT algorithms, i.e., the IMM algorithm defines the motion model of each object, while the MHT algorithm handles the association and track maintenance of these objects.

3.3 Loop Closure

The main focus of this work is the formulation and validation of a filter implementation that addresses the existence of dynamic objects in the environment, resulting in consistent results that enable the use of any loop closure technique. In the tested scenarios, the combination of the MHT and IMM algorithms provide enough accuracy in the estimates and their uncertainty, enabling the filter to detect and associate an old landmark with a new measurement, after one loop. As such, the use of advanced algorithms for loop closure was not considered. Nonetheless, the consistency to enable loop closure without advanced algorithms highly depends on the complexity and size of the environment, for which proper techniques should be implemented when dealing with large and complex scenarios, such as those presented by [30] or [31].

3.4 Extended Kalman Filter

A discrete-time EKF was designed for data fusion purposes. Every time that a measurement is available, the MHT algorithm performs data association, the motion model of the objects is defined by the IMM algorithm, and finally the filter computes the new estimate. Otherwise, the system is propagated in open loop.

3.4.1 Discrete dynamics

Consider $\mathbf{x}_k := \mathbf{x}(t_k)$, $\mathbf{y}_k := [\mathbf{y}_1^T(t_k) \dots \mathbf{y}_{n_y}^T(t_k)]^T$ and $\mathbf{u}_k := \mathbf{u}(t_k)$, with $t_k = t_0 + kT_s$, $k \in \mathbb{N}_0$, t_0 as the initial time and n_y as the number of available measurements. Hence, the forward Euler discretization of the system dynamics can be defined as

$$\begin{cases} \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_{k+1}) + \mathbf{w}_{k+1} \\ \mathbf{y}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{n}_{k+1} \end{cases}, \quad (9)$$

where f represents the state transition function, h is the measurement model function, with $w_k \in \mathbb{R}^{2+4n}$ and $n_k \in \mathbb{R}^{2n_y}$ denoting the associated process and measurement noise, which are assumed to be zero-mean white Gaussian disturbances. The state transition function is obtained using the vehicle and object's motion kinematics equations, respectively (1) and (3) and/or (4), whereas the observation function is obtained using the measurement equation, (7).

3.4.2 Prediction Step

During the prediction step the complete state vector is used, and the drone's position is predicted using the state transition function and the input vector $u(t)$. The objects' state vectors are only propagated according to the motion model that each object describes. Thus, the prediction equations are

$$\begin{cases} \hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_{k+1}) \\ \Sigma_{k+1|k} = \mathbf{F}_{k+1} \Sigma_{k|k} \mathbf{F}_{k+1}^T + \Xi_{k+1} \end{cases}, \quad (10)$$

where $\hat{x}_{k|k}$ and $\Sigma_{k|k}$ denote, respectively, the predicted state vector and covariance matrix.

The process noise covariance matrix is defined as $\Xi_k = \text{diag}(\Xi_{V_k}, \Xi_{O_k})$ where $\Xi_{V_k} \in \mathbb{R}^{2 \times 2}$ and $\Xi_{O_k} \in \mathbb{R}^{4n \times 4n}$ are the process noise covariance matrices of the vehicle and objects' dynamics, respectively. The former, is defined as $\Xi_{V_k} = \Theta_k \Lambda \Theta_k^T$, where $\Lambda = \text{diag}(\sigma_v^2, \sigma_\psi^2)$, with σ_v and σ_ψ respectively denoting the standard deviation of the process noise concerning the velocity norm and yaw of the drone. The respective Jacobian matrix, $\Theta_k = \frac{\partial f}{\partial u} \Big|_{u_k}$, is given as

$$\Theta_k = T_s \begin{bmatrix} \cos(\psi_k) & -v_{k+1} \sin(\psi_k) \\ \sin(\psi_k) & v_{k+1} \cos(\psi_k) \end{bmatrix}, \quad (11)$$

with $\psi_k = \psi_m(t_k)$ and $v_k = v_m(t_k)$. On the other hand, the objects' process noise matrix is defined as $\Xi_O = \text{diag}(\Xi_{O1}, \dots, \Xi_{On})$, with Ξ_{On} representing the process noise of each object. This matrix changes according to the motion model described by each object. As a result, for those describing a CP model, the matrix is given as $\Xi_{On} = T_s \text{diag}(\sigma_{p_{so}}^2 \mathbf{I}_2, \sigma_{v_{so}}^2 \mathbf{I}_2)$, whereas if the object describes a CV model, it is given as $\Xi_{On} = T_s \text{diag}(\sigma_{p_{mo}}^2 \mathbf{I}_2, \sigma_{v_{mo}}^2 \mathbf{I}_2)$, with $\sigma_{p_{so}}$, $\sigma_{v_{so}}$, $\sigma_{p_{mo}}$, and $\sigma_{v_{mo}}$, denoting the process noise standard deviations of the position and velocity, respectively, of SOs and MOs.

Finally, the complete transition matrix, defined as $\mathbf{F}_k = \text{diag}(\mathbf{F}_V, \mathbf{F}_O)$, encompasses the vehicle transition matrix, $\mathbf{F}_V = \mathbf{I}_2$, and the full transition matrix of all objects, given by $\mathbf{F}_O = \text{diag}(\mathbf{F}_{O1}, \dots, \mathbf{F}_{On})$, with \mathbf{F}_{On} representing the transition matrix of each object. The last-mentioned matrix, for SOs, is given by

$$\mathbf{F}_{On} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (12)$$

and for MOs, it is given by

$$\mathbf{F}_{On} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (13)$$

The initial state, $\hat{\mathbf{x}}_{0|0}$, and respective covariance matrix, $\Sigma_{0|0}$, are chosen considering conservatively the specific experimental setup and the possible uncertainty in the state initial values. This can also be done with a procedure before initializing the filter to obtain a statistical characterization of the initial state and its uncertainty. Regarding the process noise covariance matrix Ξ_k , it can be computed using the above Jacobian matrix, Θ_k , together with the standard deviations σ_v , σ_ψ , and, depending on the type of object, $\sigma_{p_{so}}$, $\sigma_{v_{so}}$, $\sigma_{p_{mo}}$, and $\sigma_{v_{mo}}$. The latter are chosen to account for model uncertainties, allowing the adjustment of possible deviations from the nominal model.

3.4.3 Update Step

Every time that a set of measurements is available, the measurements vector y_k is updated. If the measurements are with respect to new objects the state vector is augmented, otherwise only the measurements vector is changed. Then, the updated estimate can be computed using

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1}\tilde{y}_{k+1} \\ \Sigma_{k+1|k+1} = (\mathbf{I} - K_{k+1}H_{k+1})\Sigma_{k+1|k} \end{cases}, \quad (14)$$

where $\hat{\mathbf{x}}_{k+1|k+1}$ and $\Sigma_{k+1|k+1}$ denote, respectively, the updated state and covariance matrix. The measurement noise covariance matrix, Γ_k , can be obtained from the sensors' data-sheet information and, if necessary, from the statistical characterization of the computations performed to obtain the measurements y_k .

The measurement information is incorporated in this step using the measurement residuals, \tilde{y}_{k+1} , their covariance, S_{k+1} , and the Kalman gain, K_{k+1} , defined as

$$\begin{cases} \tilde{y}_{k+1} = y_{k+1} - h(\hat{\mathbf{x}}_{k+1|k}) \\ S_{k+1} = H_{k+1}\Sigma_{k+1|k}H_{k+1}^T + \Gamma_{k+1} \\ K_{k+1} = \Sigma_{k+1|k}H_{k+1}^TS_{k+1}^{-1} \end{cases}, \quad (15)$$

where $\Gamma_{k+1} \in \mathbb{R}^{2n_y \times 2n_y}$ expresses the measurement noise covariance. The measurement matrix $H_k \in \mathbb{R}^{2n_y \times 2+4n}$ is defined as

$$H_k = \frac{\partial h}{\partial \mathbf{x}} = \begin{bmatrix} -\mathbf{I}_2 & \mathbf{I}_2 & \dots & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \vdots & \vdots & \ddots & \dots & \dots \\ -\mathbf{I}_2 & \mathbf{0}_{2 \times 2} & \dots & \mathbf{I}_2 & \mathbf{0}_{2 \times 2} \end{bmatrix} \quad (16)$$

when a total of n objects have been observed and the respective measurements are all available. When a previously seen object becomes unavailable, the observation matrix is adjusted, with the respective row of the object on the matrix being removed.

At this point in the filtering algorithm, the measurement vector and matrix structure is already defined through the combination of the MHT and IMM algorithms, that track several possibilities for data association and multiple kinematic models for the state objects. Keeping information for each possible object type and propagating the respective process and measurement uncertainties, we can progressively falsify the least probable hypotheses for associating SOs with MOs and vice-versa.

3.5 Complexity and Scalability Discussion

The complexity of the overall algorithm is a determining factor for its scalability. Nonetheless, there are significant improvements both in terms of linear algebra computations and in dealing with parallelization and sparse computations that can be used to implement filtering techniques based on Kalman filters [32]. Consider the algorithm variables introduced above, in particular, the number of objects in the state vector, n , the number of objects in the measurement vector, n_y , and also the number of motion models for each object, $r = 2$.

Without algebraic optimization tools for matrix multiplication, the EKF implementation has a complexity of $\mathcal{O}(n^2 n_y + n n_y^2 + n_y^3)$, which if we consider the sensor limitations that impose an upper bound on the number of detected objects, we can assume the complexity is simply $\mathcal{O}(n^2)$, as detailed in [33]. This fact, by itself, represents a major limitation of most consistent implementations of SLAM algorithms, being a common issue even in implementations that use other filtering techniques. Regarding the added complexity of using the MHT algorithm, in general, given p existing hypothesis (or branches) in the algorithm for a given iteration, the complexity of computing the new set of hypothesis has the complexity $\mathcal{O}((pn)^2)$. Nevertheless, it has been argued that, with adequate pruning of branches of the algorithm to limit the number of hypothesis, the resulting complexity can be considered to be $\mathcal{O}(n^2)$ or lower, as discussed in [34,35]. On the other hand, the use of the IMM algorithm to identify the adequate dynamic model for each state object adds r filters for each object in the state, which represents a complexity of $\mathcal{O}(rn^2)$.

In summary, considering a constant value for the number of motion models, this would result in a complexity of $\mathcal{O}(n^2)$ for the overall algorithm. As such, the scalability of the proposed strategy can be ensured by exploring concepts in the literature to reduce the complexity of SLAM algorithms, such as using local maps and supervisory algorithms to joint them in a consistent way [33], together with efficient libraries for algebraic operations, parallelization, and sparsity handling.

4 Simulation Results

On this section the simulation details are described and the respective results are presented. These results show the effectiveness and consistency of the formulated solution. All simulations have been performed in MATLAB.

4.1 Environment

The simulated environment consists of 34 SOs, representing corners along the vehicle trajectory, and 4 MOs with different constant velocities and orientations. The total simulation time is 165 seconds, and the drone is considered to be static during 50 seconds, after which it starts to move at an average speed of 0.5 m/s. This stationary period is a standard procedure and is usually used for calibration of sensors and system initialization. The vehicle performs a circular path, detecting a total of 32 SOs, with all MOs being detected on the first lap. It is considered a sampling frequency of 100 Hz for the odometry sensors, while the sensor that provides measurements of the surroundings works at 20 Hz. The drone's trajectory and all objects are represented in Fig. 2. The simulated system disturbances are as follow, $\sigma_{v_m} = 0.15$ m/s, $\sigma_{\psi_m} = 3$ deg, and $\sigma_y = 2.5$ cm.

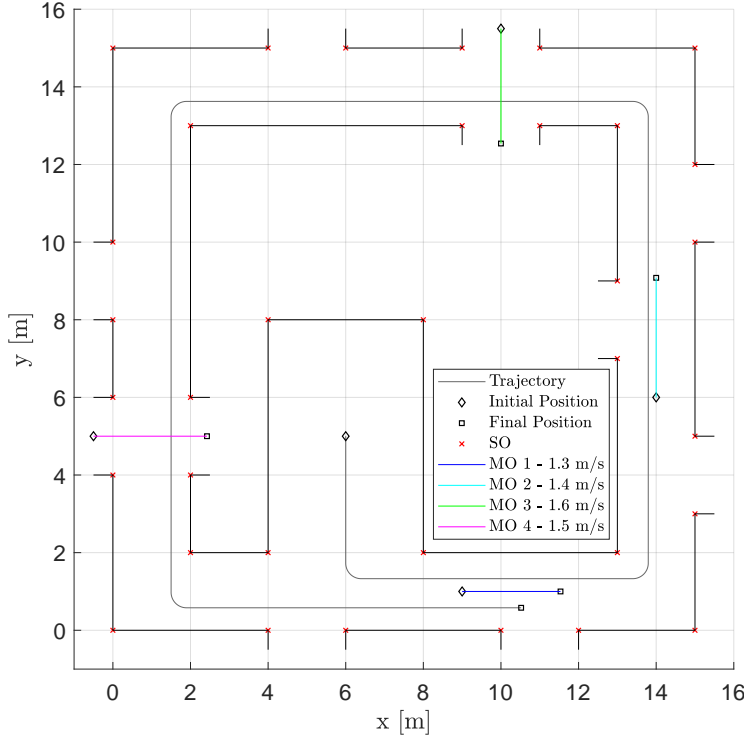


Fig. 2 Map of the simulated environment with the real drone trajectory and object positions.

4.2 Filter Parameters

The filter parameters include the MHT, IMM and EKF parameters. All these parameters have been defined empirically, taking into account the influence of each one in the filter estimations. The validation region of the MHT is assumed to be 3-sigma. Regarding the IMM algorithm, when a new object is observed, each model is initialized with the same probability, and the transition matrix is given as

$$\mathbf{T} = \begin{bmatrix} 0.75 & 0.25 \\ 0.01 & 0.99 \end{bmatrix} \quad (17)$$

which means that a SO have 75% of chance of remaining SO and 25% of chance to change to MO, while a MO have 99% of chance of remaining MO and 1% of chance to change to SO. The selection of these values is empirical, considering this particular dataset, noting that higher values in the diagonal imply more resistance to change between the two types of objects. These values translate the compromise between swiftly detecting transitions between types of motion and filtering erroneous changes due to measurement noise.

For data fusion, the drone's process noise standard deviation is assumed to be $\sigma_v = 0.3$ m/s and $\sigma_\psi = 3$ deg, while for SOs it is given by $\sigma_{p_{so}} = 2.5$ mm and $\sigma_{v_{so}} = 0$ m/s, and for MOs is considered to be $\sigma_{p_{mo}} = 1.5$ m and $\sigma_{v_{mo}} = 4$ m/s. The measurement noise covariance matrix is given as $\Gamma_{k+1} = 2\sigma_y^2 \mathbf{I}_{2n}$. When an object is observed for the first time, the uncertainty, in each direction, regarding its position and velocity is initialized with σ_y^2 . In

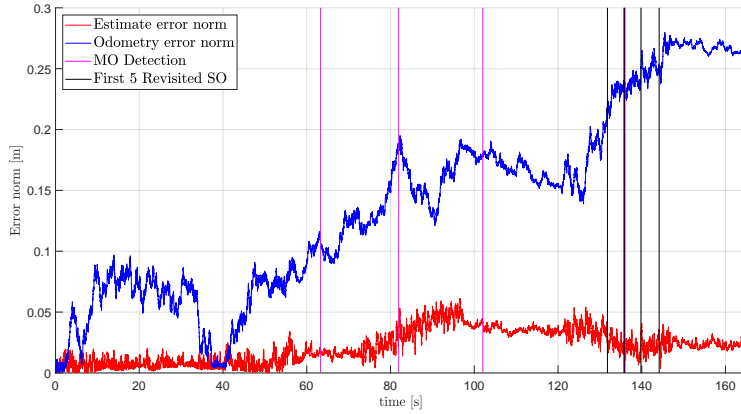


Fig. 3 Estimated and odometry-only error norm.

order to avoid an exponential growth of the number of hypotheses generated by the MHT, at each step, only the two hypotheses with the highest probability are kept.

4.3 Results

When analyzing the performance of the filter, it is important to observe a decrease on the error norm when compared with using just the odometry, as shown in Fig. 3. It is possible to notice that the error norm of the proposed filter is lower, in particular, while the drone is stationary and after 130 seconds of simulation, where multiple and consecutive loop closure events start to happen.

In Fig. 4 and Fig. 5 the standard deviation and estimation error of the nine revisited SOs is shown, respectively. When the objects are detected for the first time their uncertainty is initialized, for instance, around $t = 60$ s, but soon decreases while measurements regarding these objects are available. The uncertainty increases when the objects are not being observed, which is expected since there is process noise associated with the SO position. As soon as these objects are revisited the algorithm performs a loop closure, and, consequently, the uncertainty regarding these landmarks substantial decreases. Nonetheless, the estimation error for some objects may increase, as these landmarks have associated uncertainty, which induces the filter to correct these estimations using an estimate of the drone's position with more uncertainty than before. Nonetheless, the estimation error norm is always less than 3 cm after the initial convergence. The number and type of objects observed over time is shown in Fig. 6. From $t = 140$ s, multiple previously SOs become once again visible, and are correctly associated, since the number of SOs does not increase after that time. The total number of estimated SOs and MOs compares well with the true number of simulated objects.

The final map with all the estimates of the trajectories and final positions is shown in Fig. 8, whereas the estimated velocities of each MO are compared with the true values in Fig. 7. These results compare well to the true values, with an effective estimation of the drone and objects' states. It is worth mention that the filter correctly identifies the MO in less than 0.5 seconds and the corresponding velocity estimate converges to the true value in less than 2 seconds.

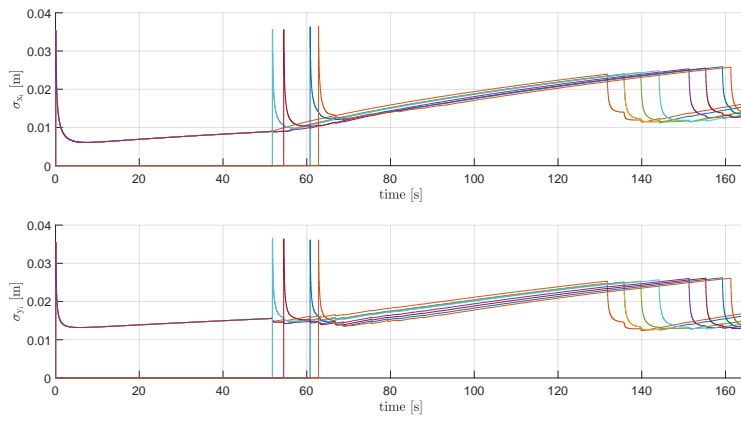


Fig. 4 Standard deviation of the nine revisited SOs positions, for complete time

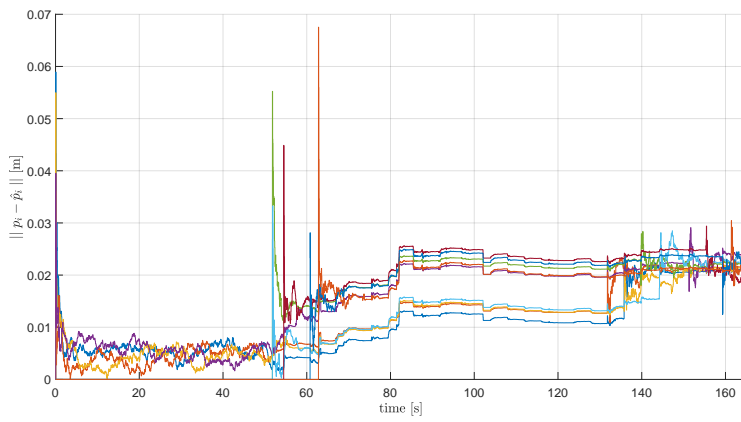


Fig. 5 Error norm of the nine revisited SOs positions, for complete time

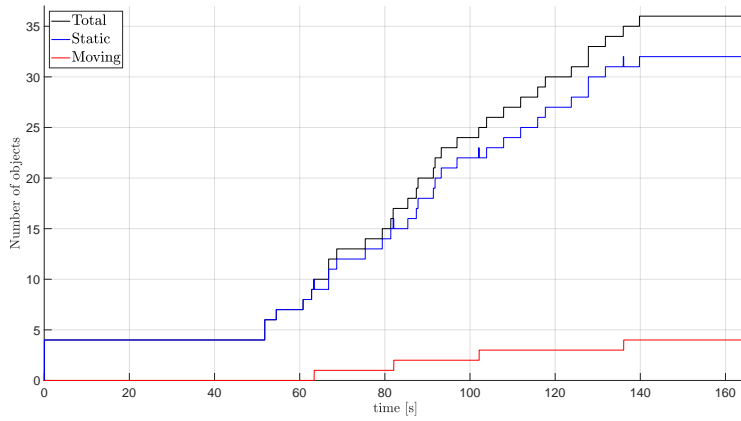


Fig. 6 Evolution of the number of different objects.

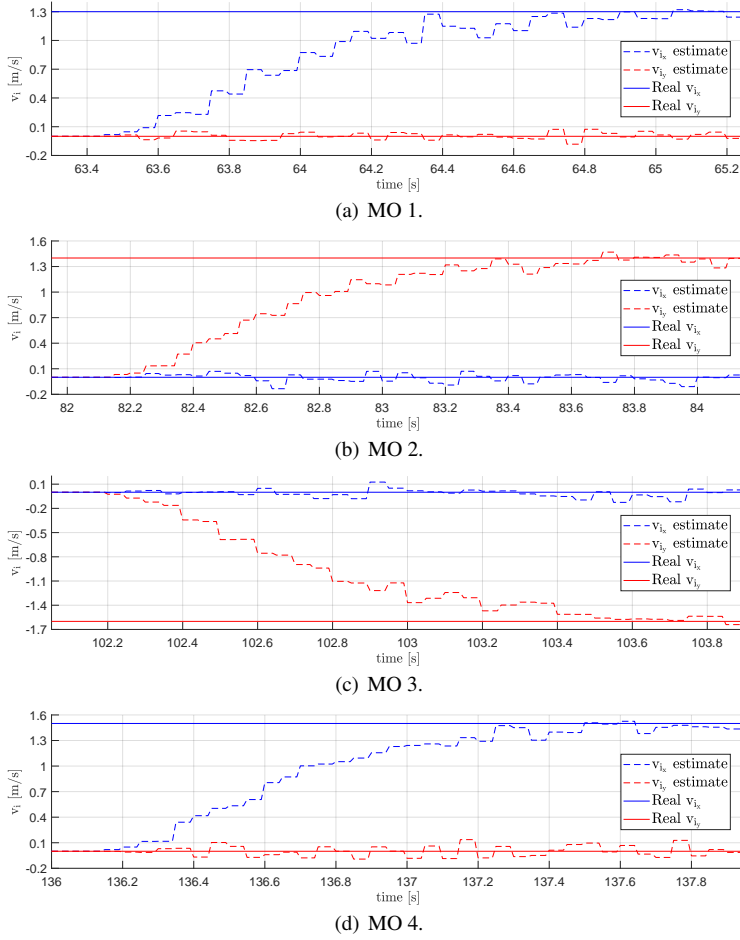


Fig. 7 Comparison of the true and estimated linear velocity of MOs.

These simulation results validate the performance of the algorithm since it is shown that the filter estimates have a lower error when compared with the odometry estimations standalone. Moreover the velocity estimates of MOs are accurately calculated and quickly converge to values similar to the real ones.

5 Experimental Results

This section describes the experimental results for the presented solution. The experimental work was developed at Institute for Systems and Robotics using an *Intel Aero RTF Drone*, equipped with a *Intel Atom* processor board and an *Intel RealSense R200 Camera*, working at 60 Hz. This dataset is made available for the community¹.

¹ REPLACE dataset 1 for SLAMMOT is available at: <https://github.com/brunojnogueiro/replace-dataset1>.

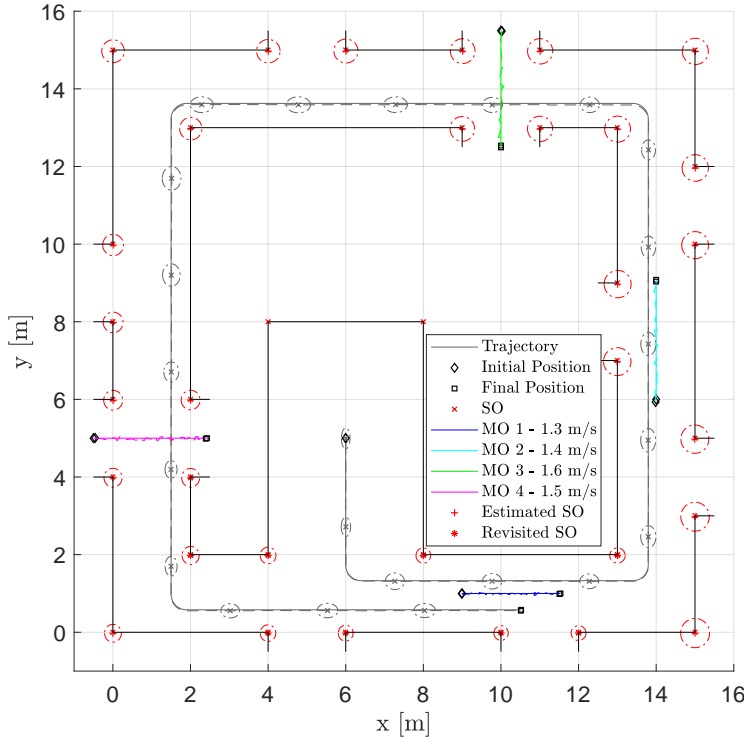


Fig. 8 Simulated environment with true and estimated trajectories and objects, where dashed lines represent estimated trajectories and ovals represent the uncertainty multiplied by a factor of 200.

5.1 Environment

The experiment was performed in a controlled room equipped with an OptiTrack motion capture system, working at 240 Hz. The quadrotor was hand-driven to perform a circular path, for about 70 seconds. The motion information of the drone, velocity norm and yaw, were computed using the positions provided by the OptiTrack system. Then, normal white Gaussian noise with zero-mean was added to these variables, to simulate the sensors' disturbance. Throughout the room some artificial markers were placed to reproduce objects and to facilitate the consecutive detection of these objects, as shown in Fig. 9(a). During this trajectory, the drone captures RGB-D images of the environment, detecting 8 SOs and a dynamic object that is moving and that stops moving at a certain point, becoming a SO.

The measurements regarding the surrounding objects were obtained using the RGB-D images provided by the *Intel RealSense R200 Camera* equipped in the drone. The full resolution images were cropped into a region of interest that avoided areas such as the ceiling and windows, that had no markers, as shown in Fig. 9(b). Then, this area was subdivided into 4 regions of interest, Fig. 9(c), to avoid having two measurements regarding the same marker. From each of these regions the strongest feature was obtained using the KAZE [36] algorithm, Fig. 9(d). Then, the features were matched with the corresponding 3-D point on the depth image. In addition, this set of 3-D points that are represented on the body-fixed frame are changed to the world-fixed frame, applying a rotation based on the drone's attitude.

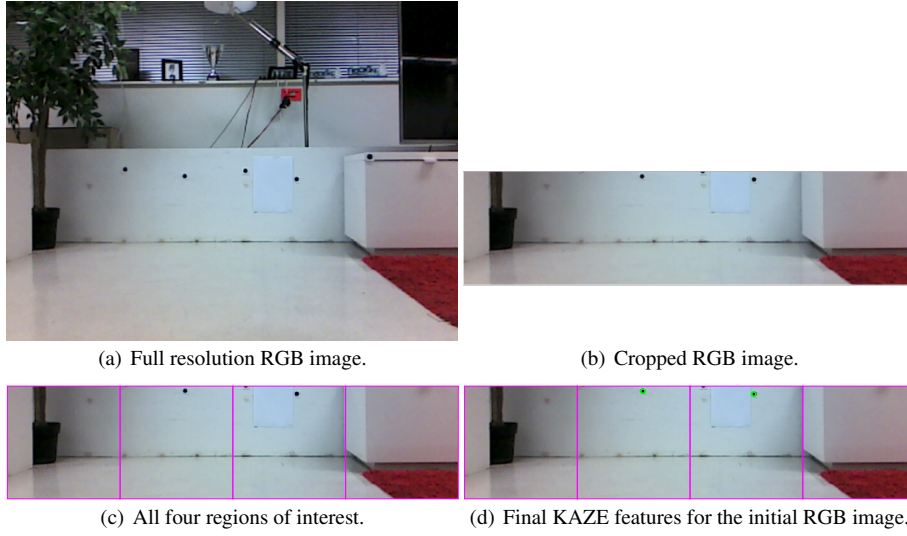


Fig. 9 Acquisition process of the experimental surrounding measurements.

The system disturbances are $\sigma_{v_m} = 0.10$ m/s and $\sigma_{\psi_m} = 10$ deg. No noise was added to the measurements concerning the surroundings since the data processing procedures already had associate disturbances.

5.2 Filter Parameters

The same filter parameters were empirically tuned for the experimental data. The validation region of the MHT is assumed to be 3-sigma. Regarding the IMM algorithm, when a new object is observed, each model is initialized with the same probability, and the transition matrix is given as

$$\mathbf{T} = \begin{bmatrix} 0.95 & 0.05 \\ 0.03 & 0.97 \end{bmatrix} \quad (18)$$

which means that a SO have 95% of chance of remaining SO and 0.05% of chance to change to MO, while a MO have 97% of chance of remaining MO and 3% of chance to change to a SO. As in the simulation results, these values are empirically selected considering this experimental dataset. In particular, when compared with the transition matrix used in the simulation results, this matrix is more close to a symmetric matrix, with a much higher probability of a SO remaining with that same dynamic than in the previous results.

For data fusion, the measurement noise is assumed to be $\sigma_y = 3.5$ cm, while the drone's process noise standard deviation is assumed to be $\sigma_v = 0.2$ m/s and $\sigma_\psi = 20$ deg/s, while for SOs it is given by $\sigma_{p_{so}} = 2$ cm and $\sigma_{v_{so}} = 0$ m/s, and for MOs is considered to be $\sigma_{p_{mo}} = 9$ cm and $\sigma_{v_{mo}} = 18$ cm/s. The measurement noise covariance matrix is given as $\Gamma_{k+1} = \sigma_y^2 \mathbf{I}_{2n}$. When a object is observed for the first time, the uncertainty, in each direction, regarding its position and velocity is initialized with $3\sigma_y^2$ and $100\sigma_y^2$, respectively. Again, for the same reason as mentioned before, at each step, the MHT only keeps the two hypotheses with the highest probability.

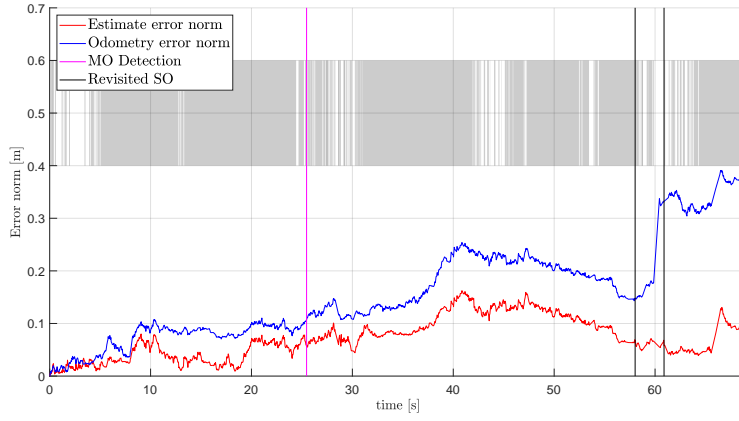


Fig. 10 Estimated and odometry-only error norm, where gray areas represent time with no measurements available.

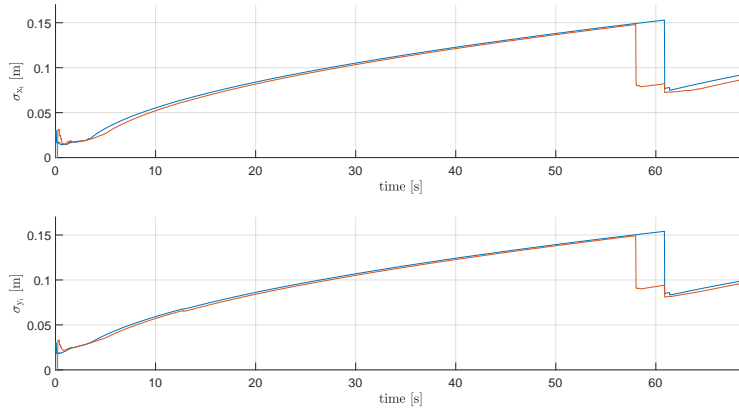


Fig. 11 Standard deviation of the two revisited SOs positions.

5.3 Results

The estimation and odometry error norm is shown in Fig. 10, where the estimation error is lower than the odometry-only error, confirming the simulation results. Around $t = 58$ s the drone starts to revisit a previous location and the algorithm successfully closes the loop, identifying the revisited landmark. These correct associations of previously seen landmarks contribute to a decrease in the uncertainty of these old landmarks, as shown in Fig. 11.

The evolution of the number and type of objects is shown in Fig. 12, from which it is possible to note that the algorithm effectively identifies the MOs and adjusts its model to CP when it stops moving. Figure 13 compares the vehicle state estimates with the ground truth obtained via OpitTrack, whereas the performance of the algorithm is confirmed since the estimates are very close to the true trajectory. Notice that around $t = 43$ s the uncertainty of the estimation decreases since new measurements are available, after a long period without any measurements. Moreover, when loop closure occurs the uncertainty decreases, but it increases right after given the fact that there are no more measurements.

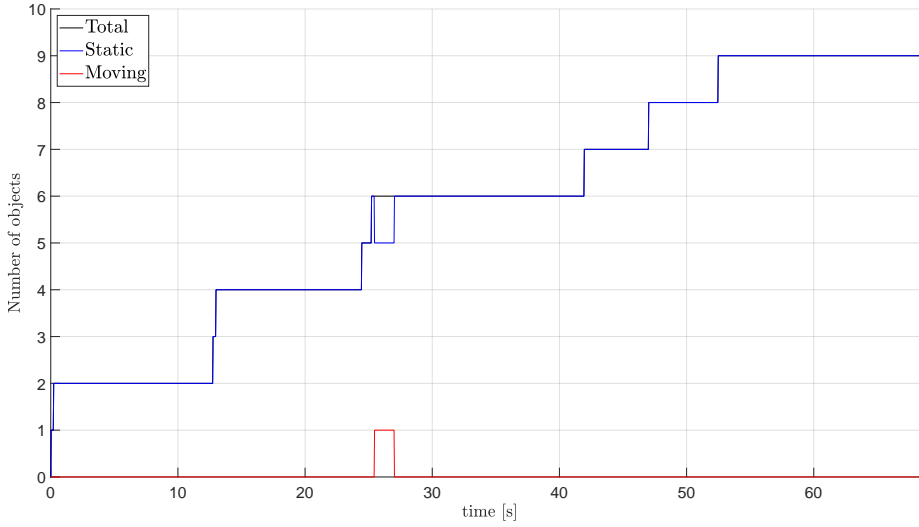


Fig. 12 Evolution of the number of different objects.

The estimated trajectory and objects, with the corresponding uncertainties, are represented in Fig. 14, along with the true trajectory and the odometry-only estimate. Results show that the revisited objects have lower uncertainty when compared to the other objects and that the estimate outperforms the odometry, especially by the end of the experiment.

These experimental results confirm complement the above simulation results, validating the proposed filter and showing that it clearly outperforms the odometry-only alternative. These results also show that the filter is robust, not only being able to correctly identify MOs, but also to detect that the MOs have stopped.

6 Conclusions

The aim of this research was to devise an SLAMMOT filter which can be applied for aerial vehicles in uncertain and dynamic environments. The main contribution of this work was to address the existence of both static and moving objects in the environment and evaluate the performance of an EKF-based SLAMMOT when using the information of all these elements for estimation purposes. The consistency and performance of the algorithm was validated with simulation results and later confirmed with experimental results.

Future work includes the incorporation of more motion models into the IMM algorithm, for example, a constant acceleration (CA) model. Additionally, the improvement of the pruning methods applied to the MHT algorithm would be very beneficial to reduce the computational cost of the data association without the loss of possibly relevant hypotheses, especially when applied in more complex environments. This can be accomplished by implementing the modifications suggested in [37] or by the implementation of a hybrid approach for data association, where SOs are associated using a less computational expensive algorithm, such as the GNN or the joint compatibility branch and bound [38] algorithms.

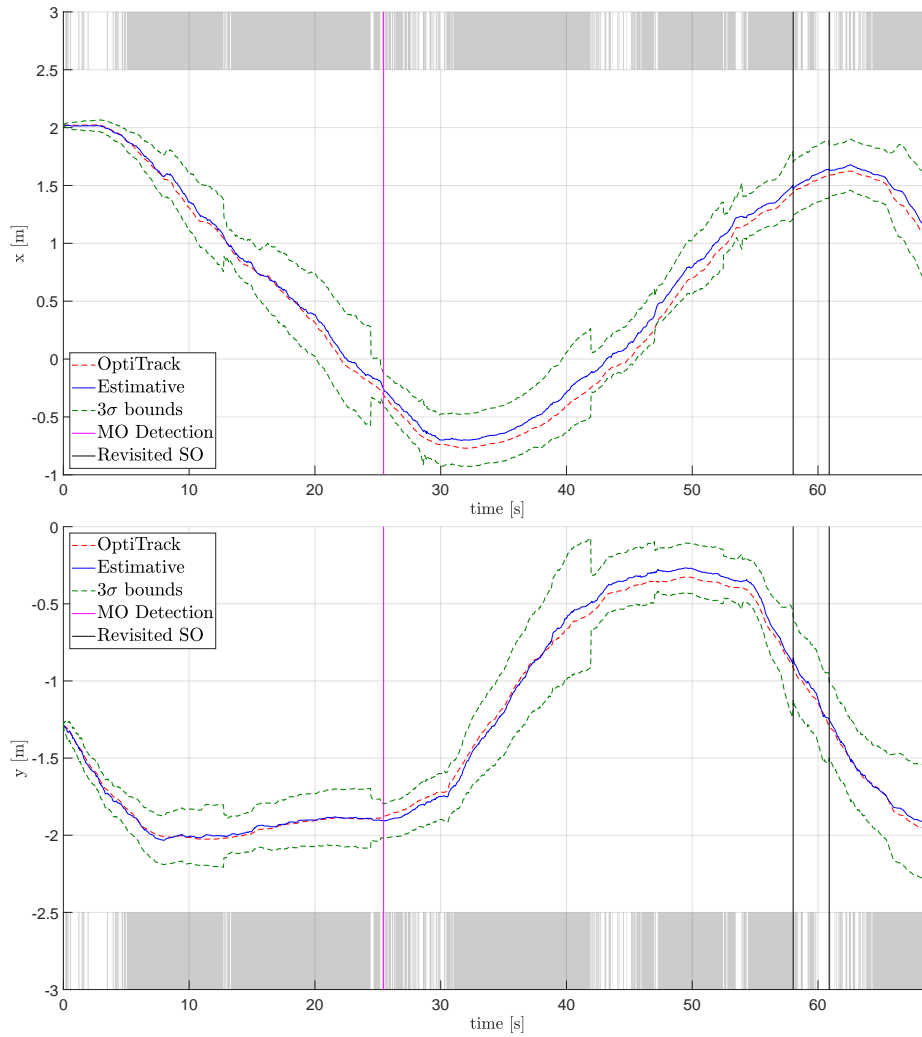


Fig. 13 Time evolution of the true and estimated vehicle's position, with 3σ bounds, where gray areas represent time with no measurements available.

References

1. M. Ben-Ari and F. Mondada, *Robotic Motion and Odometry*. Springer International Publishing, 2018, pp. 63–93.
2. D. Gala and L. Sun, “Moving sound source localization and tracking using a self rotating bi-microphone array,” in *Dynamic Systems and Control Conference*, vol. 59148. American Society of Mechanical Engineers, 2019, p. V001T09A002.
3. J. Thomas, J. Welde, G. Loianno, K. Daniilidis, and V. Kumar, “Autonomous flight for detection, localization, and tracking of moving targets with a small quadrotor,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1762–1769, 2017.
4. J. Montiel, R. Mur-Arta, and J. D. Tardós, “Orb-slam: A versatile and accurate monocular,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.
5. T. Li, S. Hailes, S. Julier, and M. Liu, “Uav-based slam and 3d reconstruction system,” in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 2496–2501.

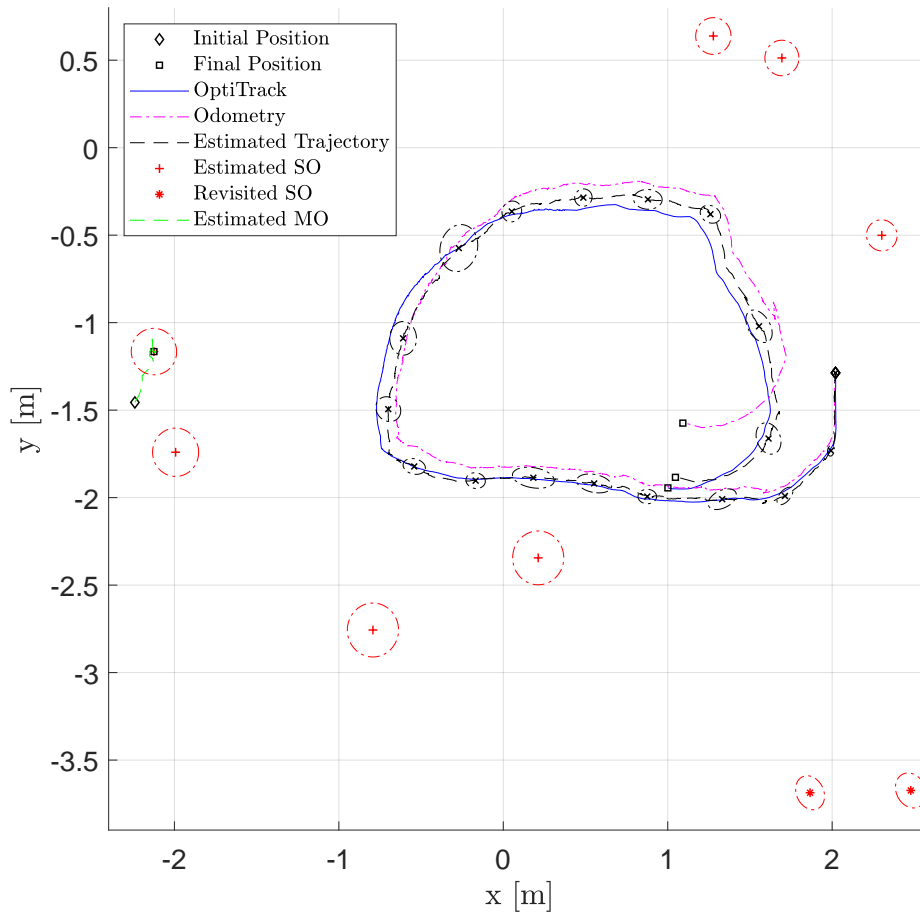


Fig. 14 True, odometry-only, and estimated trajectories, along with the objects estimates, with uncertainty ellipses.

6. P. Lourenço, B. J. Guerreiro, P. Batista, P. Oliveira, and C. Silvestre, "Preliminary Results on Globally Asymptotically Stable Simultaneous Localization and Mapping in 3-D," in *American Control Conference (ACC)*, June 2013, pp. 3087–3092.
7. —, "Uncertainty Characterization of the Orthogonal Procrustes Problem with Arbitrary Covariance Matrices," *Pattern Recognition*, vol. 61, pp. 210–220, 2017.
8. B. J. Guerreiro, P. Batista, C. Silvestre, and P. Oliveira, "Sensor-based Simultaneous Localization and Mapping – Part I: GAS Robocentric Filter," in *American Control Conference (ACC)*, 2012, June 2012, pp. 6352–6357.
9. D. Gala, N. Lindsay, and L. Sun, "Realtime active sound source localization for unmanned ground robots using a self-rotational bi-microphone array," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3–4, pp. 935–954, 2019.
10. —, "Multi-sound-source localization for small autonomous unmanned vehicles with a self-rotating bi-microphone array," *arXiv preprint arXiv:1804.05111*, 2018.
11. C.-C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 2918–2924.
12. C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007.

13. A. Elfes, "Occupancy grids: A probabilistic framework for robot perception and navigation," Ph.D. dissertation, Carnegie-Mellon University, 1989.
14. D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
15. H. A. Blom, "An efficient filter for abruptly changing systems," in *The 23rd IEEE Conference on Decision and Control*. IEEE, 1984, pp. 656–658.
16. H. A. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.
17. K.-H. Lin and C.-C. Wang, "Stereo-based simultaneous localization, mapping and moving object tracking," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3975–3980.
18. D. F. Wolf and G. S. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, 2005.
19. T.-D. Vu, J. Burlet, and O. Aycard, "Grid-based localization and local mapping with moving object detection and tracking," *Information Fusion*, vol. 12, no. 1, pp. 58–69, 2011.
20. M. Simas, B. J. Guerreiro, and P. Batista, "Preliminary results on 2-d simultaneous localization and mapping for aerial robots in dynamics environments," in *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*, 2019, pp. 180–185.
21. F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.
22. Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*. Storrs, CT: YBS Publishing, 1995, vol. 19.
23. Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
24. T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*. IEEE, 1980, pp. 807–812.
25. S. Blackman and R. Popoli, "Design and analysis of modern tracking systems," Norwood, MA: Artech House, 1999.
26. C. Rago, P. Willett, and R. Streit, "A comparison of the jpdaf and pmht tracking algorithms," in *1995 International conference on acoustics, speech, and signal processing*, vol. 5. IEEE, 1995, pp. 3571–3574.
27. G. Chiron, P. Gomez-Krämer, and M. Ménard, "Detecting and tracking honeybees in 3d at the beehive entrance using stereo vision," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, p. 59, 2013.
28. S. Jouannin, L. Trassoudaine, and J. Gallice, "Comparison of data association methods. application to road obstacle tracking using a doppler effect radar," *IFAC Proceedings Volumes*, vol. 31, no. 3, pp. 489–494, 1998.
29. R. R. Pitre, V. P. Jilkov, and X. R. Li, "A comparative study of multiple-model algorithms for maneuvering target tracking," in *Signal Processing, Sensor Fusion, and Target Recognition XIV*, vol. 5809. International Society for Optics and Photonics, 2005, pp. 549–560.
30. J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.
31. W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.
32. M. Raitoharju and R. Piché, "On computational complexity reduction methods for kalman filter extensions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 10, pp. 2–19, 2019.
33. C. Cadena and J. Neira, "SLAM in $O(\log n)$ with the Combined Kalman-Information Filter," *Robotics and Autonomous Systems*, vol. 58, no. 11, pp. 1207–1219, 2010.
34. S. Cong and L. Hong, "Computational complexity analysis for multiple hypothesis tracking," *Mathematical and computer modelling*, vol. 29, no. 9, pp. 1–16, 1999.
35. X. Ren, Z. Huang, S. Sun, D. Liu, and J. Wu, "An efficient mht implementation using grasp," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 86–101, 2014.
36. P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *European Conference on Computer Vision*. Springer, 2012, pp. 214–227.
37. I. J. Cox and S. L. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 2, pp. 138–150, 1996.
38. J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on robotics and automation*, vol. 17, no. 6, pp. 890–897, 2001.